IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 1, 2018, accepted October 15, 2018, date of publication October 26, 2018,
date of current version November 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2878273

A Word-Embedding-Based Steganalysis Method for Linguistic Steganography via Synonym Substitution

LINGYUN XIANG[1,2], JINGMIN YU[2], CHUNFANG YANG[3], DAOJIAN ZENG[1,2], AND XIAOBO SHEN[4]

[1]Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha 410114, China
[2]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China
[3]Zhengzhou Science and Technology Institute, Zhengzhou 450001, China
[4]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798

Corresponding author: Lingyun Xiang (xiangly210@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61202439, Grant 61302159, Grant 61602059, and Grant 61872448, in part by the Scientific Research Foundation of Hunan Provincial Education Department of China under Grant 16A008, and in part by the Hunan Key Laboratory of Smart Roadway and Cooperative Vehicle-Infrastructure Systems under Grant 2017TP1016.

ABSTRACT The development of steganography technology threatens the security of privacy information in smart campus. To prevent privacy disclosure, a linguistic steganalysis method based on word embedding is proposed to detect the privacy information hidden in synonyms in the texts. With the continuous Skip-gram language model, each synonym and words in its context are represented as word embeddings, which aims to encode semantic meanings of words into low-dimensional dense vectors. The context fitness, which characterizes the suitability of a synonym by its semantic correlations with context words, is effectively estimated by their corresponding word embeddings and weighted by TF-IDF values of context words. By analyzing the differences of context fitness values of synonyms in the same synonym set and the differences of those in the cover and stego text, three features are extracted and fed into a support vector machine classifier for steganalysis task. The experimental results show that the proposed steganalysis improves the average F-value at least 4.8% over two baselines. In addition, the detection performance can be further improved by learning better word embeddings.

INDEX TERMS Steganalysis, steganography, word embedding, Skip-gram language model, TF-IDF.

I. INTRODUCTION

In nowadays, with the development of communication and information technologies, smart campuses have attracted more and more attentions from industry, academia, and education [1]. Smart campuses provide a better teaching and learning environment for teachers and students with improved teaching quality and efficiency. However, smart campus system may be threatened by different attacks from viruses, trojan horses, steganography [2], [3] that could negatively impact the security of the smart campuses [4]. In smart campus system, there are massive digital information resources [5] with abundant personal privacy information. The privacy information such as passwords, credit card accounts, identification cards, etc. of teachers and students may be captured by trojan horses, while steganography can be employed to imperceptibly transmit privacy information out of smart campuses, which will not only result in great economic losses but also put teachers and students in danger. Therefore, steganalysis, which aims to discover the existence of information hidden by steganography and even extract the information hidden [6], should be an important research topic for smart campuses to prevent privacy disclosure. With the development of related computational theories [7], [8], machine learning [9]–[12], and deep learning [13], the performance of steganalysis can be greatly improved to effectively protect the privacy information in smart campuses.

With the prevalence of digital texts, linguistic steganography has attracted increasing interest during the past few years. Researchers proposed various linguistic steganographic methods [14], [15] to achieve the goal of covert communication. The secret message is embedded into a cover text by equivalent linguistic transformations to preserve the

meaning of the original text. As the existence of the secret message is unknown to the third part, it can be effectively protected. However, it is possible for the terrorist to deploy the linguistic steganography for planning terrorist activities, making illegal transactions, exchanging secret information, etc. In order to supervise and prevent the abuse of the linguistic steganography, a countermeasure technique called linguistic steganalysis is proposed to discover the existence of the secret message in a text [16], [17].

In the field of linguistic steganography, synonym substitution(SS) is a popular transformation adopted by many researchers [18]–[25]. It treats synonymous words as interchangeable elements to form a substitution set. The alterations of the choice of synonyms in a substitution set can hide information and generate unsuspicious alternatives for a cover text. According to this theory, Winstein [18] developed the first practical SS-based steganographic system, called Tlex, which employed a multi-base coding method to encode synonyms extracted from WordNet. The secret information is hidden in the text by selecting proper synonyms from substitution sets. Unfortunately, this approach is not ideal, because this system would produce many semantic and pragmatic problems in the stego text and change the statistical characteristics of the cover text.

