Detection of Hidden Information in Graphic Files using Machine Learning*

Alexander M. Kadan [1][0000-0003-3701-8100], Igor A. Sazanovetz [2][0000-0003-4531-2529]

[1] Yanka Kupala State University of Grodno, Grodno, Belarus

kadan@mf.grsu.by

[2] IntexSoft LLC, Grodno, Belarus

sazanovec_ia_13@mf.grsu.by

Abstract. The method of detecting the presence of hidden information in graphic files based on the use of machine learning methods is considered. Detection of the presence of hidden information is carried out in the absence of data on the original algorithm used to implement the hidden information. In steganalysis, methods that solve problems of this type are usually called blind. Dataset formation methods for teaching machine learning models using wavelet decomposition and test results of trained models on training data sets are described.

Keywords: hidden information, steganography, steganalysis, stegocontainer, graphical stegocontainer, blind steganalysis method, machine learning.

1 Introduction

Throughout the history of public relations, there has been a need to hide information or share it unnoticed by others. The combination of methods developed for these purposes has formed a scientific direction known as steganography.

Modern steganography methods widely use computer technology to embed hidden digital information (stego information) into other digital data called “container files” or “stegocontainers”, such as digital images, audio or video data, text, or even network packets.

In contrast to cryptography, which hides the meaning of the transmitted message, steganography hides the fact of message transmission, which is in some ways its advantage, since in this case unnecessary attention is not attracted. The interest in steganographic methods that have been emerging in recent years is largely because, in contrast to cryptography, the law does practically not regulate the use of steganography.

One of the key requirements for steganographic algorithms is that the implementation of information in steganocontainers should not noticeably change the size of the

file and/or quality of the container, such as image or sound. Therefore, steganographic algorithms often exploit the limitations of biological systems of human perception. For example, if the information is hidden in images, stegoalgorithms change the intensity of colors so that, on the one hand, encodes the stegoinformation with these changes, and on the other hand, that these changes are not perceived by the human organs of vision. Algorithms for working with sound are based on the same principle: the recorded information changes the high frequencies of the audio signal, which is probably not noticeable when listening [1].

Steganographic systems protect information primarily from the point of view of behavioral security, hiding the existence of information and communication behavior and thus ensuring the security of important information. Because of its powerful ability to hide information, the concealment system plays an important role in protecting privacy and security in cyberspace.

There are various storage media that can be used to hide information, including images [2,3], audio [4,5], text [6-8], etc. [9]. Among them, the images have a large information capacity, which has allowed the images in recent years to become a widely studied and used steganographic medium. However, protecting information security, these concealment systems can also be used by cybercriminals and transmit some malicious information, which creates potential risks for cyberspace security [10]. Therefore, the study and development of effective methods of steganalysis are becoming an increasingly promising and difficult task.

2 Tools and Scheme of Information Hiding

One of the common types of files in which messages are embedded in digital images. Digital images are typically presented in the formats * .bmp, * .jpeg, * .png or * .gif (without animation). The message can be represented by any type of digital information, for example by a file of a certain type or a line of text.

To write a hidden message to a file, special programs that implement stego algorithms are used. Once the implementation algorithm is known, you can write software that attacks the scanned images and finds out which contains hidden information and which does not.

Algorithmically, steganography consists of two phases: one for hiding information and one for extracting. The hiding process embeds the message (for example, a line entered in the terminal or another file) in the media file (or in the file container). As a result, we get a container object with an embedded message. In the process of extracting the message, on the contrary, the original message is extracted from the sterile. In case you still find the fact of the presence of a hidden message, in most steganographic programs, before embedding the message, it is first encrypted. The basic model of steganography is shown in Fig. 1.

3 Imaging Containing Hidden Data

Steganography algorithms work differently, and, accordingly, produce different types

of distortion of the original information. Because of this, it becomes hardly possible to write a clear deterministic algorithm for detecting the presence of a steganographic insertion. It is precisely in such situations machine learning methods are used.

