A Multimode Network Steganography for Covert Wireless Communication Based on BitTorrent

Mingqian Wang, Weijie Gu, and Changshen Ma

School of Information Engineering, Changzhou Vocational Institute of Mechatronic Technology, Changzhou 213164, China

Correspondence should be addressed to Mingqian Wang; 937565385@qq.com

Received 11 March 2020; Revised 15 June 2020; Accepted 20 June 2020; Published 5 July 2020

Copyright © 2020 Mingqian Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Network steganography is a hidden communication technique, which utilizes the legitimate traffic as the vehicle to transfer the secret information covertly over the untrusted network. BitTorrent (BT) is one of the most prevalent P2P services for transmitting video files over wireless networks. An enormous amount of video data is transmitted over BitTorrent traffic continuously, to make it potentially available for confidential information transfer. Hence, in this paper, the BitTorrent file-sharing service of P2P is chosen as the host for information hiding, and a multimode steganographic method based on Bitfield message is proposed. Taking advantage of BitTorrent cooperative transmission and the non-content-authentication mechanism of Bitfield message, the secret information is delivered during the exchange process of BitMapInfo between two peers. The steganographic mode is dynamically selected in view of the secret size, achieving adaptive bandwidth. The experimental results show that our scheme can resist statistical-based detection effectively and outperform the existing method by obtaining a lower degree of detection rate under machine learning-based steganalysis.

1. Introduction

Data transmission security, such as information hiding, is an important field that has been discussed by many works [1, 2]. As a subfield of data transmission security, network steganography exploits the normal traffic as the carrier to transmit information stealthily via untrusted networks. Compared with the static multimedia steganography, such as image steganography [3], it is difficult for the monitor to locate and extract the covert data in tremendous network flow. Hence, network steganography is an effective means of transporting confidential information in networks. In recent years, it has become a hot research topic in the field of information security due to the fine properties of network traffic.

There are two broad types of network covert channels: covert storage channel and covert timing one. Covert storage channel embeds the secret information into the redundancies of network protocols [4–7]. Although it is simple and easy to implement, it can be easily detected by the existing methods. Covert timing channel delivers the secret information by exploiting time-relevant events of network packets and it has better stealthiness than the covert storage one. Generally, it can be divided into three subclasses: On-Off covert channel [8], interpacket delay- (IPD-) based covert channel [9–12], packet sorting [13, 14], and combination-based ones [15, 16]. Synchronization is always a difficult problem to solve since the covert timing channel is susceptible to the unstable network condition, such as jitter and delay. To guarantee the reliability, Archibald and Ghosal [17] designed a mechanism by using TCP ACKs to synchronize the covert channel. Houmansadr [18] and Archibald [19] used the Error Correction Code to encode the secret information to increase accuracy, which sacrificed the bandwidth of the covert channel and increase the transmission overhead. Countering the above deficiencies, network steganography tends to mimic the normal traffic by shape-fitting. The feature model is considered in the modulation process of the secret information to resist statistical detection tools. Predominantly, appropriate and
feasible network services with more popularity, reliability, and security are sought as a steganographic carrier.

It is revealed in the 2017 network traffic report that the P2P traffic accounts for 79.25%, 83.46%, 63.94%, and 67.19% of the total flow in Germany, Eastern Europe, Middle East, and Australia, respectively, which wins the top ranking. Among them, the BitTorrent (BT) file sharing service occupies more than 50% to 70% of P2P traffic. Additionally, the result of traffic monitor between the boundary route of Jiangsu Education Network and the national trunk route indicates that the BitTorrent traffic accounts for 60% of all, which has become the most prevailing P2P protocol.

As we know, an ideal steganographic carrier should possess two properties: popularity and complexity. Since the massive communication, traffic and complex pattern of such a carrier can improve the undetectability of steganography. Nowadays, wireless network has become a predominant means of data transmission, dynamically evolving network steganography subfield. Under this background, recent network steganography solutions exploit popular P2P services like file-sharing systems like BitTorrent, search features like Google Suggest, and multimedia and Voice-over-IP services like Skype and WeChat [20–24]. The continuous and large amount of video data transmission during BitTorrent wireless communication provides chances of launching steganography.