Consequently, researchers have started looking into selecting grammatically and semantically correct synonym transformations to generate more fluent stego texts and ensure that the statistical characteristics of the stego text are nearly the same with its cover one. For example, Chang and Clark [19] proposed a sophisticate vertex coding method to encode collected synonyms into unrepeated bit string, and checked the applicability of each synonym in its context by a N-gram count method with the Google N-gram corpus. Barmawi [20] used lexical substitution and additional syntactical transformation to increase the payload capacity. Xiang *et al.* [21] proposed a synonym run-length encoding method to embed information by self-adaptively making a positive or negative synonym transformation. As a result, the number of relative high and low frequency synonyms were almost preserved to reduce the embedding distortion. Yang *et al.* [22] bought ideas from image steganography into the SS linguistic steganography, which applied the matrix embedding to improve the embedding efficiency by reducing the number of synonym substitutions. Hu *et al.* [23] propose a double-layered STC (Syndrome-trellis code) scheme to minimize statistical and linguistic distortion caused by synonym substitutions. Particularly, Xiang *et al.* [24] only selected the most and the second most frequent synonym (with highest and second highest frequency) in a synonym set for embedding information. The employed synonyms were effectively compressed by adaptive arithmetic coding to provide a spare for accommodating additional data. The statistical changes caused by synonym substitutions can be reduced by compressing the secret message, which shortened the length of the practical embedded payload [25].

The methods mentioned above have been shown to be effective for hiding information. However, during the information embedding process, the statistical and linguistic characteristics of a text are inevitably changed, which are clues for the steganalysis. In steganalysis methods, it is the key issue to extract detection features to comprehensively depict the changes caused by embedding operations [26]. Some useful features derived from the analysis of the N-gram language model [27], the inverse document frequency(IDF) [28], the context cluster [29], the relative frequency [30]–[32], etc. were proposed for steganalysis task.

The N-gram language model-based steganalysis [27] was the first work against SS linguistic steganography, which extracted features by estimating N-gram probabilities from unmodified and steganographically modified sentences. In [28], the context information weighted by the IDF was employed to measure the suitability of a synonym in its context, and then statistical features were extracted to distinguish normal texts from stego ones. While [29] introduced the context cluster, which composed of a synonym and its context words, to estimate the context fitness. These methods have undesirable detection accuracy or low running efficiency. Thus, researchers focused on relative frequency-based steganalysis, which performed best. Reference [30] sorted the synonymous words in descending order in terms of their frequencies in a large corpus to form a synonym vector. Each synonym has its own attribute pair including its position in a synonym vector and the vector's dimension. The relative frequencies of part of attribute pairs appearing in the detected text are computed for extracting detection features. Reference [30] had greatly improved the detection accuracy compared with the method in [28]. Meanwhile, Chen *et al.* [32] proposed a similar method based on relative frequency analysis, which evaluated the natural relative frequency(NRF) of each substitution element (synonym) within a certain substitution set (a synonymous word set) by their frequencies occurring in a large corpus. Then, the expectation and variance of NRF values of all synonyms in the text were computed as features to detect the secret message. This method performed slightly better than the method in [30].

The existing steganalysis methods can distinguish stego texts from the normal texts with high detection accuracy. Nevertheless, they only consider shallow features, which cannot well capture word semantics and do not include much more informative, various low-level word interactions and relations. As evidence by [33] and [34], lacking such essential information may decrease the representation performance and cause hazard for the following learning task. Therefore, detection performance of the steganalysis should be further improved by making useful of deep semantic characteristics of words, which can sensitively capture the changes caused by synonym substitutions for embedding information.

Motivated by the advantages of word distributed representation, which is beneficial to many natural language processing tasks, such as named entity recognition [35], sentiment classification [36], etc., a linguistic steganalysis

based on word embedding is proposed in this paper. It extracts semantic features by using Neural Network Language Model (NNLM) to reveal the deep and implicit semantic relationships between a synonym and its context words, which are all represented to word embeddings. Based on this, the context fitness measuring the suitability of a synonym in a certain context is modeled by using word embeddings and TF-IDF. Finally, three detection features are extracted by analyzing the context fitness values of synonymous words and fed into a SVM classifier to distinguish stego texts from cover ones. Several experiments are conducted to verify the performance of the proposed method. The experimental results show that the proposed steganalysis method can effectively detect the synonym-substitution-based steganography and it has good generalization ability. Specially, if the word embeddings are learned from the similar subject with the detected stego texts, the proposed steganalysis will perform better.