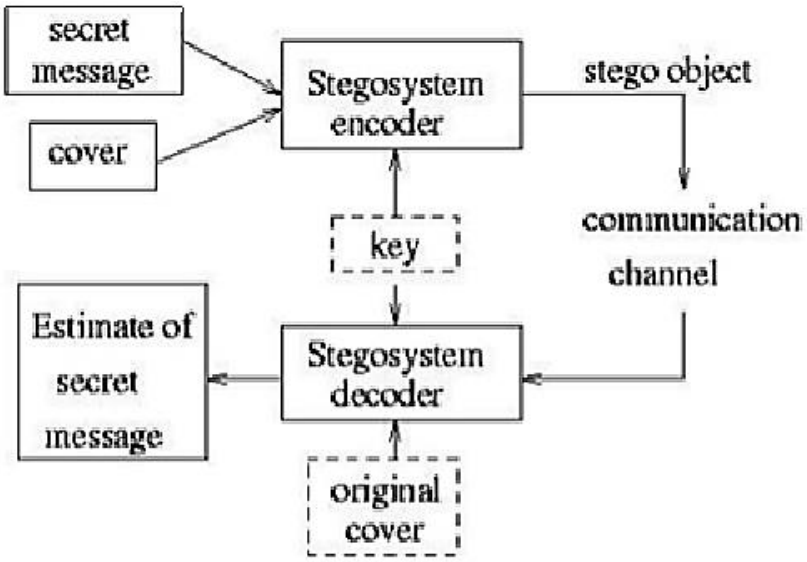

Fig. 1. The basic model of steganography

Over the past decade, many steganographic algorithms have been proposed for hiding data within a stegocontainer. Such embedding schemes can work in the spatial domain, such as, for example, MiPOD [11], STABYLO [12], SUNIWARD [13], HILL [14], WOW [15] or HUGO [16], as well as in the frequency domain of the image.

The vast majority of approaches to image steganalization are two-stage. At the first stage, useful information about the image content is generated by calculating the set of attributes, and at the second stage, it is used to teach the machine learning model that allows distinguishing empty steg containers from containers with hidden information.

For the first step, various Rich Models (RM) for the spatial domain (SRM) [17] and JPEG [18] were proposed, while for the second step, the most common choice is Ensemble Classifier (EC) [19]. This RM + EC combination is used in many modern image steganalization tools. So in [20], stegoimages obtained using the HUGO steganographic algorithm were detected with errors of 13% and 37%, respectively, for embedding payloads of 0.4 and 0.1 pp. These errors were slightly reduced (12% and 36%) in [21], and a similar model was applied to stego images obtained using the JUNIWARD steganographic scheme.

Since we are dealing with the task of blindly detecting the presence of stegoinformation, we should train the machine learning model using examples created using different algorithms.

In this work, the following programs were used to create training and test data sets:

- Steganography Software F5 (algorithm f5) [22];
- StegHide (steganography based on graph theory) [23];
- OpenStego (RandomLSB, a modified least significant bit algorithm) [24].

The graphic files for stego-information used in this work were taken from open sets on the Internet, as well as from the blogs of photographers, without copyright infringement. A total of 750 images were selected.

Most of these images were originally in high definition. However, models of machine learning methods are better trained using previously prepared small examples. Therefore, the size of the images was reduced to an average of between 640 x 480 and 1147 x 768.

4 Generating Characteristic Files

To use machine learning models, it is necessary to form a dataset of attributes based on the original graphic images, which would reflect the characteristic features of "clean" images and images containing stego information. The dataset is presented as a CSV file. Each dataset record contains 84 features, as well as the 85th classification feature: “0” for a “clean” image and “1” otherwise. As a result, the dataset contains 3000 records - 750 records for “clean” images, and 750 records for images with stego-information introduced by Steganography Software F5, StegHide, and OpenStego, respectively.

The signs characterizing the graphic image will be generated using discrete wavelet transforms and statistical moments 1-4 orders of magnitude.

By the time this work was completed, there were already attempts to detect the presence of steganographic content in graphic files using the discrete Fourier transform [5] or wavelet transforms [6] in combination with machine learning methods.

The wavelet transform is an integral transform that is a convolution of a wavelet function with a signal. The wavelet transform translates the signal from the time representation into the time-frequency [7].

Since we are dealing with digital images, it is worth considering and applying discrete wavelet transformations. The decomposition will be performed according to the wavelet functions of the Haar, db2, bior1.3, and rbio1.3 (Fig. 2).