Kopiczko et al. [25] proposed a BitTorrent-based network steganographic method called StegTorrent. It was based on modifying the order of data packets in the peer-peer data exchange protocol, which can provide steganographic bandwidth of up to 270 b/s. However, the perturbation of inherent network noise such as jitter or packet loss may affect the order of certain packet on the receiver side. Meanwhile, ordering of packets with distinct IP inevitably altered the interpacket delays, which could be easily differentiated from the normal distribution of BitTorrent IPDs by the adversary. To overcome the drawbacks of the existing method, the steganographic procedure should retain a normal communication mode of network service. In order to improve undetectability, the modulation of secret information cannot generate abnormal traffic or properties. Therefore, in this paper, BitTorrent is selected as the steganographic carrier, of which the system structure, communication mode, and protocols are analyzed comprehensively. On this basis, the protocol redundancy is deeply tapped in which the secret information can be embedded. The main contribution of this paper is as follows:

(1) Taking advantage of the non-content-authentication mechanism of Bitfield message, the secret information is transferred during the exchange process of BitMapInfo (BitMap Information) between two legitimate BT peers (Clients) and is embedded into the content of bitfield according to the given format. Thus, the covert traffic of our scheme preserves normal behavior and property, so as to resist detection.

(2) The multimode steganographic method is proposed which exploits the cooperative transmission of BT peers. The steganographic mode (Single-Link Steg or Multi-Link Steg) is dynamically selected in view of the secret size, achieving adaptive bandwidth. In Multi-Link Steg, multiple peers participate in the transmission of secret information concurrently to accomplish collaborative steganography. Meanwhile, such method is noise-tolerated due to the reliable transmission mechanism of TCP, making the proposed scheme more robust.

(3) Multidimensional steganalysis is employed to comprehensively verify the undetectability of the proposed scheme. Statistical properties-based detection methods are used to measure the traffic regularity, such as Entropy test (EN-test). The non-parametric statistical method is exploited to measure the distance between two distributions, such as the Kolmogorov–Smirnov test (K-S test). Machine learning-based steganalysis is used to classify the tested traffic into covert or normal, such as SVM, Random Forest, and Deep Neural Networks.

The remainder of this paper is organized as follows. In Section 2, related works are reviewed. The basics of the BitTorrent system are described in Section 3. In Section 4, the proposed scheme is introduced in detail. In Section 5, experimental results are presented and analyzed. Finally, the whole paper is concluded in Section 6.

2. Related Work

Kopiczko et al. [25] proposed a BitTorrent-based network steganographic method called StegTorrent, which is illustrated in Figure 1. It is assumed that both the secret data sending and receiving sides are in control of a certain number of BitTorrent clients and, as mentioned above, their IP addresses are known to each other. In Figure 1, for the sake of clarity, only single direction steganographic transmission is presented, but of course, end-to-end bidirectional communication is possible and the other direction is analogous. No knowledge of the network’s topology is necessary. The hidden data sender uses the modified BitTorrent client—StegTorrent client—to share a resource that is downloaded by the second StegTorrent client that consists of a group of controlled BitTorrent clients.

For the sake of the proposed method’s description and analysis, the term data package is defined as a set of IP addresses that are sent within the IP packets in a predetermined order and the term data package size as the total number of elements in this set. For example, it is assumed that the data package size is 2. In this case, two packets with two different IP addresses (e.g., IP1 and IP2) are used to send bits of hidden data. In this simple scenario, if the order of the packets is modified for steganographic purposes and the BitTorrent client receives a packet that was sent from IP1 and then from IP2, then it will be interpreted as binary “0” and in other cases as binary “1.” It is assumed that the data package and its size are a shared secret between transmitting and receiving StegTorrent clients.
It must be noted that this method’s performance depends on the size of the data package while the latter relies on the number of available receiving IP addresses (receiving BitTorrent clients under control). However, the perturbation of inherent network noise such as jitter or packet loss may affect the order of certain packet on the receiver side. Meanwhile, ordering of packets with distinct IP inevitably altered the interpacket delays, which could be easily differentiated from the normal distribution of BitTorrent IPDs by the adversary.

