Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi
Afyon Kocatepe University – Journal of Science and Engineering
https://dergipark.org.tr/tr/pub/akufemubid

Araştırma Makalesi / Research Article
DOI: https://doi.org/10.35414/akufemubid.1427644
AKÜ FEMÜBİD 24 (2024) 055102 (1138-1149)
AKU J. Sci. Eng. 24 (2024) 055102 (1138-1149)

HGNN: A Hybrid Graph Neural Network Based on Transfer Learning for Linguistic Steganalysis

*Makale Bilgisi / Article Info
Alındı/Received: 31.01.2024
Kabul/Accepted: 29.06.2024
Yayımlandı/Published: 01.10.2024

HGNN: Dilsel Steganaliz için Transfer Öğrenimine Dayalı Hibrit Grafik Sinir Ağı

Merve VAROL ARISOY*

Burdur Mehmet Akif Ersoy Üniversitesi, Bucak Bilgisayar ve Bilişim Fakültesi, Bilişim Sistemleri Mühendisliği Bölümü, Bucak-Burdur, Türkiye

Abstract

Steganography, especially in the form of text generation based on secret messages, has become a current research topic. It is more difficult to identify the hidden message when it embedded di-rectly into the text without using a cover text, and it also has a higher embedding capacity. Owing to the high rate of imperceptibility and resistance to steganalysis of this type steganography, it is essential that steganalysis methods, generate better performance. Although the complexity of deep learning models increases the accuracy rate, it also increases the inference time. In this study, a linguistic steganalysis was performed with a lower inference time and a higher accuracy rate. In the developed model, first, differences between non-stega and steganographic texts were mod-elled by a finetuned Bert using the custom dataset. The disparity information obtained by fine-tuned model was distilled into 3 separate networks, BertGCN, BertGAT and BertGIN, for faster and more accurate inference. Then, these 3 distilled networks were combined through Transfer Learning to form a new model. Experiments demonstrates that the proposed model surpass other methods in terms of the accuracy (a success of 0.9879 at 3.22 bpw on text encoded through SAAC Encoding) and the effectiveness of inference (1.09 second).

Keywords : *Knowledge distillation; Llinguistic steganalysis; Transfer learning; GAT (Graph Attention Network); GCN (Graph Convolutional Networks); GIN (Graph Isomor-phism Network.*

Öz

Özellikle gizli mesajlara dayalı metin üretimi şeklindeki steganografi güncel bir araştırma konusu haline gelmiştir. Gizli mesajın kapak metni kullanılmadan doğrudan metnin içine gömülmesi durumunda tespit edilmesi daha zor olduğu gibi gömme kapasitesi de daha yüksektir. Bu tip steganografinin algılanamazlık oranının yüksek olması ve steganalize karşı direnci nedeniyle, steganaliz yöntemlerinin yüksek performans üretmesi önemlidir. Derin öğrenme modellerinin karmaşıklığı doğruluk oranını arttırsa da çıkarım süresini de arttırmaktadır. Bu çalışmada, daha düşük çıkarım süresi ve daha yüksek doğruluk oranıyla dilsel steganaliz gerçekleştirilmiştir. Geliştirilen modelde öncelikle stega olmayan ve steganografik metinler arasındaki farklar, özel veri seti kullanılarak hassas ayarlı Bert (Bidirectional Encoder Representations from Transformers) tarafından modellendi. İnce ayarlı modelle elde edilen eşitsizlik bilgisi, daha hızlı ve daha doğru çıkarım için BertGCN (Bert Graph Convolutional Network), BertGAT (Bert Graph Attention Network) ve BertGIN (Bert Graph Isomorphism Network) olmak üzere 3 ayrı ağa ayrıştırıldı. Daha sonra bu 3 damıtılmış ağ, Transfer Öğrenme yoluyla birleştirildi ve yeni bir model oluşturuldu. Deneyler, önerilen modelin doğruluk (SAAC Kodlama yoluyla kodlanan metinde 3,22 bpw'de 0,9879 başarı) ve çıkarımın etkinliği (1,09 saniye) açısından diğer yöntemleri geride bıraktığını göstermektedir.

Anahtar Kelimeler: *Bilgi damıtma; Dilsel steganaliz; Öğrenme aktarımı; GAT (Graf Dikkat Ağı); GCN (Graf Evrişimli Ağlar); GIN (Graf İzomorfizm Ağı).*

1. Introduction

The realization that a hidden message can be embedded in a text document and that this new version of the document carrying a message can be transferred seamlessly over normal channels has recently led to the field of steganography becoming a research topic that has drawn intense attention. Textual information can be transferred over platforms with high frequency of use in daily life, such as social media tools, blog pages, emails, etc., quickly and effortlessly. While steganography is a useful method when it comes to protecting and securely transferring information that is not intended to be accessed by malicious people, it is known that this method becomes completely harmful when it is used by terrorists, hackers, and those who carry out illegal activities to transfer suspicious messages. At this point, we encounter "steganalysis", which is the opposite of steganography. Steganalysis classifies whether a text document contains a secret message or not.

In steganography, it is known that the generation-based steganography method, which can embed messages at a high rate and is more resistant to steganalysis in terms of

*Sorumlu Yazar/Corresponding Author: Merve VAROL ARISOY e-posta/e-mail:mvarisoy@mehmetakif.edu.tr

imperceptibility, gives more effective results (Yang et. al 2019 (a), Kang et. al 2020, Fang et. al 2017). Hence, the steganalysis models should have a high detection rate on texts containing hidden messages created through generation-based steganography in order to be regarded successful. Most of the work in the field of steganalysis to date is ML (Machine Learning) based (Xiang et. al 2018, Chen et. al 2011, Meng et. al 2010). Steganalysis studies using ML methods examine the changes in some statistical features such as word frequency (Yang and Cao 2010, Xiang et. al 2014), word occurrence probability (Meng et. al 2009), content relevance (Chen et. al 2011). However, such steganalysis methods tend to detect only stego texts created using ste-ganography algorithms designed to generate statistical variations. Therefore, there is no universality of these ML-based methods against all steganography methods, especially due to their inability to detect steganographic texts that are produced based on hidden messages.

DL (Deep learning) techniques have been adapted to steganalysis by researchers working in the opposing field of steganography due to the recent focus on the high embedding capacity and imperceptibility rates of DL-based steganography algorithms that generate text based on hidden messages. In this context, many methods such as, local word level relevance (Yang et. al 2019 (b), Yang et. al 2020), global-level knowledge sharing between words (Wu et. al 2021), usage of one-dimensional hidden property (Yang et. al 2019 (b), Yang et. al 2019 (c), Zou et. al 2020), multidimensional hidden property representation (Yang et. al 2020, Niu et. al 2019), leveraging isolated in-text semantic features (Zou et. al 2020), semantic and syntactic features (Yang, J et. al 2021) have been applied in DL-based steganalysis studies.