Since each color channel in the image is represented by a rectangular matrix, the discrete wavelet transform must be two-dimensional. You can do this using the wavedec2 method from the PyWavelets module of the Python programming language. Usually, when analyzing images by a two-dimensional wavelet transform, decomposition no higher than the third level is used. Decompositions of higher levels usually do not provide valuable additional information.

The wavedec2 function returns a structure of the form [cAn, (cHn, cVn, cDn), ... (cH1, cV1, cD1)], where:

- cAn is the approximation coefficient of decomposition of the nth level;
- cHn is the horizontal decomposition coefficient of the nth level;
- cVn is the vertical decomposition coefficient of the nth level;

– cDn is the diagonal expansion coefficient of the nth level.

Each of these coefficients is a multidimensional array. To apply statistical methods to it, we will make it one-dimensional using the flatten() function from the NumPy module for Python. Before creating a frequency dictionary, we additionally round up all the coefficients to integers. This will avoid the presence of many coefficients in the frequency dictionary, slightly differing from each other, having a frequency of 1.

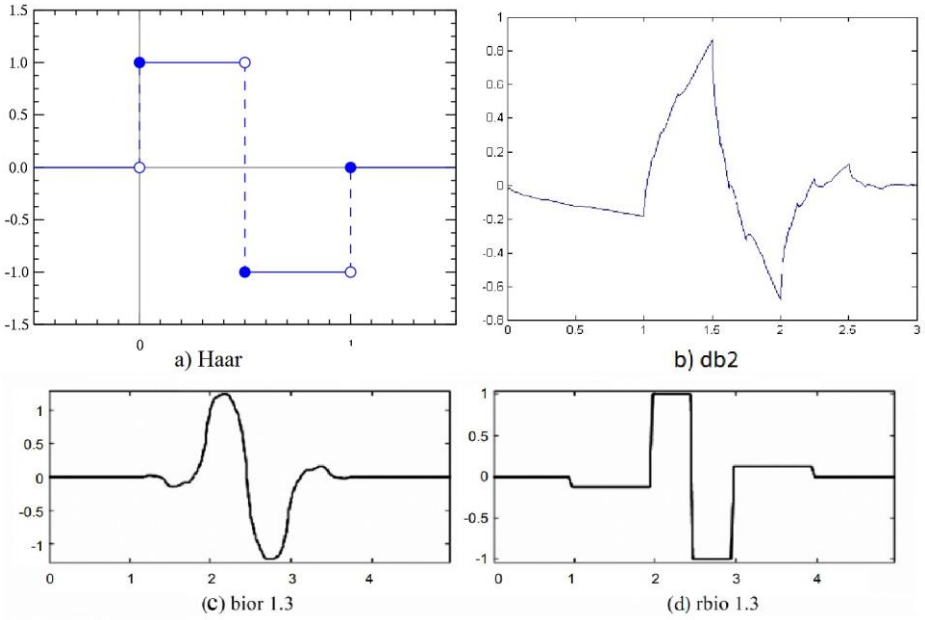

Fig. 2. Representation of wavelet functions a) Haar, b) db2, c) bior1.3, d) rbio1.3

Next, we calculate the statistical moments from the first to the fourth-order on such a frequency plane. To do this, use the describe method from the scipy.stats module. The described method returns an object from which you can get: stat. mean - average value; stat.variance - variance; stat.skewness asymmetry coefficient; stat.kurtosis coefficient of excess.

Similarly, datasets in the format of CSV files for other wavelet functions can be obtained.

5 Machine Learning Methods in Steganalysis

Stegoanalysis can be considered as a problem of two-class classification of machine learning, for the solution of which a whole range of methods can be used. For example, K-nearest neighbors, decision trees, random forest algorithm, support vector method, neural network technologies.

Traditional modern techniques of image steganalysis typically consist of a classifier

trained using the features provided by rich models. Since the stages of extracting and classifying features are perfectly implemented in the deep learning architecture and convolutional neural networks (CNN), various studies have tried to develop a CNN-based stego analyzer.

Deep learning [25-26] led to breakthrough improvements in various complex tasks in the field of computer vision, becoming modern for many of them. A key reason for this success is the current availability of powerful computing platforms, in particular GPU-accelerated ones. Among the various network architectures that belong to this family of machine learning methods, convolutional neural networks (CNNs) [27] are very effective for solving image classification problems. For example, in the MNIST problem, which consists of the automatic recognition of handwritten numbers [28] or the tasks of the CIFAR benchmark test [29]. Since steganalysis is a similar problem, the goal is to classify the input image as a cover or as a stego, the development of a CNN-based stego analyzer in the past few years has attracted increasing attention.