3. BitTorrent Analysis

BitTorrent is a P2P file-sharing system that allows its users to distribute large files over networks. BitTorrent is distinguished from other similar file transfer applications in that instead of downloading a resource from a single central server, users download fragmented files from other users simultaneously. As a result, the file transfer time is considerably decreased because the group of users that share the same resource or part of it may consist of several to thousands of hosts. Such a group of users interested in the same resource known as “peers” combine together with a central component known as a “tracker” in BitTorrent. This combination of peers and trackers is called a “swarm.” Trackers are responsible for controlling the resource transfer between peers. Peers that hold onto a particular resource or part of a resource are required to share the resource and to perform the transfer.

There are two types of BitTorrent peers based on the stage at which they are involved in downloading or sharing a given resource:

1. Seeder: peers that possess the complete resource and are only sharing it.
2. Leecher: peers that do not possess the complete resource but are interested in doing so. They also share the fragments they have already downloaded. When a leecher obtains all the remaining fragments of the resource, it automatically becomes a seed.

In order to preserve the communication mode and properties of normal BT traffic during the steganographic process, it is essential to analyze the operation mode and protocols in the BT system. The concrete communication procedure of BT file sharing is shown in Figure 2:

1. A seed file (.torrent) is produced by the seeder and then released to the Tracker, which is combined with the web service.
2. Peer1 (Leecher) queries and downloads the seed file of the required resource from the Tracker.
3. Peer1 (Leecher) requests the list of peers which possess the shared resource from the Tracker.
4. The Tracker returns the corresponding peers-list to Peer1 (Leecher).
5. Peer1 (Leecher) conducts “three-way handshake” of TCP with the other peer, and then the connection between them is established.
6. Once Peer1 (Leecher) is connected with Peer2 successfully, they will immediately send and reply the Handshake messages, in order to confirm their identities.
7. Peer1 (Leecher) exchanges the Bitfield message with Peer2, informing each other of the indexes of file fragments which are already owned by themselves.
8. Peer1 (Leecher) exchanges a series of negotiation messages with Peer2, such as choke, unchoke, interested, and not interested.
9. Peer1 (Leecher) sends the Request message to Peer2, asking for the specific file fragments.
10. Peer2 replies the Piece message to Peer1 (Leecher), containing the corresponding file fragments.

Among them, the Bitfield message is used to indicate the bitmap information of certain file fragments, which have already been obtained by the current peer. In the BT client, a file is generally divided into several fragments whose size is 256 kB. Then, the fragments are indexed from 0 in sequence. Since the number of fragments is
distinct for each file, the length of the Bitfield message is variable.

The format of the Bitfield message is shown in Figure 3, where \( len \) refers to the length of the Bitfield message, which occupies 4 Bytes. And \( id \) is the identifier of Bitfield message, of which the value is set to 5 occupying 1 Byte. The \(<\text{bitfield}>\) of \( X \) Bytes indicates the possession of specific file fragments, as depicted in Figure 4. The fragment with index 0 corresponds to the highest bit of the first byte, and so on. If the bit is “1” in the position, it is revealed that the corresponding fragment is possessed while a bit “0” means that certain fragment is not possessed by the peer.

It can be observed that the Bitfield message is only sent after completing a “handshake” immediately. Since there is no content-authentication mechanism of Bitfield message in BT client, the modification of \(<\text{bitfield}>\) may not arise abnormally. In other words, the altered \(<\text{bitfield}>\) will default as the original content. Although the Bitfield message is only exchanged for once during the single interaction of two peers, the size of the delivered data is considerable. Therefore, the Bitfield message is employed as the steganographic carrier in this paper.