One of the studies where DL algorithms are applied in the field of steganalysis is in (Wen et. al 2019). They suggested a convolutional neural network-based text ste-ganalysis model that can automatically learn feature representations from texts and capture complicated dependencies. In (Yang et. al 2019 (c)), they discovered that the conditional probability distributions of words are modified after hidden information is inserted into a text. In order to extract the differences in feature distribution, they created a model using recurrent neural networks. Based on the recovered feature data, they then categorised a text as cover or stego text. A hybrid steganalysis model utilizing CNN (Convolutional Neural Network) and BiLSTM (Bidirectional Long Short Term Memory) is reported in (Niu et. al 2019). To improve the detection accuracy of their model, they prioritized learning both local features and long-term semantic information of the text. In addition, there have been studies aiming to find out the relationship between words and the texts in which these words take place and to make a classification based on this. In the first of these studies, after adding a secret information to a text, it was tried to detect that the correlation between words of that text was broken, with a model based on CSW (Convolutional Sliding Windows) (Yang et. al 2020). In the other study, it is mentioned that there is a relationship between the words of the text and the text that is sensitive to overt and hidden steganography, and to extract this relationship a method called as Explicit and Latent Text Word Relation Mining is mentioned (Li et. al 2022).

The work in (Yang, H et. al 2020), a feature pyramid that combines basic text properties and a densely linked LSTM network are combined to suggest a neural linguistic steganalysis scheme. The developed models has become more and more comprehensive in order to extract the high-level properties of the text by delving deeper, which has resulted in computing processes becoming more challenging and inference times increasing. In order to provide a solution to these disadvantages, in the study of (Peng et. al 2021) a multi-stage strategy based on transfer learning to circumvent time-consuming computations and produce more effective inference is de-scribed. In their model, they distilled the steganographic text information obtained by a finetuned Bert model into LSTM and CNN networks separately. The semantic knowledge obtained from each of these networks was transferred to the new network formed by the combination of the 2 networks by transfer learning method. This reduces the time spent on inference and calculation.

In linguistic steganalysis studies, the words are usually given to the model in the form of a sequence and training is provided through this sequence (Yang et. al 2020, Niu et. al 2019, Li et. al 2022). Although LSTM models are suitable for long-term learning, the connection between words in the text should not only be considered between words that are next to each other but also the connection between words at different points in the text should also be considered. At this point, graph-based networks have started to attract attention. There are some studies in the literature that address this issue. In contrast to sequence-based linguistic steganalysis methods, a graph updater model made up of GGNN (Gated Graph Neural Network) layers was mentioned in the steganalysis study in (Fu et. al 2022) to extract the properties of word nodes. They employed graph channel attention learning to determine the critical di-mensions of the nodes in the graph and used the graph attention module as a graph updater to

concentrate on the text's keywords. In (Yang, J. et. al 2021), they describe a steganalysis framework that integrates semantic and syntactic features simultaneously. They implemented a language model with transformer architecture as a semantic feature extractor. They used the GAT network to identify syntactic features. Since graph-based networks may more effectively capture the relationship between words in a text than sequence-based networks, the graph approach was also chosen for this study.

When the studies in the literature are evaluated, it is known that steganalysis methods, especially DL-based ones, are universal and give more successful results than traditional methods in detecting texts created with generation-based steganography based on hidden messages. However, this increased success leads to an increase in the complexity and computational cost of the DL model. Therefore, the steganalysis process becomes difficult to implement in practice.

At this point, it has been noticed that in the steganalysis studies in the literature, graph-based networks have not been used by combining them through transfer learning in detecting the texts produced through secret messages and providing an acceptable performance in terms of inference time. Therefore, in this study, a new model based on graph-based and multi-stage transfer learning method, which is a combination of GCN, GAT and GIN networks, is developed. Prior to creating the model, a finetuned BERT model was used to model the distinctions between regular texts and stega texts. Then, the feature information of the finetuned Bert model is distilled into the BertGCN, BertGAT and BertGIN networks separately for fast inference. In the last step, the semantic features obtained from the previous step are combined into a new network structure consisting of a combination of GCN, GAT and GIN networks. Inspired by the work presented in (Peng et. al 2021), the work presented here is based on a graph-based structure that not only reduces computational effort, but also captures the semantic relationship between adjacent words, as well as the correlation between each word of the text. In this way, instead of only extracting the relationship between side by side word sequences, as implemented in (Peng et. al 2021), the level of relationship between all words is captured, which allows for a more detailed analysis of the text. The work here shares commonalities with the work in (Peng et. al 2021) at using only transfer learning to reduce computational effort and enable more efficient inference. However, the network structure created here is completely different from the work in (Peng et. al 2021). The general framework of the model is given in Figure 1 and the details are provided in section 3. The benefits of the study to the literature are listed below.

• Finetuning the BERT model for text classification on a dataset with a heter-ogeneous graph structure consisting of stega and normal texts

• Separately distillation into the GCN, GAT, and GIN networks from the finetuned Bert model carrying stega text feature data. Thus, stega-normal text detection by each of these three networks based on the weights obtained by BERT

• Reducing computational time and increasing inference efficiency by transferring the text classification information obtained by BertGCN, BertGAT and BertGIN networks to the new model that combines these three networks through transfer learning

The rest of the paper is organized as follows: Section 2 presents a review of the literature on text steganalysis. Section 3 discusses the details of the experimental setups and proposed model. Results and discussions are presented in Section 4. Section 5 presents the general conclusion of the study.

2. Materials and Methods

There are three distinct stages in the text stage analysis model. The dataset produced for this work, which contains both normal and stega texts, was used to train a pre-trained BERT model in the first step of the model to capture feature differences between normal and stega texts. The second stage is the information distillation stage, in which the weight information of the finetuned BERT model is distilled separately to the GCN, GAT and GIN networks, in other words, inference is attempted based on the weights obtained by the BERT model. In the last stage, a new network model was created by combining the weights obtained from the GCN, GAT and GIN networks in the previous stage under a single roof. The aim of combining these three networks is to maximize the performance of stega text detection.