6 Application of Machine Learning Methods

The task of blindly detecting the presence of stegoinformation in a graphic image, considered in this paper, relates to the problems of binary classification and can be solved within the framework of machine learning technology with a teacher.

An experiment was conducted on the application of the following methods to solve the problem, using their implementations from the scikit-learn library for Python:

- K-nearest neighbors algorithm;
- naive Bayes classifier;
- decision tree;
- linear regression;
- method of support vectors;
- neural network direct distribution.

Records of the prepared dataset were evenly mixed and divided into training and test sets in a ratio of 8:2.

For more reasonable model training, a cross-validation mechanism was used, which provides for dividing the training sample into n equal parts with n-fold repetition of the training process. Moreover, for the kth time, model training takes place in all parts except the kth one, and it is the kth part of the training data set that is used to test the quality of training.

The experiment was carried out using two methods of scaling (normalizing) the MinMaxScaling and StandardScaling dataset data. With MinMax normalization, the values of some attribute in all records of the dataset are changed so that they fit into a fixed range (from 0 to 1). With standard normalization, the data has an average of 0 and a standard deviation from the average of 1.

After applying the normalization phase, the model training phase was carried out. The training was carried out on a pair of feature sets - test and training for the generation of which the selected wavelet function was used.

Machine learning algorithms have, depending on their type, parameters, and/or hyperparameters. For convenience, where possible, the enumeration of parameter variants was automated using the GridSearchCV tool included in the scikit-learn library.

Some learning outcomes are presented in table 1:

Table 1. Some results of machine learning methods (Haar wavelet)

Machine learning method	Method Parameters / Hyperparameters	The level of correct predictions (cross-validation results, n = 5)
K-Nearest Neighbor Algorithm	K=3, metric=euclidean	[0.56 0.61 0.6 0.61 0.57]
	K=5, metric=euclidean	[0.64 0.68 0.63 0.65 0.65]
	K=10, metric=euclidean	[0.7 0.67 0.66 0.65 0.67]
	K=3, metric= manhattan	[0.56 0.6 0.61 0.59 0.55]
	K=5, metric= manhattan	[0.68 0.63 0.67 0.57 0.64]
	K=10, metric= manhattan	[0.66 0.66 0.69 0.7 0.63]
Bayesian Classifier	N/A	[0.55 0.52 0.61 0.55 0.51]
Decision tree	max_depth=5, max_features=2	[0.56 0.55 0.66 0.57 0.59]
	max_depth=5, max_features=10	[0.58 0.67 0.64 0.61 0.6]
	max_depth=5, max_features=40	[0.65 0.58 0.61 0.56 0.67]
	max_depth=10, max_features=2	[0.52 0.59 0.54 0.59 0.57]
	max_depth=10, max_features=10	[0.55 0.52 0.59 0.54 0.64]
	max_depth=10, max_features=40	[0.59 0.59 0.59 0.54 0.54]
	max_depth=20, max_features=2	[0.49 0.53 0.52 0.55 0.55]
	max_depth=20, max_features=10	[0.53 0.52 0.52 0.55 0.55]
	max_depth=20, max_features=40	[0.5 0.59 0.57 0.53 0.53]
Random Forest Algorithm	n_estimators=2, max_depth=20, max_features=20	[0.54 0.58 0.54 0.56 0.58]
	n_estimators=5, max_depth=20, max_features=20	[0.57 0.55 0.54 0.63 0.5]
	n_estimators=10, max_depth=20, max_features=20	[0.59 0.57 0.59 0.62 0.6]
	n_estimators=20, max_depth=20, max_features=20	[0.63 0.56 0.55 0.59 0.56]
	n_estimators=40, max_depth=20, max_features=20	[0.59 0.54 0.55 0.6 0.55]
Support Vector Method	C=100, gamma=0.001, kernel='rbf'	[0.66 0.65 0.67 0.66 0.66]
	C=1000, gamma=0.001, kernel='rbf'	[0.73 0.68 0.71 0.74 0.7]
	C=1000, gamma=0.01, kernel='rbf'	[0.7 0.7 0.7 0.7 0.68]
Multilayer perceptron	hidden_layer_sizes=[90, 20], alpha=0.0001	[0.76 0.71 0.65 0.73 0.72]
	hidden_layer_sizes=[30, 30], alpha=0.0001	[0.69 0.78 0.73 0.72 0.69]
	hidden_layer_sizes=[90, 20], alpha=0.0001	[0.72 0.74 0.74 0.75 0.68]
	hidden_layer_sizes=[30, 30], alpha=0.001	[0.72 0.74 0.75 0.74 0.68]