4. The Proposed Scheme

4.1. System Model. The proposed steganographic system model is presented in Figure 5. The steganographic peers include steganographic sender and receiver, which disguise as the legitimate BT clients. The open-source code of the BT client is modified according to the proposed scheme, which is implemented as follows:

(1) Steg-preparing: first, the steganographic mode (Single-Link Steg or Multi-Link Steg) is selected by the sender-peer in accordance with the secret size. And a suitable video file is chosen as a shared resource. Second, the critical information of the shared video file, such as file name and format, is delivered to the receiver-peer via e-mail, instant messaging, and so on.

(2) Normal BitTorrent communication/before-steg: the steganographic peers request the common file resource from the Tracker and establish TCP link with each other.

(3) Steg-synchronization: the steganographic peers exchange the Handshake message to authenticate their identities in covert communication.

(4) Steg-implementation: the sender-peer embeds the secret information into the Bitfield message according to the selected steganographic mode. Then, the altered Bitfield message is sent to the receiver-peer, from which the secret information can be extracted.

(5) Normal BitTorrent communication/after-steg: after accomplishing the transmission of secret information, the steganographic peers still exchange the negotiation messages and transfer the required video file fragments, as the other normal BT peers.

4.2. Multimode Steganography. In BT communication, two peers only exchange Bitfield message for once during the entire process of video file transfer, in order to share the concrete bitmap information of themselves. As mentioned above, the bitmap information is used to inform the other
peer which file fragments have been possessed by one peer and is sent after completing "handshake" immediately. Hence, the secret data that can be transferred is limited during the single interaction of two peers. If more secret data is required to be delivered, multiple peers might be employed in sharing the common resource. Multipeers participate in the transmission of secret information concurrently to accomplish cooperative steganography. Accordingly, there are two proposed steganographic modes based on the data size of secret information: Single-Link Steg and Multi-Link Steg. The main notations and symbols of our scheme are presented in Table 1.

**Table 1: The main notations and symbols.**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Link Steg</td>
<td>Single-link steganography mode</td>
</tr>
<tr>
<td>Multi-Link Steg</td>
<td>Multi-link steganography mode</td>
</tr>
<tr>
<td>Mode</td>
<td>Steganography mode</td>
</tr>
<tr>
<td>(S_{len})</td>
<td>The length of secret information</td>
</tr>
<tr>
<td>Secret_info</td>
<td>The content of secret information</td>
</tr>
<tr>
<td>Padding</td>
<td>The remainder of Bitfield field</td>
</tr>
<tr>
<td>File</td>
<td>The shared video file</td>
</tr>
<tr>
<td>(\text{sizeof})</td>
<td>Function of calculating the shared file size</td>
</tr>
<tr>
<td>Index</td>
<td>The index of secret data block</td>
</tr>
<tr>
<td>(S_{\text{block}}(i))</td>
<td>The (i)-th secret data block</td>
</tr>
</tbody>
</table>
4.2.1. Single-Link Steg. The Single-Link Steg mode is suitable for transmitting less secret information such as key and parameter. In this scenario, there are only two peers participating in covert communication. As mentioned above, the steganographic sender must be a seeder. The Single-Link Steg is implemented as follows:

Step 1. Bitmap Info <bitfield> is partitioned into four steganographic fields, as shown in Figure 6. Assume that the length of <bitfield> is X Bytes. The meaning of each field is illustrated as follows:

(i) Mode refers to the steganographic mode, which occupies 1 Byte. When this value is set to “0,” it is denoted that our steganography is working in Single-Link state

(ii) S_len refers to the length of secret information, which occupies 1 Byte. And it is defined as L Bytes

(iii) Secret_info refers to the content of secret information, whose size is L Bytes

(iv) Padding refers to the remaining original content of <bitfield> after the substitution, whose size is (X-L-2) Bytes. And it should be satisfied that L + 2 ≤ X.

Step 2. The original <bitfield> is substituted with the secret information according to the aforementioned steganographic format. In addition, the shared video file between steganographic peers must be appropriately selected in accordance with the secret size L. In particular, the size of the video file should satisfy the certain requirement, as denoted in

\[
\text{sizeof (File) } \geq \lceil (L + 2) \times 8 \times 256 \rceil / 10^6 \text{ GB}, \quad (1)
\]

where sizeof is represented as the function of calculating the file size. The video file is generally divided into several fragments whose size is 256 kB.