2.1. Dataset collection

The dataset of stega/non-stega texts used in this study is based on the encoding methods used in (Shen et. al 2020). Linguistic steganography methods applied in the creation of stega texts are AC (Arithmetic Coding), Huffman Coding, SAAC (Self Adjusting Arithmetic Coding) and Bin-LM (Block Based Coding). In addition to the plain texts in (Shen et. al 2020), which contains no hidden information (where plain texts are categorized into 4 categories), various plain texts were also taken from the web environment. With the inclusion of these texts, a corpus

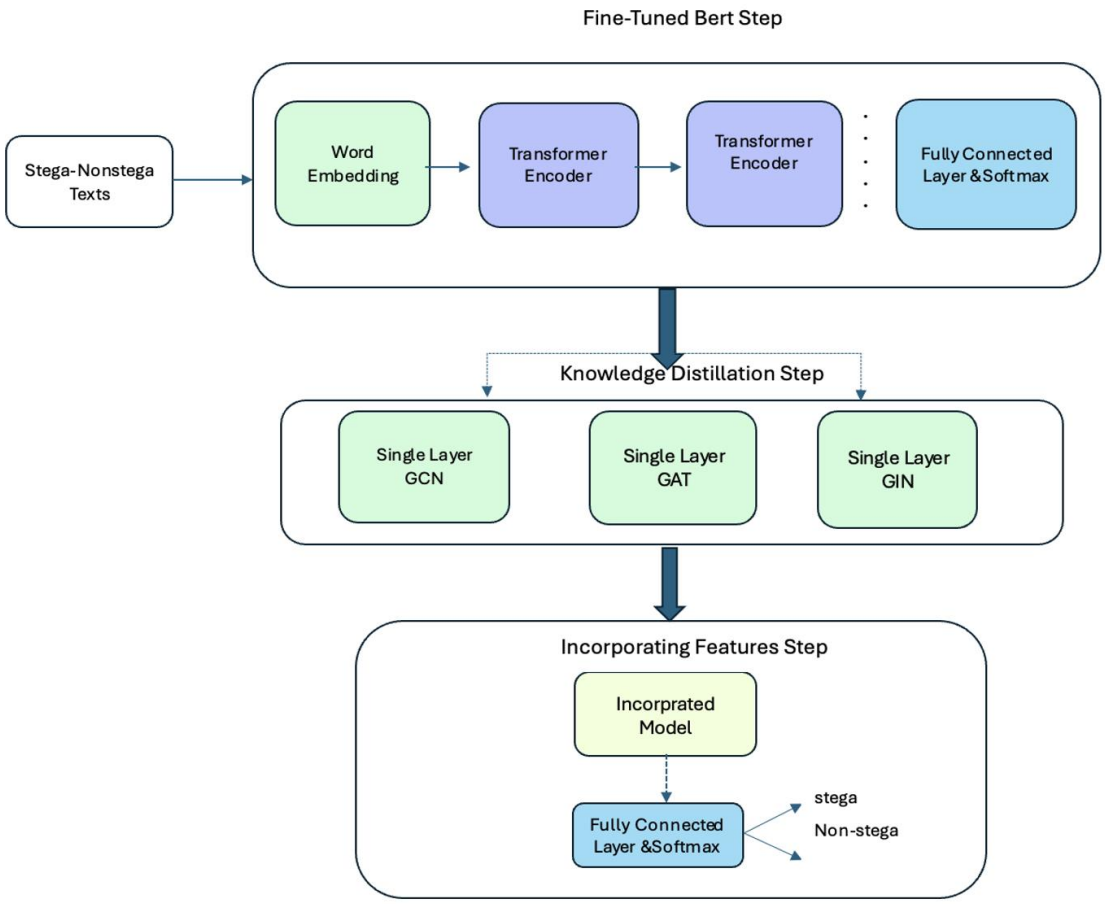

Figure 1. The general framework of the proposed text steganalysis method based on transfer learning.

containing non-stega texts in 5 different categories was created. Then these corpus were transformed into stega texts with the usage of AC, Huffman,Saac, Bin-LM methods. As a result, a corpus structure including both stega and nonstega texts was reached.

For each of the non-stega texts in 5 different subject categories in this study, 4 different encoding methods were used during stega text generation. In other words, 4 separate stega texts were obtained for any category of plain text. This process was applied to non-stega texts in all categories. The stega texts obtained by each of the 4 encoding methods are included in the dataset used here. In other words, in each of the 5 subject categories, there is 1 nonstega document and 4 stega documents. During the creation of the entire corpus, an equal number of non-stega documents (50 sentences from each) were taken from each subject category, creating a corpus of 250 nonstega sentences in total for 5 subject categories. During the addition of stega texts to the whole corpus, 15 sentences were taken from the stega documents in the encoding method for each subject category and 240 stega sentences were obtained in total. Thus, there are 250 nonstega and 240 stega sentences in the whole corpus. In addition, the accuracy rates of the proposed incorporated steganalysis model and the GCN, GAT, GIN networks are tested separately on the texts produced by each of the four encoding methods. During these experiments, 50 stega test sentences and 50 nonstega test sentences (400 in total from 4 subject categories for each of saac, ac, huffman and bins coding) were taken from documents in each subject category. The results obtained from these trials are presented in Table 3.

In order to transform the generated dataset into a graph as in benchmark datasets such as 20NG , R8 , R522 which are widely used in text classification, the work in (Yao et. al 2019) is taken as a reference. In the knowledge distillation phase, the output vector of the BERTbert-base-uncased model [CLS] token used as a tutor model is taken as the document embedding vector of the GCN, GAT and GIN networks. Then a feed-forward layer was added to allow each of the student models to make classification predictions. The GAT network uses a multi-head attention mechanism with 8 heads.

2.2. Experimental setting

The hyperparameters used in the three phases of the study are as follows: 1e-5 was determined as the learning rate in all phases. In the Knowledge distillation and Incorporating features stages, GCN, GAT and GIN networks consist of 1 layer. The number of neurons in the hidden

layers of the GCN and GIN networks is 200, while the GAT network has a neuron size of 200 units multiplied by the number of heads used. Fur-thermore, MLP (multi-layer perceptron) was used during the formation of the GIN network and an iterative neighbor node aggregation (i.e. message passing) method was adopted. Adam (Kingma and Ba 2014) was used as the optimization algorithm. Dropout value is 0.5 and batch size is 128. In the GIN network, sum pooling is used as graph pooling and node pooling method. Acc (Accuracy) and R (Recall) metrics were used for detection perfor-mance in the experiments. The inference time measure, which is the time in seconds that it takes to determine whether a text is a stega or not, was used to determine the efficiency of the steganalysis model.

2.3. Finetune BERT for modelling feature differences

A large corpus and a complicated network structure are needed in order to train a model from scratch on the dataset that will be utilized for the study and to achieve ex-cellent performance. However, it is not always possible to obtain a corpus of large-scale and labelled texts. In this case, the model to be created is expected to have a more detailed structure in order to capture feature differences even in a small number of data sets. The medium size of the data set used in this study necessitated the use of a robust model structure. For this purpose, a pre-trained BERT model (Bert Based Uncased) was finetuned using the dataset prepared for the study, which was generated by the text generation method based on hidden messages. Instead of a BERT model pre-trained on a sizable corpus, a BERT model finetuned on the dataset utilized here was employed as a tutor model to reduce the training time in the second phase of the study. The operations performed at this stage of the study constitute the "Fine-Tuned Bert" part of Figure 1. Finetune steps performed using the relevant dataset of the Bert model are given in Algorithm 1.