As can be seen from the table, the best results were obtained using the support vector method and direct distribution neural network with the “multilayer perceptron” architecture with two hidden layers, each of which has several tens of neurons.

In the case of using the db2 and bior1.3 wavelet functions, the results were worse than when using the Haar wavelet function.

They are also worse when using min-max normalization instead of the standard one.

The most significant influence on the result was the significance of the features associated with the moments of the third and fourth orders. Features associated with moments of the first and second-order did not significantly affect the result. Also, the values of the features calculated on the horizontal decomposition coefficients did not significantly affect the result.

7 Conclusions

Based on the approaches described in the work, a console application was developed using the methods of the scikit-learn library, which allows the use of a classical method of machine learning and trained a neural network to search for signs of hidden information in graphic files.

As a result of the experiment on the blind detection of stegoinformation in graphic files using machine learning methods, it was found that the best results were obtained the support vector method with parameters C = 1000, gamma = 0.001 and kernel = RBF, and a multilayer perceptron with two hidden layers (90 and 20 neurons).

As effective features, asymmetry and kurtosis coefficients calculated in the frequency plane for approximation, vertical, and diagonal coefficients of a two-dimensional three-level wavelet transform using the Haar wavelet function can be used. Using trained models, it is possible with a probability close to 0.7 to predict whether a graphic image contains a hidden message or not.

References

1. Wheeler D. Audio Steganography Using High Frequency Noise Introduction (2012). Available at: https://pdfs.semanticscholar.org/d547/3318c5c9171fe38abc550b89a15022d559cb.pdf
2. Fridrich J. Steganography in digital media: principles, algorithms, and applications. Cambridge University Press, 2009.
3. Chen K., Zhou H., Zhou W., Zhang W., and Yu N. Defining cost functions for adaptive jpeg steganography at the microscale. IEEE Transactions on Information Forensics and Security, vol. 14, no. 4, pp. 1052–1066, 2019.
4. Yang Z., Peng X., and Huang Y. A sudoku matrix-based method of pitch period steganography in low-rate speech coding. In International Conference on Security and Privacy in Communication Systems. Springer, 2017, pp. 752–762.
5. Yang Z., Du X., Tan Y., Huang Y., and Zhang Y.-J. Aag-stega: Automatic audio generation-based steganography. ArXiv preprint arXiv:1809.03463, 2018.
6. Yang Z.-L., Guo X.-Q., Chen Z.-M., Huang Y.-F., and Zhang Y.-J. Rnn-stega: Linguistic

