A New Approach for Network Steganography Detection based on Deep Learning Techniques

Cho Do Xuan[1], Lai Van Duong[2]
Information Assurance Department
FPT University, Hanoi
Vietnam

***Abstract*—One of the techniques that current cyber-attack methods often use to steal and transmit data out is to hide secret data in packets. This is the network steganography technique. Because millions of packets are sent and received every hour in internet activity, so it is very difficult to detect the theft and transmission of system data out using this form. Recent approaches often seek ways to compute and extract abnormal behaviors of packets to detect a steganography protocol or technique. However, such methods have the difficult problem of not being able to detect abnormal packets when an attacker uses other steganography techniques. To solve the above problem, this paper proposes a network steganography detection method using deep learning techniques. The highlight of this study is some new proposed features based on different components of the packet. By combining these many components, this proposal will not only provide the ability to detect many steganography techniques in the network, but also improve the ability to accurately detect abnormal packets. Besides, this study proposes to use deep learning for the task of detecting normal and abnormal packets. The authors want to take advantage of the big data analysis and processing capabilities of deep learning models in order to improve the ability to analyze and detect network steganography techniques. The experimental results in Section IVD have proved the effectiveness of this proposed method compared with other approaches.**

Keywords*—Network steganography; network steganography detection method; abnormal packets; deep learning techniques

I. Introduction

A. The Problem

The study [1] listed 11 different techniques commonly used to hide information in the network. These techniques are generally divided into three main technique groups: packet modification, stream modification, and hybrid. The research [2] presented some main difficulties that make it very difficult to detect and prevent network steganography techniques. To fix the problems in the research [2], current approaches often use two main methods: i) technique-specific methods, comprises methods proposed as countermeasures for specific steganographic techniques. Methods in this category usually operate on low-level network data, require relatively much computation resources, and are not able to detect other steganographic techniques instead of the one or several for which they are designed; ii) generic methods, comprises methods that are not designed to detect one specific steganographic technique but offer a comprehensive approach to network anomaly detection and categorization of network traffic for potential steganographic utilization. Methods in this category may not provide detailed information on detected suspicious traffic but can label it for further investigation. Most generic methods fall into two subcategories that characterize their approach: statistical or machine learning. The studies [3, 4, 5, 6, 7, 8] presented several studies and proposals for detecting network steganography based on the abnormal behavior analysis technique and the ruleset database. However, noticed that these approaches have two problems [1, 2, 9, 10, 11, 12]: using the available dataset and focusing on detecting only one steganography technique. Therefore, although these studies brought very high efficiency on experimental datasets, when applied in reality, they did not bring the desired result. To solve the above problems, this paper proposes a new method based on a group of generic methods. Specifically, this proposal will seek a way to optimize two main problems: i) defining and proposing features and characteristics of abnormal behavior of network steganography techniques; ii) use deep learning techniques on the basis of big data analysis to detect and classify cyber-attack techniques based on their unusual behavior defined in the task (i). Details of abnormal behaviors and algorithms for classifying network steganography techniques are presented in Section III of the paper. The results of evaluating the effectiveness of the proposed method are presented in detail in Section IV of the paper. Evaluation, conclusion, and future development direction are presented in Section V of the paper.

B. Contributions of the Paper

The practical significance and scientificity of this paper include:

- Proposing some features and characteristics of the packet. The features proposed in the paper are new study, and are synthesized and extracted on many different components of the packet. The experimental results have proved these proposed features have brought many meanings.
- Proposing the use of deep learning models for the task of detecting network steganography. In the experimental section, this paper tunes the parameters in each deep learning model to provide the ability to choose for the systems to ensure a balance between the time and the efficiency of the detection method.

II. Related Work

In the study [13], Mike et al. proposed a method to detect network steganography using the IDS tool. Specifically, the authors used the rulesets built in the IDS tool to detect hidden information in the network based on data sections of packets. In the study [14], the authors proposed a method to detect steganography in VoIP using the Least Significant Bits technique. Taeshik Sohn et al. [15] proposed a network steganography detection method using the Support Vector Machine (SVM) algorithm for detecting hidden information in TCP/IP protocols. Similarly, research [16] proposed using the Naive Bayes algorithm to detect secret information hidden in TCP/IP header. Cho et al. [3] proposed a method of detecting storage-based network steganography using machine learning. Specifically, in their research, the authors used the Random Forests (RF) algorithm to classify abnormal behaviors on ICMP and TCP/IP packets. Smolarczyk [2] proposed a method to detect steganography in the network using the multi-layer analysis technique.