Algorithm 1: (Finetune BERT for Modelling Feature Differences)
Input: dataset (D)
Output: Finetuned Bert Model

1.Do build text graph from D
2.create train, validation and test nodes
3. Pretrained Bert encodings
4. for nodes in D do
5. do train steps
6. end for

2.4. Knowledge distillation

At the second stage of the model, in order to overcome the problem of increased inference time due to the complex BERT architecture, the weights learned by BERT were transmitted independently to each of the GCN (Kipf and Welling 2016), GAT (Velickovic et. al 2017), and GIN (Xu et. al 2018) networks, and information distillation was conducted. Here, the BERT model serves as the instructor, while the GCN, GAT, and GIN networks represent the students. At this step of the investigation, we adapted the BertGCN technique described in (Lin et. al 2021) to the GAT and GIN networks. At the distillation stage, the BertGCN, BertGAT, and BertGIN models each have one hidden layer, softmax is used as the activation function, and cross-entropy is utilized as the loss function. Moreover, in order to improve the capacity of the BertGAT model and to stabilize the learning process, a multi-head attention structure is used in the GAT architecture.

In each of the BertGCN, BertGAT and BertGIN models, the representation of the document nodes is initialized by the pre-trained BERT model. In other words, these representations (the X value in equation 2) constituted an input for each of the GCN, GAT and GIN networks. The contribution of the BERT representations used at this stage is revealed in the fact that the BERT model is able to transfer its weight values to other networks, since it has already been trained with very large amounts of raw data. During the training period, document representations are iteratively updated over the GCN, GAT and GIN networks. Then the outputs of the document nodes are sent to the softmax classifier for prediction. The 3 models created in the distillation phase use a heterogeneous graph structure as they contain both word nodes and document nodes. The work in (Yao et. al 2019) is taken as a reference for the construction of this structure. The operations performed at this stage of the study constitute the “Knowledge Distillation” part of Figure 1. Knowledge Distillation steps are included in Algorithm 2.

Algorithm 2: Knowledge Distillation

Input: Finetuned Bert Model, dataset (D)
Output: BertGCN, BertGAT, BertGIN distilled models
At this stage, the Bert weights are transferred to each of the GCN, GAT and GIN networks separately.

1.Do build text graph from D
2.create train, validation and test nodes
3. Bert encodings
4. for nodes in D do
5. if selected model is GCN then
6.make prediction on embeddings via BERT
7.Send the node features (edge weights, node features) to GCN model
8.Combine the classification results obtained by the BERT and GCN
9.else if selected model is GAT then
10.repeat steps 6 to 8 for GAT

11.else
12.repeat steps 6 to 8 for GIN
13.end if
14. end for

TF-IDF (Term frequency-Inverse Document Frequency) is applied for word-document edges in the graph structure and PPMI (Positive Point-Wise Mutual İnformation) is applied for word-word edges. The definition of edge weights between nodes i and j is given in (1) (Lin et. al 2021).

$$A_{i,j} = \begin{cases} PPMI(i,j), & i,j \; are \; words \; and \; i \neq j \\ TF-IDF(i,j), & i \; is \; document, j \; is \; word \\ 1, & i = j \\ 0, & otherwise \end{cases} \quad (1)$$

The BertGCN, BertGAT and BertGIN models all use the finetuned Bert model to find document embeddings and treat these embeddings as inputs to the document nodes. It is expressed in terms of embeddings of document nodes $X_{doc} \in \mathbb{R}^{n_{doc} \times d}$. d is the embedding dimension. The initial feature matrix of the nodes is as given in (2) (Lin et. al 2021).

$$\mathrm{X} = \binom{X_{doc}}{0}_{(n_{doc}+n_{word}) \times d} \quad (2)$$

i. output feature matrix of the GCN, GAT and GIN layers is calculated as in (3). Here, ρ is the activation function, $\tilde{A}$ is the normalized neighborhood matrix and $W^i \in \mathbb{R}^{d_{i-1} \times d_i}$ represents the weight matrix of the layer. $L^0 = \mathrm{X}$ defines the input feature matrix of the model. The outputs of each of the BertGCN, BertGAT and BertGIN models, calculated according to equation (3) (Lin et. al 2021), were then sent to the softmax classifier in (4) (Lin et. al 2021).

$$L^{(i)} = \rho\left(\tilde{A} L^{(i-1)}\right) \quad (3)$$

$$Z_{GCN,GAT,GIN} = softmax(g(X,A)) \quad (4)$$

The g in Equation (4) represents the model used. Cross entropy loss is used as the loss function.

In this study, BertGCN, BertGAT and BertGIN models are optimized with a classifier that operates on Bert embeddings. This resulted in faster extraction and higher performance. Accordingly, document embeddings (denoted by X in Equation (5)) are sent to a dense layer with softmax activation. Then, these weight values were sent to each of the GCN, GAT and GIN models separately based on the Bert weights obtained in Equation 5, and the GCN, GAT and GIN networks were trained within their own network architectures.

The Bert model used here plays the role of teacher network, while the GCN, GAT and GIN models play the role of student network. In the last stage of the information distillation process, the prediction values obtained by the Bert network and the prediction values obtained by each of the GCN, GAT and GIN networks were combined (for BertGCN, Bert prediction values were added with GCN prediction values; for BertGAT, Bert prediction values were added with GAT prediction values; and for BertGIN, Bert prediction values were added with GIN prediction values) to obtain the total predicted value (Equations 6,7,8) (Lin et. al 2021).

$$Z_{BERT} = softmax(WX) \quad (5)$$

$$Z_{totalforBertGCN} = Z_{Bert} + Z_{GCN} \quad (6)$$

$$Z_{totalforBertGAT} = Z_{Bert} + Z_{GAT} \quad (7)$$

$$Z_{totalforBertGIN} = Z_{Bert} + Z_{GIN} \quad (8)$$

2.5. Incorporating features

This step of the study was carried out after experiments showed that combining the features obtained by different networks and performing a stega text detection study based on these new values greatly improved the detection rate. Transfer Learning method was used in the last stage of the study. In this way, by utilizing the weight information of the BERT model trained using a large corpus, a faster learning is provided on the dataset created for this study. In the last stage of the study, the weights obtained by combining the distilled BertGCN, BertGAT and BertGIN models from the previous stage, Knowledge Distillation, were transferred to the newly created model and this new model was trained on the old acquired weights. Therefore, the parameters of these 3 student networks, which do not have a fully connected layer, were transferred to the newly created model, causing the previous weights to be the input information of the new model. This phase of the study is based on the work in (Peng et. al 2021). The operations performed at this stage of the study constitute the " Incorporating Features " part of Figure 1. Incorporating Features steps are included in Algorithm 3.