steganography based on recurrent neural networks. IEEE Transactions on Information Forensics and Security, vol. 14, no. 5, pp. 1280–1295, 2019.
7. Yang Z., Zhang P., Jiang M., Huang Y., and Zhang Y.-J. Rits: Real-time interactive text steganography based on automatic dialogue model. In International Conference on Cloud Computing and Security. Springer, 2018, pp. 253–264.
8. Yang Z., Jin S., Huang Y., Zhang Y., and Li H. Automatically generate steganographic text based on Markov model and Huffman coding. ArXiv preprint arXiv:1811.04720, 2018.
9. Johnson N. F. and Sallee P. A. Detection of hidden information, covert channels and information flows. Wiley Handbook of Science and Technology for Homeland Security, 2008.
10. Theohary C. A. Terrorist use of the internet: Information operations in cyberspace. DIANE Publishing, 2011.
11. Sedighi V., Cogranne R., and Fridrich J. Content-adaptive steganography by minimizing statistical detectability. IEEE Transactions on Information Forensics and Security, vol. 11, no. 2, pp. 221–234, Feb 2016.
12. Couchot J., Couturier R., and Guyeux C. STABYLO: steganography with adaptive, bbs, and binary embedding at low cost. Annales des Tel´ e-communications ´, vol. 70, no. 9-10, pp. 441–449, 2015. [Online]. DOI: http://dx.doi.org/10.1007/s12243-015-0466-7
13. Holub V., Fridrich J., and Denemark T. Universal distortion function for steganography in an arbitrary domain. EURASIP Journal on Information Security, vol. 2014, no. 1, 2014. [Online]. DOI: http://dx.doi.org/10.1186/1687-417X-2014-1
14. Li B., Wang M., Huang J., and Li X. A new cost function for spatial image steganography. in 2014 IEEE International Conference on Image Processing (ICIP). IEEE, 2014, pp. 4206–4210.
15. Holub V. and Fridrich J. J. Designing steganographic distortion using directional filters. In WIFS. IEEE, 2012, pp. 234–239.
16. Pevny T., Filler T., and Bas P. Using high-dimensional image models ´ to perform highly undetectable steganography. In Information Hiding - 12th International Conference, IH 2010, Calgary, AB, Canada, June 28-30, 2010, Revised Selected Papers, ser. Lecture Notes in Computer Science, R. Bohme, P. W. L. Fong, and R. Safavi-Naini, ¨ Eds., vol. 6387. Springer, 2010, pp. 161–177. [Online]. DOI: http://dx.doi.org/10.1007/978-3-642-16435-4
17. Fridrich J. and Kodovsky J. Multivariate Gaussian model for designing ´ additive distortion for steganography. In Acoustics, Speech, and Signal Processing (ICASSP), 2013 IEEE International Conference on, May 2013, pp. 2949–2953.
18. Holub V. and Fridrich J. J. Low-complexity features for JPEG steganalysis using undecimated DCT. IEEE Trans. Information Forensics and Security, vol. 10, no. 2, pp. 219–228, 2015. [Online]. DOI: http://dx.doi.org/10.1109/TIFS.2014.2364918
19. Kodovsky J., Fridrich J. J. and Holub V. Ensemble classifiers ´ for steganalysis of digital media. IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 432–444, 2012. [Online]. DOI: http://dx.doi.org/10.1109/TIFS.2011.2175919
20. Holub V. and Fridrich J. J. Random projections of residuals for digital image steganalysis. IEEE Trans. Information Forensics and Security, vol. 8, no. 12, pp. 1996–2006, 2013. [Online]. DOI: http://dx.doi.org/10.1109/TIFS.2013.2286682
21. Dang-Nguyen D.-T., Pasquini C., Conotter V. and Boato G. RAISE: a raw images dataset for digital image forensics. In Proceedings of the 6th ACM Multimedia Systems Conference, MMSys 2015, Portland, OR, USA, March 18-20, 2015, W. T. Ooi, W. chi Feng, and F. Liu, Eds. ACM, 2015, pp. 219–224. [Online]. Available at: http://dl.acm.org/citation.cfm?id=2713168
22. F5-steganography. The world's leading software development platform – GitHub. Available at: https://github.com/matthewgao/F5-steganography

23. Steghide. Sourceforge. Available at: http://steghide.sourceforge.net
24. OpenStego. Available at: https://www.openstego.com
25. Dang-Nguyen D.-T., Pasquini C., Conotter V., and Boato G. RAISE: a raw images dataset for digital image forensics. In Proceedings of the 6th ACM Multimedia Systems Conference, MMSys 2015, Portland, OR, USA, March 18-20, 2015, W. T. Ooi, W. chi Feng, and F. Liu, Eds. ACM, 2015, pp. 219–224. [Online]. Available at: http://dl.acm.org/citation.cfm?id=2713168
26. LeCun Y., Bengio Y. and Hinton G. Deep learning. Nature, vol. 521, no. 7553, pp. 436–444, 2015.
27. Krizhevsky A., Sutskever I. and Hinton G. E. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, 2012, pp. 1097–1105.
28. Wan L., Zeiler M., Zhang S., Cun Y. L. and Fergus R. Regularization of neural networks using dropconnect. In Proceedings of the 30th International Conference on Machine Learning (ICML-13), 2013, pp. 1058–1066.
29. Xu G., Wu H.-Z. and Shi Y.-Q. Structural design of convolutional neural networks for steganalysis. IEEE Signal Processing Letters, vol. 23, no. 5, pp. 708–712, 2016