III. Proposing the Network Steganography Detection Method using Deep Learning

A. Proposing the Method to Select and Extract Abnormal Behavior of Network Steganography Techniques

As mentioned above, the purpose of this paper is to use deep learning algorithms to detect network steganography based on analyzing different components of the packet. Specifically, three network steganography techniques studied in this paper are:

- Size Modulation: The covert channel uses the size of a header element or of a PDU to encode the hidden data.
- Random Value: The covert channel embeds hidden data in a header element containing a random value
- Reserved/Unused: The covert channel encoded hidden data into a reserved or unused header/PDU element.

Based on these attack techniques, this study will find ways to collect and analyze packets to look for their abnormal behaviors. Table I present 38 features proposed to extract and chose to use.

TABLE I. Proposed Abnormal Features of the Packet

No.	Category	Feature	Description	Type
1	IP	ip.id	IP Identification	Integer
2		ip.flags	IP Flags	Integer
3		ip.frag_offset	IP Fragment Offset	Integer
4		ip.checksum	IP Header Checksum	Integer
5		ip.ttl	IP Time to live	Integer
6		ip.tos	IP Type of Service	Integer
7		ip.src	IP Source Address	String
8		ip.dst	IP Destination address	String
9		ip.num_option	IP Number of options	Integer
10	TCP	tcp.flags	TCP Flags	Integer
11		tcp.checksum	TCP Checksum	Integer
12		tcp.seq	TCP Sequence Number	Integer
13		tcp.flags.res	TCP Reserved	Boolean
14		tcp.urgent_pointer	TCP Urgent Pointer	Integer
15		tcp.ack	TCP Acknowledgment Number	Integer
16		tcp.srcport	TCP Source Port	Integer
17		tcp.dstport	TCP Destination Port	Integer
18	UDP	udp.srcport	UDP Source Port	Integer
19		udp.dstport	UDP Destination Port	Integer
20	ICMP	icmp.type	ICMP Type number	Integer
21		icmp.unused	ICMP Unused	Bytes
22		icmp.reserved	ICMP Reserved	Bytes
23		icmp.seq	ICMP Sequence number (BE)	Integer
24		icmp.seq_le	ICMP Sequence number (LE)	Integer
25		icmp.length	ICMP Length	Integer
26		icmp.ident	ICMP Identifier (BE)	Integer
27		icmp.checksum	ICMP Checksum	Integer
28	Frame	frame.number	Frame number	Inetger

29		frame.len	Frame length on the wire	Integer
30		frame.cap_len	Frame length stored into the capture file	Integer
31	Ethernet	eth.src	Source	String
32		eth.dst	Destination	String
33	IPv6	ipv6.nxt	Next Header	Integer
34		ipv6.src	Source Address	String
35		ipv6.dst	Destination Address	String
36	MQTT	mqtt.topic	Topic	String
37		mqtt.msgtype	Message Type	Integer
38		mqtt.client.id	Client ID	String

B. The Detection Method

Thus, based on features of anomalous behaviors of packets defined and extracted in Table I, this paper will propose a method to classify these packets. It can be seen that to detect network steganography, previous studies often used algorithms such as SVM [4, 15], RF [3]. To improve the efficiency of the network steganography detection method, this paper proposes to use some deep learning algorithms and models. Specifically, some deep learning algorithms and models proposed to use include: Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long short term memory (LSTM). Regarding the MLP network, the study [17] presented in detail the architecture of an MLP network that is built by simulating the way neurons work in the human brain. MLP networks usually have 3 or more layers including 1 input layer, 1 output layer, and more than 1 hidden layer. Besides, the efficiency of the MLP network depends on the activation function. This paper will tune activation functions to evaluate the effectiveness and suitability of activation functions for the network intrusion detection task. The CNN network is defined as a set of basic layers including convolution layer + nonlinear layer, fully connected layer. The detailed structure of CNN as well as the terms (stride, padding, MaxPooling) are presented in detail in the paper [18]. In which, the activation function used is ReLU.

The study [19] introduced the LSTM and its ability to remember information for a long time. This is reflected in the structure of the gates in each memory cell. A memory cell consists of three main components: input gate, forget gate, and output gate. Firstly, the forget gate decides what information should be discarded in the cell state. Next, the input gate decides what information is updated into the cell state. Finally, the output gate performs computing the desired output. During this process, the cell state is propagated through and updated when it passes through all nodes.