4.2.2. Multi-Link Steg. In order not to disrupt the legitimate BT communication of file sharing, when it is necessary to transfer a larger amount of secret data, the steganographic peers are not allowed to send Bitfield message several times. Thus, the Multi-Link Steg mode is exploited in case that more secret information is required to deliver. Cooperative steganography can be realized by the collaborative transfer of multiple BT peers. In this scenario, the steganographic peers disguise as the legitimate BT clients intended to download the common video resource. They collaborate to transfer the secret segments in accordance with prior careful planning. The Multi-Link Steg is implemented as follows, which is shown in Figure 7.

Step 1. Bitmap Info <bitfield> is partitioned into five steganographic fields, as shown in Figure 8. Assume that the length of <bitfield> is X Bytes. The meaning of each field is illustrated as follows:

(i) Mode refers to the steganographic mode, which occupies 1 Byte. When this value is set to “1,” it is denoted that our steganography is working in Multi-Link state

(ii) S_len refers to the length of the secret block, which occupies 1 Byte. And it is defined as L Bytes

(iii) Index refers to the index of the secret block, which initiates from 1

(iv) S_block refers to the content of the secret block, whose size is L Bytes

(v) Padding refers to the remaining original content of <bitfield> after the substitution, whose size is (X-L-3) Bytes. And it should be satisfied that L + 3 ≤ X.

Step 2. The secret information is divided into n blocks, whose size is L. S_block (i) refers to the i-th secret data block, where i = 1, 2, . . . , n.

Step 3. n peers (legitimate BT clients) are controlled by the steganographic sender to transfer the secret blocks collaboratively.

Step 4. The sender-peers then connected with the steganographic receiver, respectively, establishing n covert links.

Step 5. For each sender-peer, the original <bitfield> is substituted with the secret block according to the aforementioned steganographic format.

Step 6. The steganographic receiver extracts the secret blocks according to the agreed format. Then, the blocks are reordered to retrieve the complete secret information, which is denoted as secret_info as follows:

\[
\text{secret_info} = \sum_{i=1}^{n} s_{\text{block}} (i). \quad (2)
\]

5. Experiment Results and Analysis

5.1. Data Set and Implementation. Single-Link Steg and Multi-Link Steg are realized in the experiment, respectively. The open-source BT clients are modified to implement the proposed scheme, delivering the secret information covertly. Under the Single-Link Steg mode, steganographic receiver disguises as the BT seeder. The data size of secret information is 255 bytes and the shared video file is selected whose size is 104 MB. The communication packets between the steganographic peers are captured by Wireshark, as shown in Figure 9. It can be seen that the secret data is transferred successfully by format substituting the partial content of the Bitfield message. Besides, it is verified that the legitimate BT communication has not been affected by the revision of the Bitfield message. The negotiation messages such as Interests and Unchoke are exchanged subsequently and so are the file fragment transmission messages such as Request and Piece. In that, it can be concluded that the proposed steganography retains normal communication without introducing any additional anomaly.

Under the Multi-Link Steg mode, the data size of secret information is 1 kB, and the shared video file is selected whose size is 90 MB. In this scenario, there are three steganographic peers involving in the covert communication, in which peer1 and peer2 are all controlled by the steganographic sender, in order to cooperatively transfer the secret data. Peer3 is the steganographic sender, which acts as the
Further experiments are performed to evaluate the main performance metrics of the proposed scheme, which contain the undetectability, robustness, and capacity analysis. As the
undetectability and robustness will not be affected by the number of steganographic peers, only the mode of Single-Link Steg is considered in the corresponding experiment.

5.2. Undetectability. As the core property, undetectability refers to the covert traffic that cannot be differentiated from the normal one, which is all depended on the similarity between the two. Therefore, in order to improve undetectability, the modulation of secret information cannot generate abnormal traffic or properties. In the experiment, normal traffic of downloading general video files in BT clients (BitTorrent, µTorrent, and Vuze) is captured by Wireshark. Then, the lengths of \(<\text{bitfield}>\) in bitfield messages are extracted to form the normal samples. The number of normal and steganography samples is 20,000. In the following, statistical and machine learning-based steganalysis is utilized to detect our proposed scheme, respectively.