Algorithm 3: Incorporating Features
Input: BertGCN, BertGAT, BertGIN distilled models
Output: Proposed incorporated model

1.Combine GCN-GAT-GIN then do incorporated model
2.Sum the weights of BertGCN, BertGAT and BertGIN then transfer to incorporated model

In the incorporated model created by combining the features of 3 networks, the power of the GCN network to

process non-sequential graph embeddings that cannot be processed by traditional network structures such as RNN (Recurrent Neural Networks) and CNN is utilized. Therefore, the different order of the input nodes does not affect the outcome of the GCN network. As the study in (Jing et. al 2021, Liu et. al 2022) emphasized that the GAT network is superior to the GCN network due to its ability to use the attention mechanism to weight the sum of the properties of adjacent nodes, so in this study, GAT network is also included and the effect of the GAT model on the classification is evaluated. According to the study in (Veličković et. al 2017), the graph attention mechanism is different from the self-attention mechanism. In the self-attention mechanism, weights are as-signed to all nodes of a document, whereas in the graph attention mechanism, nodes with different numbers of neighboring nodes can be assigned individual weight values and these nodes can be processed simultaneously in parallel with high computational efficiency (Wang and Li 2022, Zhang et. al 2023). While GNN (Graph Neural Network) variants are quite successful in learning node embed-dings, they may be inadequate in the case of learning entire graph embeddings and thus classifying an entire graph. In this case, a new network approach is needed to learn the entire graph by combining node embeddings. At this point, the GIN network provides a solution to this need thanks to its global pooling layer. In this study, the GIN network was preferred be-cause it gives more successful results in graph classification than the GCN network (Rassil et. al 2020, Xu et. al 2018).

3. Results and Discussions

3.1. Comparison with state-of-the-arts

In this study, nine distinct DL-based steganalysis approaches are compared to the method employed in this study in terms of both extraction times and accuracy rates. The findings of this comparison are provided in Table 2. Hence, it is seen that the graph-based Incorporated model developed in this study achieves better results than state-of-the-art methods for both huffman and Bin-LM encoding methods, regardless of the embedding rate. In contrast to the work in (Peng et. al 2021), which is referenced at a new integrated networking point, it is observed that the accuracy does not decrease or even increases in cases where huffman encoding is used, although the embedding rate increase. In (Peng et. al 2021), when huffman coding was used, an increase in the embedding rate had the effect of decreasing the detection accuracy. The superiority of the detection performance of the proposed model is because the pre-trained Bert model, which has huge information capacity, is subjected to finetune and distillation processes on a dataset containing stega-nonstega texts, to capture the local and global features of the texts in more detail. In addition, combining the strengths of each of the GAT, GIN and GCN networks under a single network roof is the underlying reason for this superiority. The aim of the study is to develop a model with high accuracy and faster inference. For this purpose, when the test result values given in Figure 2a-2b and Table 1 are analyzed, it is seen that the proposed work can provide faster inference compared to previous studies in both huffman and Bin-LM coding methods. It is thought that distilling the Bert model, which has a more complex network structure and therefore a longer inference time, into the GAT, GIN and GCN networks, which are much smaller and less complex networks, shortens this time. In terms of inference time, it is close to, but shorter than, the work in reference (Peng et. al 2021) were obtained. The reason for this is that graph-based networks are better able to capture non-ordered word sequences and the relationship between these words.

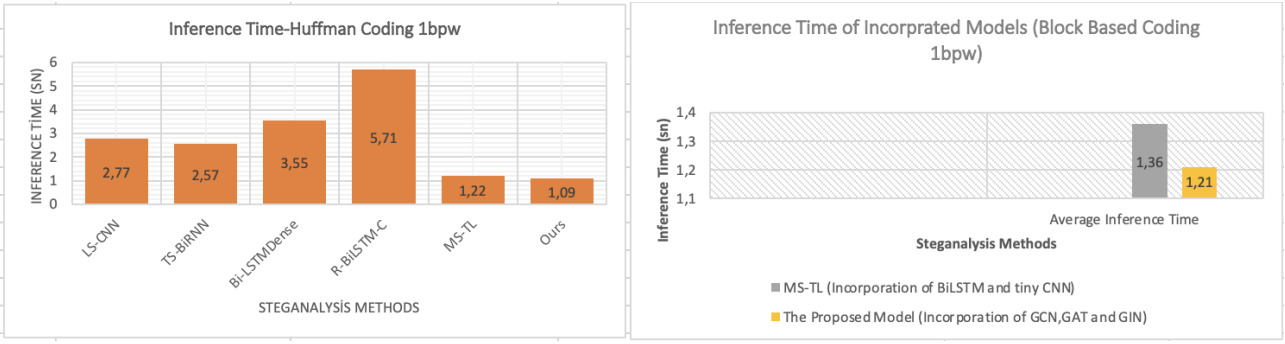

Figure 2a-2b. The inference efficiency of different steganalysis methods on Huffman and Bin-LM coding.

3.2. Effects of integrated training

Examining Table 3 reveals that finetuning the Bert network with the related dataset, increases the detection success. In contrast to the models in (Wen et. al 2019, Yang et. al 2019 (c), Niu et. al 2019), the inference time of the finetuned model presented here is significantly longer. In order to shorten the inference time, a transfer learning technique is used to transfer the knowledge gathered by the distilled small-scale graph-based models

(BertGCN, BertGAT, and BertGIN) to the new model, which is a combination of the three tiny models. Thus, the extraction time is drastically reduced. At 1bpw, the VLC encoding method has a modest drop in precision (Our FineTuned: 0,9907; Our Incorporated: 0.9706), an increase in accuracy was observed at 3.22 bpw of the same coding method (Our FineTuned: 0.9732; Our Incorporated: 0.9812). At 1bpw, in the Bin-LM coding, the proposed incorporated model shows an increase in accuracy (Our FineTuned: 0.8995 ; Our Incorporated: 0.9353) but at 3 bpw a small decrease was observed (Our FineTuned: 0.9689 ; Our Incorporated: 0.9535).

Figure 3 illustrates the accuracy rates produced by the incorporated model for each encoding technique and bpw value. Hence, the corpus including stega texts encoded using the Saac approach yields the best accuracy rate. According to the study in (Shen et. al 2020), the Saac approach is more effective than Bin-LM encoding, huffman, and arithmetic encoding in terms of imperceptibility of stega texts. It is remarkable that the incorporated model realized here was able to obtain a high accuracy rate even with the Saac approach, which has a high imperceptibility rate.