IV. Experiments and Evaluation

A. Description of Data Collection Method

1) For normal dataset: The dataset of normal packets is collected at [20]. This dataset belongs to the "MAWI Working Group" and the "WIDE Project" which collected network traffic at ISP points in Japan. PCAP files are network traffic collected on April 30, 2021. Then 2,200,000 packets in these PCAP files are taken to conduct experiments. Table II shows the number of collected and extracted normal packets.

2) For stego dataset: This study proposes to use some network steganography tools to generate stego packets. Table III below describes the tools and the steganography type of these tools in detail.

After successfully installing the above tools, running those tools and use Wireshark to capture the network traffic generated by those tools. Network traffic generated by each tool is saved as separate files. For example, network traffic generated by ptunnel is saved as a separate PCAP file, network traffic generated by covert_tcp is also be saved as a separate PCAP file. Only packets generated by these tools are saved. Other unrelated packets such as ARP or system packets are deleted to ensure that the PCAP file contains only stego packets containing secret information. Table IV below presents the number of stego packets generated by the tools listed in Table III.

TABLE II. Support Tools for Generating Normal Packets

Name	**Generation tool**	**The number of packets**
Normal Packet	mawi.wide.jp	2,200,000

TABLE III. Support Tools for Generating Packets Containing Secret Information

Tool	**Protocol**	**Field**
Ptunnel	ICMP	Payload
pingtransfer	ICMP	Payload
dns2tcp	DNS/UDP	TXT record
covert_tcp	TCP/IP	IP Identification / TCP ISN
syn-file	TCP	TCN Syn Sequence number
netcat	TCP	Space in payload
hcovert	TCP	Segment data

TABLE IV. The Total Number of Generated Stego Packets

Name	**Generation tool**	**The number of packets**
Size Modulation	netcat	421,839
Random Value	covert_tcp, syn-file	1,703,306
Reversed/Unused	ptunnel, pingtransfer, dns2tcp	481,651
Others	hcovert	109,208
Total		**2,716,004**

3) Data synthesis: Based on the data collection method in Sections 1) and 2) above, obtaining an aggregated dataset for training and testing for detecting network steganography as shown in Table V.

TABLE V. THE TOTAL NUMBER OF COLLECTED PACKETS

Description	The number of packets
Normal packet	2,200,000
Stego packet	2,716,004
Total	**4,916,004**

B. Evaluation Criteria

The following measures will be used in this paper to evaluate the accuracy of models:

- Accuracy: The ratio between the number of samples classified correctly and total number of samples.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (1)$$

In which: TP - True positive: The number of stego packets classified correctly; FN - False negative: The number of stego packets classified as normal; TN - True negative: The number of normal packets classified correctly; FP - False positive: The number of normal packets classified as stego.

- Precision: The ratio between the true positive value and total number of samples classified as positive. The higher value of precision, the more accurate in stego packet detection.

$$precision = \frac{TP}{TP + FP} \times 100\% \quad (2)$$

- Recall: The ratio between the true positive value and the total real stego packets. The higher value of recall, the lower rate of missing positive samples.

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (3)$$

- F1-score: The harmonic mean of precision and recall. The higher F1 score, the better the model is

$$F1\text{score} = \frac{2 \times precision \times \text{Re}call}{precision + \text{Re}call} \quad (4)$$

- TP: The same to Recall. This shows the ability to detect the true stego packet.
- FP: The ratio between false positive value and the false positive plus true negative. This shows the false alarm rate of stego packet.

C. Experimental Scenario

1) Scenario for experimental dataset: With the experimental dataset listed in Table V, the dataset is divided into different parts and then conduct experiments and evaluate the accuracy of the proposed models based on these experimental datasets. The whole process of dividing the experimental dataset into the scenarios will be chosen randomly in which 80% of the dataset is used in the training process, the remaining 20% is used in the testing process.

2) Evaluation scenarios: To see the effectiveness of the proposed method, this paper conducts two experimental scenarios as follows:

- Scenario 1: Compare and evaluate the effectiveness of deep learning methods. For this scenario, this study conducts the evaluation according to the following algorithms: MLP, CNN, LSTM. During the experiment process, the authors tune parameters to see the effectiveness of the deep learning models.
- Scenario 2: Compare and evaluate the deep learning model with some other approaches on the same dataset.