5.2.1. Statistical-Based Steganalysis. Statistical-based steganalysis is the most common and popular method to detect the potential covert traffic, in which statistical properties such as traffic regularity or distribution function are exploited to distinguish the normal and covert traffic. As we know, the histogram is a significant property that can reveal the statistical distribution feature of traffic. Therefore, the histograms of normal and covert traffic of our scheme are compared in Figure 11, where the x-axis shows the field length of \(<\text{bitfield}>\) ranging from 0 to 2500 Bytes and the y-axis indicates the number of lengths that occurred within each bin (the x-axis is divided into eight bins). As shown in the figure, the field length of normal \(<\text{bitfield}>\) occurs most between 800 and 1200 Bytes, with a peak value of 1000 Bytes. It is obvious that the histogram of our scheme matches the normal one quiet well. The file size which is calculated is approximately 4.9 GB corresponding to the maximum \(<\text{bitfield}>\) length of 2500 Bytes.

Meanwhile, two notable detection methods are employed to reckon the detection resistance of our scheme compared with StegTorrent [25] quantitatively, which are the Entropy test [26] and Kolmogorov–Smirnov test [27]. For normal and covert samples, they are both divided into 20 consecutive windows whose size is 1000. Certain statistical feature of each window is calculated and used during the detection process, as depicted in Figure 12.

(1) Entropy Test. Entropy can describe the degree of chaos in a process. In the Entropy test (EN-test), it is utilized to measure the regularity of data traffic [26]. If the traffic is less regular, the Entropy value will be larger, and vice versa. Since the less regularity indicates more randomness, the more amount of information is contained in the traffic. The Entropy value is obtained by calculating the statistical average of all possible self-information, which is denoted in

$$H(X) = E[I(x_i)] = - \sum_{i=1}^{n} p(x_i) \log p(x_i),$$  \hspace{1cm} (3)

where $X$ represents a one-dimensional discrete random variable, whose set of values is $\Omega = \{x_i \mid i = 1, 2, \ldots, n\}$. The
self-information of $x_i$ is $I(x_i)$ and the probability of $x_i$ is denoted as $p(x_i) = P(X = x_i)$. The Entropy values of 20 windows for normal and covert samples are compared in Figure 13. From the result, it can be found that most Entropy values of normal samples range approximately from 0.5 to 1.3, whereas those of the covert samples generated by StegTorrent are from 0.8 to 1.5. But the values of our scheme mix with those of the normal samples, which can hardly be differentiated.

Then, 20 windows of normal and covert samples are tested using the Entropy test, respectively, when the window size is 1000. The results are presented in Table 2, where the detection threshold is denoted as $THD$. It is observed that the false-negative rate of normal samples declines when the threshold increases. Meanwhile, the detection rates (true positive rates) of covert samples are shown in the table. And we can see, the detection rate of StegTorrent ranges from 91% to 98%, while that of our scheme is only below 7%. Hence, the Entropy test fails to distinguish the covert samples of our scheme from the normal one.

(2) Kolmogorov–Smirnov Test. K-S test [27] measures the maximum distance between two distributions. A small value indicates that two distributions are close to each other. Conversely, a large value means that one distribution does not fit the other one. The Kolmogorov-Smirnov test value (KS-test value) is attained by taking the supremum of the absolute difference between two empirical distribution functions for all $x$, which can be defined in

$$KSTEST = \sup|S_1(x) - S_2(x)|,$$

where $S_1(x)$ and $S_2(x)$ refer to the empirical distribution functions of two samples. The comparison of KS-test values between the normal and covert samples is shown in Figure 14. Likewise, 20 windows of normal and covert samples are tested in the experiment. The $x$-axis is the window number and the $y$-axis shows the corresponding KS-test value. It is observed that the KS-test values of our scheme are under 0.15, confused with those of the normal traffic. Thus, the distribution of our scheme is close to that of the normal one. Nevertheless, the corresponding values of StegTorrent occur from 0.15 to 0.25, which is deviated from the normal case.