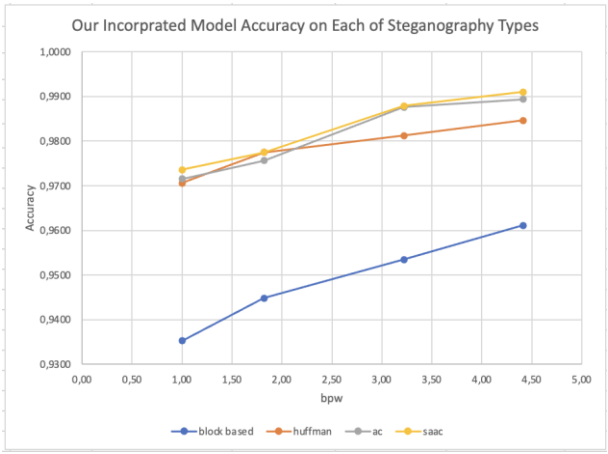

Figure 3. Our model accuracy on each of encoding methods.

Table 4 displays the detection rates of the model presented in (Peng et. al 2021) and the model developed in this paper using various embedding strategies. Analyzing the results in this table reveals that the model used here is superior in terms of detection capability for all bpw ratios for both Bin-LM and Huffman coding. The reason why the proposed model has a higher detection rate than the model created by combining BILSTM and CNN networks is because graph-based networks are better able to capture not only the relationship between ordered words, but also the level of interest (global features) of all words in relation to one another. This study also reveals that tiny GCN networks are inferior than tiny GAT and tiny GIN networks in terms of detection success for each encoding technique. This is because the GAT network use the attention mechanism to weight the features of adjacent nodes, whereas the GIN network is superior at classifying at the graph level as opposed to the node level. The investigations in (Xu et. al 2018, Zhang et. al 2021) support the results acquired in this study by the GIN net-work. Another conclusion that can be drawn from Table 3 is that the incorporated model has a higher accuracy rate than the distilled GCN, GAT and GIN models. The main reason for this is considered to be that the global characteristics learned in the distill stage are learned in more detail in the incorporated stage.

4. Conclusions

In the steganalysis process, the model's applicability to real life, or its lowest inference time, is just as crucial as its detection capability. This study was conducted to address a gap in the literature upon being noticed that the combination of GCN, GAT, and GIN networks through transfer learning has not been employed in the field of steganalysis before.

A new model with an effective inference time and high accuracy has been developed. In the study, which adopts the transfer learning method, the pre-trained BERT model was finetuned to model feature differences. Then, the feature information acquired by the finetuned model was transferred to the tiny GCN, GAT and GIN networks separately. In the last phase, a transfer learning approach was used to combine the knowledge that BertGCN, BertGAT, and BertGIN had acquired. When the findings are evaluated, it is found that the work applied here provides results that are superior to those achieved by state-of-the-art methods in terms of both inference time and model accuracy. In later versions of the study, it is intended to assess the classification and inference time per-formance of quantum ML on graph-based steganalysis data.

Table 1. Comparison of detection accuracy, model size and inference results.

Steganography	Dataset	Steganalysis Method	Acc- 1 bpw	Acc- 3,22 bpw	Average Model Size	Average Inference Time
VLC-Huffman (Yang et. al 2019 (a))	IMDB	MS-TL (Incorporation of BiLSTM and tiny CNN) (Peng et. al 2021)	0.9711	0.9285	18.70M (x2.67)	1.22s (x8.13)
	Custom Dataset	The Proposed Model (Incorporation of GCN,GAT and GIN)	0.9706	0.9812	19.23M (x1.12)	1.09 s(x2.45)
			Acc - 1bpw	**Acc - 3bpw**		
Bin-LM (Fang et. al 2017)	IMDB	MS-TL (Incorporation of BiLSTM and tiny CNN) (Peng et. al 2021)	0.8660	0.9435	19.21M(x2.08)	1.36s(x7.56)
	Custom Dataset	The Proposed Model (Incorporation of GCN,GAT and GIN)	0.9353	0.9535	20.15M(x1.65)	1.21 s(x3.67)

Table 2. Detection accuracy of different steganalysis methods on the Bin-LM and the VLC-based encoding methods.

		Steganography Type							
		VLC-Huffman (Yang et. al 2019 (a))				Bin-LM (Fang et. al 2017)			
	Dataset	**IMDB**				**IMDB**			
Method	bpw	1.00	1.82	3.22	4.41	1	2	3	4
LS-CNN (Wen et. al 2019)	Acc	0.9720	0.9525	0.9270	0.8585	0.8395	0.8965	0.9395	0.9630
	R	0.9740	0.9430	0.9250	0.8600	0.8280	0.9010	0.9510	0.9620
TS-BiRNN (Yang et. al 2019 (c))	Acc	0.9595	0.9575	0.9100	0.8565	0.8470	0.8855	0.9335	0.9660
	R	0.9540	0.9620	0.8940	0.8540	0.8580	0.8800	0.9280	0.9750
R-BiLSTM-C (Niu et. al 2019)	Acc	0.9765	0.9600	0.9175	0.8545	0.8445	0.9060	0.9395	0.9645
	R	0.9750	0.9540	0.9060	0.8390	0.8680	0.8990	0.9490	0.9740
BiLSTM-Dense (Yang, H et. al 2020)	Acc	0.9633	0.9458	0.9233	0.8580	0.8435	0.8980	0.9415	0.9675
	R	0.9100	0.9410	0.9243	0.8890	0.8260	0.9100	0.9350	0.9770
MS-TL (Peng et. al 2021)	Acc	0.9711	0.9655	0.9285	0.8715	0.8660	0.9115	0.9435	0.9720
	R	0.9711	0.9664	0.9440	0.8670	0.8770	0.9120	0.9490	0.9680
GCN (Wu et. al 2021)	bpw	1.000	2.183	3.285	-	1	2	3	-
	Acc	0.784	0.913	0.960	-	0.859	0.939	0.967	-
	R	0.769	0.911	0.964	-	0.851	0.929	0.967	-
LS-BGAT (Xiang et. al 2022) Task-2 (Only with VLC)	bpw	(Any bpw data was not given by them)				-	-	-	-
	Acc	0.951				-	-	-	-
	R	0.976				-	-	-	-
Sesy (Bert-GAT) (Yang, J et. al 2021)	bpw	1.000	1.838	2.498	3.721	1.000	1.777	2.467	3.855
	Acc	0.839	0.916	0.937	0.977	0.931	0.953	0.971	0.988
	R	0.976	0.975	0.972	0.985	0.944	0.973	0.978	0.989
HGA (Steganalysis) (Fu et. al 2022)		Steganography Type							
		VLC-Huffman (Yang et. al 2019 (a))				(Yang et. al 2021)			
	bpw	1	2	3	-	1	2	3	-
	Acc (Recall is not provided by them)	0.935	0.923	0.91	-	0.898	0.883	0.849	-
OURS		Steganography Type							
		VLC-Huffman (Yang et. al 2019 (a))				Bin-LM (Fang et. al 2017)			
		(Custom Dataset. All corpus included)							
	bpw	1.00	1.82	3.22	4.41	1	2	3	4
	Acc	0.9706	0.9775	0.9812	0.9846	0.9353	0.9448	0.9535	0.9611
	R	0.9534	0.9679	0.9769	0.9853	0.8959	0.9024	0.9133	0.9275