D. Experimental Results

1) Experimental results of scenario 1: Comment: From the experimental results in Tables VI, VII, VIII, noticed that:

- Regarding accuracy: Based on the classification results, found that the LSTM model yielded better performance than other deep learning models. Specifically, at the Accuracy measure, the LSTM model reached the absolute rate with two and three layers. This result is higher than that of CNN models by 1.5 % and MLP by 1.45%. Similarly, with the Recall measure, the LSTM model is higher than other models from 0.01 to 0.3%. In general, deep learning models brought high efficiency for the task of classifying normal and abnormal packets. The authors think the reason is that the packets are analyzed by us into different components, and then features are extracted from these components. This makes their abnormal behaviors are highlighted so it supports the classification process better. In addition, deep learning models, especially LSTM with the ability to remember features and hidden layers, have synthesized many important features. Therefore, it can be seen that this proposal is completely correct and reasonable.
- Regarding prediction time: Based on the experimental results, noticed that the LSTM model takes more time than other models for both training and testing processes. In particular, the training time of the LSTM model is about 2 times higher than the CNN model and 7 times higher than the MLP model. Regarding the detection time, the LSTM model is about 12 times higher than the CNN model and 4 times higher than the MLP model. From this result, seeing that although the LSTM model is more efficient than other models, they are many times more time-consuming than other models. Therefore, in reality, monitoring and detection systems need to choose the appropriate model to balance both detection time and efficiency.

TABLE VI. EXPERIMENTAL RESULTS OF DETECTING NETWORK STEGANOGRAPHY USING MLP

MLP	**1st**	**2nd**	**3rd**
Precision	1.00	1.00	1.00
Recall	0.97	0.97	0.97
F1- Score	0.98	0.98	0.98
Accuracy (%)	98.50	98.50	98.67
Training time(s)	263.28	377.65	465.67
Prediction time(s)	37.06	41.43	41.24

Deep Learning Model	**N. of layer**	**N. of nodes / Dimension of hidden state**
MLP	5	32-16-16-8-8-1
	5	64-32-16-8-1
	4	64-32-16-1

TABLE VII. EXPERIMENTAL RESULTS OF DETECTING NETWORK STEGANOGRAPHY USING THE CNN MODEL

CNN	**1st**	**2nd**	**3rd**
Precision	0.97	0.98	0.98
Recall	0.99	0.99	0.99
F1- Score	0.98	0.98	0.99
Accuracy (%)	98.1	98.3	98.5
Training time(s)	1163	1129.6	2557.5
Prediction time(s)	22.8	19.67	21.5

Deep Learning Model	**N. of layer**	**N. of nodes / Dimension of hidden state**
CNN	2	32 - 32
	2	64 - 64
	3	64 – 64-64

TABLE VIII. EXPERIMENTAL RESULTS OF DETECTING NETWORK STEGANOGRAPHY USING THE LSTM MODEL

LSTM	**1st**	**2nd**	**3rd**
Precision	0.99	1.00	1.00
Recall	1.00	1.00	1.00
F1- Score	0.99	0.99	0.99
Accuracy (%)	99.99	100	100
Training time(s)	29439.51	29812.13	30810.10
Prediction time(s)	53.69	136.12	175.51

Deep Learning Model	**N. of layer**	**N. of nodes / Dimension of hidden state**
LSTM	1	64
	2	64 - 128
	3	64 – 128 - 64

2) Experimental results of scenario 2: For this scenario, this paper will compare and evaluate the effectiveness of this proposed method with two other algorithms, RF and SVM, which were proposed in previous research. Table IX describes the experimental results of these algorithms.

TABLE IX. EXPERIMENTAL RESULTS OF DETECTING NETWORK STEGANOGRAPHY WITH SOME OTHER APPROACHES

Algorithm	**RF [3]**	**SVM [4, 15]**	**LSTM [This proposal]**
Precision	0.98	0.95	1
Recall	0.95	0.989	1
F1- Score	0.96	0.96	0.99
Accuracy (%)	0.972	0.962	1
Training time(s)	777.33	1430.5	29812
Prediction time(s)	15.71	27.61	136.12

The results in Table IX show that the LSTM model proposed in this study gave 2% to 5% better performance than other algorithms in the same approach. It can be seen that the result of this study is superior to other related studies. The reason is that this study has proposed new meaningful features, and the deep learning classification algorithm also developed the ability to synthesize and analyze features.

V. CONCLUSION

In this paper, with the purpose to propose a new method to improve the efficiency of the network steganography detection process, the study has accomplished two tasks: i) proposing features and characteristics of abnormal packets; ii) using deep learning models for the abnormal packet classification task. Regarding the problem of analyzing abnormal features and characteristics, based on different components of the packet, this study has extracted many important and meaningful features. This is a breakthrough proposal in the task of analyzing and extracting features of packets. Regarding proposing the deep learning model, this study has succeeded in training the models to support the classification process. The experimental results in section IV.D have proved that this approach not only has scientific meaning, but also has many practical meanings, because this proposal has yielded better results than other models on all metrics. In the future, to improve the ability to detect abnormal packets, based on the research results in this paper, the authors think that it is possible to consider improving and supplementing two main issues: i) abnormal features of packets; ii) classification methods using combined deep learning networks or Attention networks.