Then, the covert traffic is detected using the K-S test and the detection results are shown in Table 3, where the detection threshold is denoted by THD. It is observed that the false negative (FN) rate of the normal traffic declines when the threshold increases. FN refers to the normal sample which is misclassified as the covert one. Hence, the detection threshold is set appropriately from 0.13 to 0.15, in order to guarantee that the false-negative rate remains under 1%. Meanwhile, the true positive (TP) rates of covert samples are presented in the table. In this paper, the detection rate is represented by TP. From the results, it is easily seen that the detection rate of StegTorrent is more than 92% when tested with different thresholds. But in our case, it is located under 3%, indicating that the Kolmogorov–Smirnov test cannot effectively detect the covert traffic generated by our scheme.
5.2.2. Machine Learning-Based Steganalysis. Recently, the machine learning technique performs quite well in resolving complex problems in various domains. In particular, it has progressively become a novel and effective means of detecting covert channels. In machine learning-based steganalysis, various statistical metrics (features) of normal and covert samples are utilized by classifier models and eventually be trained to distinguish covert traffic. The classifiers used in machine learning-based detection mainly include SVM, Neural Network, Logistic Regression, Naive Bayes, Random Forest, and Deep Neural Network [28–30]. In this paper, Deep Neural Network (DNN) is principally employed to further estimate the undetectability of our scheme compared with StegTorrent.

(1) Detection Process. The proposed scheme is detected using DNN by the following steps, as depicted in Figure 15.

Step 1. Network traffic of downloading general video files in BT clients is captured by Wireshark. Then, the lengths of \textit{<bitfield>} are extracted to form the normal or covert samples, whose size is 5,000,000, respectively. The samples are divided into 10,000 subsamples, each containing 500 lengths.

Step 2. For each subsample, values of five statistical features, including mean, median, entropy, standard deviation, and root of average mean error are calculated as described in Table 4. The data set of statistical features contains two types of samples, which are the normal and covert ones. It will be then used for training or testing in the classifier.

Step 3. The data set is divided into two parts, 70% of which is used for training in the DNN classifier model and 30% of which is used for testing. The normal traffic is labeled “0” and the covert one is labeled “1.” After training the DNN classifier, it can be exploited to detect the covert traffic online.

The structure of DNN is shown in Figure 16. In the input layers, 5 statistical features are fed to DNN as the input variables. In the hidden layers, each layer consists of a number of neurons involved in the prediction phase. Each neuron adjusts its weight based on the learning process and participates in calculating the coefficients of the final equations, which will be used to determine the class label (normal or overt) of tested samples. The output layer is responsible for determining the predicted value of the class label.

(2) Detection Result. Figure 17 depicts the effect on the detection rate of covert samples when increasing the number of neurons inside the DNN hidden layers. It can be noted that the detection rate improves as the number of neurons increases until it reaches 13, where the highest rate of 37% is achieved in detecting our proposed scheme. Nevertheless, at most 96% of StegTorrent is differentiated successfully by the DNN classifier.

Subsequently, the effect on the detection rate of increasing the number of hidden layers in DNN is shown in Figure 18. It is observed that the detection rate also increases as the increment of hidden layers until reaching a certain level. And the rate declines after the peak value since the classifier model is overfitted. It is easily found that 43% of covert samples of our scheme are detected when the number of hidden layers is 5 while the detection rate of StegTorrent reaches above 97% under the same circumstances.

Finally, the proposed scheme is tested by other machine learning-based detection methods, such as SVM, Logistic Regression, Naive Bayes, Random Forest. And the detection rates of our scheme and StegTorrent are compared in Figure 19. It is observed that 24% to 43% of our scheme is detected by different classifiers, while the detection rates of StegTorrent appear from 92% to 98%. It is clearly noticeable that the proposed scheme has outperformed StegTorrent by obtaining a lower degree of detection rate. Therefore, it can be concluded that our scheme possesses better undetectability than the existing method.