Table 3. Detection accuracy and inference time of steganalysis methods and modelling feature differences (finetuned)

	Method	Acc-1bpw	Acc-3,22bpw	Inference Time (sn)
VLC-Huffman (Yang et. al 2019 (a))	LS-CNN (Wen et. al 2019)	0.9720	0.9270	2.77
	TS-BiRNN (Yang et. al 2019 (c))	0.9595	0.9100	2.57
	R-BiLSTM-C (Niu et. al 2019)	0.9765	0.9175	5.71
	MS-TL Finetuned BERT (Peng et. al 2021)	0.9835	0.9570	53.63
	Proposed Finetuned BERT (Custom Dataset)	0.9907	0.9732	38.17
	MS-TL (Incorporation of BiLSTM and tiny CNN) (Peng et. al 2021)	0.9711	0.9285	1.22
	The Proposed Model (Incorporation of GCN,GAT and GIN)	**0.9706**	**0.9812**	**1.09**
	Method	**Acc 1bpw**	**Acc 3bpw**	**Inference Time (sn)**
Bin-LM (Fang et. al 2017)	LS-CNN (Wen et. al 2019)	0.8395	0.9395	2.92
	TS-BiRNN (Yang et. al 2019 (c))	0.8470	0.9335	2.84
	R-BiLSTM-C (Niu et. al 2019)	0.8445	0.9395	6.02
	MS-TL Finetuned BERT (Peng et. al 2021)	0.8830	0.9510	55.78
	Proposed Finetuned BERT (Custom Dataset)	0.8995	0.9689	40.13
	MS-TL (Incorporation of BiLSTM and tiny CNN) (Peng et. al 2021)	0.8660	0.9435	1.36
	The Proposed Model (Incorporation of GCN,GAT and GIN)	**0.9353**	**0.9535**	**1.21**

Table 4. Comparison of Distilled and Incorporated Models Detection Accuracy

Steganography	Model	Acc Values on IMDB and Our Dataset	
		1bpw	3bpw
Bin-LM (Fang et. al 2017)	Tiny CNN (Peng et. al 2021)	0.8505	0.9410
	Tiny BiLSTM (Peng et. al 2021)	0.8580	0.9405
	Tiny CNN-BiLSTM (Peng et. al 2021)	0.8532	0.9413
	Incorporated CNN+BİLSTM (Peng et. al 2021)	0.8660	0.9435
		1bpw	3bpw
	Tiny BertGCN	0.8792	0.9534
	Tiny BertGAT	0.8816	0.9598
	Tiny BertGIN	0.8897	0.9612
	Proposed Incorporated Model	**0.9353**	**0.9535**
VLC-Huffman (Yang et. al 2019 (a))		1bpw	3bpw
	Tiny CNN (Peng et. al 2021)	0.9700	0.9150
	Tiny BiLSTM (Peng et. al 2021)	0.9705	0.9195
	Tiny CNN-BiLSTM (Peng et. al 2021)	0.9711	0.9188
	Incorporated CNN+BİLSTM (Peng et. al 2021)	0.9700	0.9285
		1bpw	3.22bpw
	Tiny BertGCN	0.9224	0.9556
	Tiny BertGAT	0.9396	0.9673
	Tiny BertGIN	0.9445	0.9738
	Proposed Incorporated Model	**0.9706**	**0.9812**
AC (Ziegler et. al 2019)		1bpw	3.22bpw
	Tiny BertGCN	0.9109	0.9628
	Tiny BertGAT	0.9376	0.9653
	Tiny BertGIN	0.9532	0.9694
	Proposed Incorporated Model	**0.9715**	**0.9876**
SAAC (Shen et. al 2020)		1bpw	3.22bpw
	Tiny BertGCN	0.9257	0.9689
	Tiny BertGAT	0.9449	0.9721
	Tiny BertGIN	0.9574	0.9768
	Proposed Incorporated Model	**0.9736**	**0.9879**

Declaration of Ethical Standards
The author declare that they comply with all ethical standards.

Credit Authorship Contribution Statement
Author: Conceptualization, investigation, methodology and software, experimental study, visualization and writing – original draft, supervision and writing – review and editing

Declaration of Competing Interest
The author have no conflicts of interest to declare regarding the content of this article.

Data Availability Statement
All data generated or analyzed during this study are included in this published article.

Acknowledgement
I would like to thank the "Sky Translation Office" for the language editing of the article.

5. References

Chen, Z., Huang, L., Miao, H., Yang, W., Meng, P. 2011. Steganalysis against substitution-based linguistic steganography based on context clusters. *Computers & Electrical Engineering*, 37(**6**), 1071-1081. https://doi.org/10.1016/j.compeleceng.2011.09.014

Fang, T., Jaggi, M., Argyraki, K. 2017. Generating steganographic text with LSTMs. *arXiv preprint* arXiv:1705.10742. https://doi.org/10.48550/arXiv.1705.10742

Fu, Z., Yu, Q., Wang, F., Ding, C. 2022. HGA: Hierarchical feature extraction with graph and attention mechanism for linguistic steganalysis. *IEEE Signal Processing Letters*, **29**, 1734-1738. https://doi.org/10.1109/LSP.2022.3192534

Jing, W., Song, X., Di, D., Song, H. 2021. GeoGAT: Graph model based on attention mechanism for geographic text classification. *Transactions on Asian and Low-Resource Language Information Processing*, 20(**5**), 1-18. https://doi.org/10.1145/3450626

Kang, H., Wu, H., Zhang, X. 2020. Generative text steganography based on LSTM network and attention mechanism with keywords. *Electronic Imaging*, 2020(**4**), 291-1. https://doi.org/10.2352/ISSN.2470-1173.2020.4.MWSF-291

Kingma, D.P., Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint* arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980

Kipf, T.N., Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint* arXiv:1609.02907. https://doi.org/10.48550/arXiv.1609.02907

Li, S., & Wang, J., Liu, P. 2022. Detection of generative linguistic steganography based on explicit and latent text word relation mining using deep learning. *IEEE Transactions on Dependable and Secure Computing*, 20(**2**),1476-148. https://doi.org/10.1109/TDSC.2021.3062703

Lin, Y., Meng, Y., Sun, X., Han, Q., Kuang, K., Li, J., Wu, F. 2021. BertGCN: Transductive text classification by combining GCN and BERT. *arXiv preprint* arXiv:2105.05727. https://doi.org/10.48550/arXiv.2105.05727

Liu, P., Tian, B., Liu, X., Gu, S., Yan, L., Bullock, L., Zhang, W. 2022. Construction of power fault knowledge graph based on deep learning. *Applied Sciences*, 12(**14**), 6993. https://doi.org/10.3390/app12146993

Meng, P., Hang, L., Chen, Z., Hu, Y., Yang, W. (2010). STBS: A statistical algorithm for steganalysis of translation-based steganography. Information Hiding: 12th International Conference. IH 2010. Calgary, AB, Canada, 208-220.