REFERENCES

[1] Steffen Wendzel , Sebastian Zander , Bernhard Fechner , Christian Herdin, “Pattern-Based Survey and Categorization of Network Covert Channel Techniques,” ACM Computing Surveys, vol. 47(3), pp. 1–26. https://doi.org/10.1145/2684195.

[2] Smolarczyk M., Szczypiorski K., Pawluk J., “Multilayer Detection of Network Steganography,” Electronics, vol. 9 (12), pp. 2128, 2020. https://doi.org/10.3390/electronics9122128.

[3] Cho D.X, Thuong D.T.H, Dung N.K, “A Method of Detecting Storage Based Network Steganography Using Machine Learning,” Procedia Computer Science, vol. 154, pp. 543-548, 2019.

[4] M. Chourib, “Detecting Selected Network Covert Channels Using Machine Learning,” 2019 International Conference on High Performance Computing & Simulation (HPCS), pp. 582-588, 2019. https://doi.org/10.1109/HPCS48598.2019.9188115.

[5] James Collins, Sos Agaian, “Trends toward real-time network data steganography.” arXiv, arXiv:1604.02778.

[6] S. Sayadi, T. Abbesm, A. Bouhoula, "Detection of Covert Channels Over ICMP Protocol," 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), pp. 1247-1252, 2017. https://doi.org/10.1109/AICCSA.2017.60.

[7] Murdoch S.J., Lewis S, "Embedding Covert Channels into TCP/IP," In: Information Hiding (IH 2005), Springer, vol 3727, 2005. https://doi.org/10.1007/11558859_19.

[8] Lubacz J., Mazurczyk W., Szczypiorski K., "Principles and overview of network steganography," IEEE Commun. Mag., vol. 52, pp. 225–229, 2014.

[9] Fraczek, W.; Mazurczyk, W.; Szczypiorski, K. Hiding information in a Stream Control Transmission Protocol. Comput. Commun. 2012, 35, 159–169.

[10] Bieniasz J., Stepkowska M., Janicki A., Szczypiorski K., "Mobile agents for detecting network attacks using timing covert channels," J. Univ. Comput. Sci., pp. 1109–1130, 2019.

[11] Lu S., Chen Z., Fu G., Li Q., "A Novel Timing-based Network Covert Channel Detection Method," J. Phys. Conf. Ser., 1325, 012050, 2019.

[12] Szczypiorski K., Tyl T, "MoveSteg: A Method of Network Steganography Detection," Int. J. Electron. Telecommun., vol. 62, pp. 335–341, 2016.

[13] Mike Sieffert, Rodney Forbes, Charles Green, Leonard Popyack, Thomas Blake, "Assured Information Security: Stego Intrusion Detection System," The Digital Forensic Research Conference, 2004.

[14] Dittmann J, Hesse D, Hillert R, "Steganography and steganalysis in voice-over IP scenarios: operational aspects and first experiences with a new steganalysis tool set," In: Proc SPIE, Security, Steganography, and Watermarking of Multimedia Contents VII, vol 5681, pp. 607–618, 2005.

[15] Taeshik Sohn, JungTaek Seo, Jongsub Moon, "A study on the covert channel detection of TCP/IP header using support vector machine," In Proceedings of the 5th international conference of information and community security, pp. 313–324, 2003.

[16] Ms. Apurva, N. Mahajan, Prof. I. R. Shaikh, "Detect Covert Channels in TCP/IP Header using Naive Bayes," International Journal of Computer Science and Mobile Computing, vol 4, pp. 881-886, 2015.

[17] Daniel Svozil, Vladimir Kvasnicka, Jiří Pospíchal, "Introduction to multi-layer feed-forward neural networks," Chemometrics and Intelligent Laboratory Systems, vol. 39(1), pp. 43-62, 1997.

[18] Keiron O'Shea, Ryan Nash, "An Introduction to Convolutional Neural Networks." arXiv, arXiv:1511.08458.

[19] Sepp Hochreiter, Jürgen Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9(8), pp. 1735 - 1780, 1997.

[20] Packet traces from WIDE backbone. https://mawi.wide.ad.jp/mawi/.