5.3. Robustness. Robustness requires the covert channel to keep working with relatively high accuracy and low bit error rate (BER), resisting the perturbation of network noise such as network jitter and packet disorder and loss. In the experiment, the robustness of our proposed scheme is measured considering packet loss (pl) and packet disorder (pd). The BERs of the proposed scheme are compared with those of StegTorrent in terms of different rates of packet disorder/
loss, as given in Figure 20. It is obvious that the secret information about our scheme can be accurately obtained under different rates of packet loss or disorder. However, the BER of StegTorrent increases with the increment of packet loss/disorder rate. The BER of StegTorrent reaches up to 11% when 20% of packets are lost, which will degrade the reliability of covert communication in StegTorrent.

On the one hand, the good performance in resisting packet loss and disorder of our scheme is due to the TCP reliable transmission mechanism of normal BT traffic, which serves as the carrier of our steganography. Therefore, the proposed method is noise-tolerated. On the other hand, packet loss or disorder alters the packet-arriving order in StegTorrent, which will lead to the misrecovery of secret data on the receiver side. Hence, we can conclude that our scheme is superior to StegTorrent in respect to robustness.

5.4. Capacity. Capacity is the maximum data size that can be reliably transmitted over the covert channel per second or packet. In other words, capacity refers to the transfer rate of secret information. It is closely related to the bandwidth of normal carrier and the steganographic modulation algorithms. As revealed in Figure 21, the field length of $<\text{bitfield}>$ ranges from 0 to 2500 Bytes in normal BT communication,
which means that the maximum capacity of Single-Link Steg is 2500 B/P. Meanwhile, in Multi-Link Steg, the capacity will increase linearly with the number of steganographic peers, which is shown in Figure 21. Since the field length of normal <bitfield> occurs most between 800 and 1200 Bytes as mentioned above, the secret data of a certain size ($L$) is transmitted by each peer engaged in the steganography. It is found that when 64 peers transfer the secret information concurrently, the capacity reaches up to 76,800 B/P.

However, more peers might increase the overhead of system resources and the complexity of the steganographic control mechanism, which will make the scheme more difficult to implement. Thus, the tradeoff between the number of steganographic peers and system overhead will be taken into consideration in future research. And then, the capacity of Multi-Link Steg mode can be analyzed under the optimal number of steganographic peers.

6. Conclusions

Bit Torrent file sharing the protocol of P2P is a steganographic carrier with high covertness, which has massive network traffic and complex communication mechanism. The steganographic peers are confused with numerous legitimate BT peers, owing to the cooperative transmission in the P2P network. Thus, it is extremely difficult to locate steganographic peers in the tremendous BT traffic. The steganographic peers disguise as the legitimate BT clients, who are interested in possessing the common video file. They participate in downloading the same resource following the normal BT communication mode, without introducing any
extra traffic. Taking advantage of the non-content-authentication mechanism of Bitfield message, the secret information is embedded into the content of $<\text{bitfield}>$ according to the given format. The altered Bitfield message can bypass the security censorship of the BT system and network monitor device. Hence, our scheme has proved better undetectability and robustness than the current methods. In the future work, another BitTorrent-based steganographic algorithm will be designed and researched, in which the tradeoff between the numbers of steganographic peers and system overhead will be taken into consideration. And then, the optimal steganographic mode can be analyzed and selected.

**Data Availability**

The software code and data used to support the findings of this study are available from the corresponding author upon request.

**Conflicts of Interest**

All authors declare that there are no conflicts of interest regarding the publication of this paper.

**Acknowledgments**

This work was supported by the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province under Grant no. 19KJB5100019, Innovation and Entrepreneurship Training Program for College Students of Jiangsu Province under Grant no. 201913114004Y, Changzhou Key Laboratory of Industrial Internet and Data Intelligence under Grant no. CM20183002, and the Project of Changzhou Vocational Institute of Mechatronic Technology under Grant no. 2019-YBKJ-05.

**References**