Meng, P., Hang, L., Yang, W., Chen, Z., Zheng, H. (2009). Linguistic steganography detection algorithm using statistical language model. Proceedings of the 2009 International Conference on Information Technology and Computer Science. Kiev, Ukraine, 25-26.

Niu, Y., Wen, J., Zhong, P., Xue, Y. 2019. A hybrid R-BILSTM-C neural network based text steganalysis. *IEEE Signal Processing Letters*, 26(**12**), 1907-1911. https://doi.org/10.1109/LSP.2019.2955374

Peng, W., Zhang, J., Xue, Y., Yang, Z. 2021. Real-time text steganalysis based on multi-stage transfer learning. *IEEE Signal Processing Letters*, **28**, 1510-1514. https://doi.org/10.1109/LSP.2021.3105493

Rassil, A., Chougrad, H., Zouaki, H. (2020). The importance of local labels distribution and dominance for node classification in graph neural networks. Proceedings of the 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). Miami, FL, USA, pp. 1505-1511.

Shen, J., Heng, J., & Han, J. 2020. Near-imperceptible neural linguistic steganography via self-adjusting arithmetic coding. *arXiv preprint* arXiv:2010.00677. https://doi.org/10.48550/arXiv.2010.00677

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y. 2017. Graph attention networks. *stat*, 1050(**20**), 10-48550. https://doi.org/10.48550/arXiv.1710.10903

Wang, H., Li, F. 2022. A text classification method based on LSTM and graph attention network. *Connection Science*, **34**(**1**), 2466-2480. https://doi.org/10.1080/09540091.2022.2044605

Wen, J., Zhou, X., Zhong, P., Xue, Y. 2019. Convolutional neural network based text steganalysis. *IEEE Signal Processing Letters*, **26**(**3**), 460-464. https://doi.org/10.1109/LSP.2019.2895260

Wu, H., Yi, B., Ding, F., Feng, G., Zhang, X. 2021. Linguistic steganalysis with graph neural networks. *IEEE Signal Processing Letters*, **28**, 558-562. https://doi.org/10.1109/LSP.2021.3058369

Xiang, L., Liu, Y., You, H., Ou, C. 2022. Aggregating local and global text features for linguistic steganalysis. *IEEE Signal Processing Letters*, **29**, 1502-1506. https://doi.org/10.1109/LSP.2022.3190781

Xiang, L., Sun, X., Luo, G., Xia, B. 2014. Linguistic steganalysis using the features derived from synonym frequency. *Multimedia Tools and Applications*, **71**, 1893-1911. https://doi.org/10.1007/s11042-012-1303-4

Xiang, L., Yu, J., Yang, C., Zeng, D., Shen, X. 2018. A word-embedding-based steganalysis method for linguistic steganography via synonym substitution. *IEEE Access*, **6**,64131-64141. https://doi.org/10.1109/ACCESS.2018.2876935

Xu, K., Hu, W., Leskovec, J., Jegelka, S. 2018. How powerful are graph neural networks?. *arXiv preprint* arXiv:1810.00826. https://doi.org/10.48550/arXiv.1810.00826

Yang, H., Bao, Y., Yang, Z., Liu, S., Huang, Y., Jiao, S. (2020). Linguistic steganalysis via densely connected LSTM with feature pyramid. Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security. 5-10.

Yang, H., Cao, X. 2010. Linguistic steganalysis based on meta features and immune mechanism. *Chinese Journal of Electronics*, **19**, 661-666. https://doi.org/10.1049/cje.2010.661666

Yang, J., Yang, Z., Zhang, S., Tu, H., Huang, Y. 2021. SeSy: linguistic steganalysis framework integrating semantic and syntactic features. *IEEE Signal Processing Letters*, **29**, 31-35. https://doi.org/10.1109/LSP.2021.3131807

Yang, Z., Guo, X., Chen, Z., Huang, Y., Zhang, Y. 2019 (a). RNN-Stega: linguistic steganography based on recurrent neural networks. *IEEE Transactions on Information Forensics and Security*, 14(**5**), 1280-1295. https://doi.org/10.1109/TIFS.2018.2871746 (a)

Yang, Z., Huang, Y., Zhang, Y.J. 2019(b). A fast and efficient text steganalysis method. *IEEE Signal Processing Letters*,26(**4**),627-631. https://doi.org/10.1109/LSP.2019.2903902 (b)

Yang, Z., Wang, K., Li, J., Huang, Y., Zhang, Y.J. 2019(c). TS-RNN: text steganalysis based on recurrent neural networks. *IEEE Signal Processing Letters*, **26(12**), 1743-1747. https://doi.org/10.1109/LSP.2019.2950464 (c)

Yang, Z., Huang, Y., Zhang, Y.J. 2020. TS-CSW: text steganalysis and hidden capacity estimation based on convolutional sliding windows. *Multimedia Tools and Applications*, **79**, 18293-18316. https://doi.org/10.1007/s11042-019-08345-7

Yang, Z.L., Zhang, S.Y., Hu, Y.T., Hu, Z.W., Huang, Y.F. 2021. VAEStega: linguistic steganography based on variational auto-encoder. *IEEE Transactions on Information Forensics and Security*, **16**, 880-895. https://doi.org/10.1109/TIFS.2020.3037121

Yao, L., Mao, C., Luo, Y. (2019). Graph convolutional networks for text classification. Proceedings of the AAAI Conference on Artificial Intelligence. 7370-7377.

Zhang, L., Ding, J., Xu, Y., Liu, Y., Zhou, S. 2021. Weakly-supervised text classification based on keyword graph. *arXiv preprint* arXiv:2110.02591. https://doi.org/10.48550/arXiv.2110.02591

Zhang, Y., Xu, Y., Zhang, Y. 2023. A graph neural network node classification application model with enhanced node association. *Applied Sciences*, 13(**12**), 7150. https://doi.org/10.3390/app13127150

Ziegler, Z., Deng, Y., Rush, A. (2019). Neural Linguistic Steganography. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China, 1210-1215.

Zou, J., Yang, Z., Zhang, S., Rehman, S.U., & Huang, Y. (2020). High-Performance Linguistic Steganalysis, Capacity Estimation and Steganographic Positioning. In Digital Forensics and Watermarking: 19th International Workshop, IWDW 2020. Melbourne, VIC, Australia, 80-93.