In summary, the key contributions of this work include:

1) A novel linguistic steganalysis is proposed, which provides an effective way to analyze texts for recognizing synonym-substitution-based stego texts.
2) This paper first introduces word embeddings for steganalysis. Incorporating with word embeddings of synonyms and words in their context, effective detection features can be extracted to better capture the statistical and linguistic changes caused by embedding information.
3) We demonstrate that, by using word embeddings, we can able to get better detection performance than other linguistic steganalysis methods. Moreover, the detection performance of the proposed method can be improved by enhancing the quality of word embeddings.

The rest of this paper is organized as follows. In Section II, we review related work on word representation. The details of the proposed steganalysis will be described in Section III. In Section IV, we present and discuss experimental results. Finally, conclusions are drawn in Section V.

II. WORD REPRESENTATION

The ability to model differences among diverse stego and cover texts plays a critical role in linguistic steganalysis. The first task is to represent words affected by SS steganography to capture the changes of statistical and linguistic characteristics. The existing linguistic steganalysis methods are just employed shallow and insufficient information to represent a word and measure its suitability in a certain context. In order to well capture semantics of natural language words, specially, synonymous words for extracting more effective detection features, we employ word representation learning to map a word into low-dimensional dense vectors.

Word representation learning is very worthwhile, which is a mathematical object associated with each word and represents each word into a vector with word features, semantic or grammatical interpretations. It has received more and more attention in the field of Natural Language Processing (NLP). One of the traditional representations is One-hot representation. It converts each word to a symbolic ID, which indicates the word's index in the vocabulary, and then transforms a word into a feature vector. The dimension of the vector of each word is the same as the size of the vocabulary, but only one dimension is on, while others are off. However, this one-hot representation has two significant drawbacks: (1) curse of dimensionality (2) fail of capturing the semantic similarity between words.

To overcome the shortcomings of one-hot representation, low-dimensional distributed word representation learning represents a word to a dense real-valued low-dimensional word embedding induced by neural language models [37]. Most works for learning word embeddings are based on the thought of modeling the semantic relationship between a word and its context words. The most representative work is done by Bengio *et al.* [38]. They firstly employed a feedforward Neural Network Language Model (NNLM) to learn the word representations, also known as word embeddings. Later, Mikolov *et al.* [39] proposed a Recurrent Neural Network (RNN) based language model to represent words into word embeddings, which made full use of all the context information to predict the next word.

To improve the quality of word embeddings and the training speed, Mikolov *et al.* [40] again proposed two novel model architectures for computing continuous vector representations of words from very large data sets. One is Continuous Bag-of-Words Model(CBOW) predicting the word w_t with the context words $\{w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2}\}$. The second one is Continuous Skip-gram, which uses each word w_t to predict the context words $\{w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2}\}$. Both can generate high quality word embeddings, and are widely used in the NLP tasks due to their effectiveness and efficiency in word representation learning.

As each dimension of an word embedding represents a latent feature of the word, it was found that word embeddings can capture useful syntactic, semantic properties, and meaningful latent linguistic regularities in a very simple way. For example, word embedding operations $V(Paris) - V(France) + V(Italy)$ result in a vector that is very close to $V(Rome)$ [40], [41]. Constant vector offset between a pair of words may share a particular relationship implying meaningful regularities. Moreover, some word embeddings can jointly represent the probability of word sequences from natural text. Significantly, word embeddings can surprisingly effectively express the semantic similarity between two words. Word embedding have been successfully at improving and simplifying many NLP tasks, such as Named Entity Recognition [35], sentiment analysis [36], part-of-speech (POS) [42], relation extraction [43], etc.

III. WORD EMBEDDING BASED STEGANALYSIS

Word embeddings reveal context information and contain rich semantic information, thus, the suitability of a synonym in the context can be effectively measured by exploiting word embeddings, which is benefit for linguistic

steganalysis task. By taking advantages of word embeddings, we propose a linguistic steganalysis method to leverage the word-level semantics, correlations within word embeddings to aid us in modeling the changes caused by synonym substitutions, and further enhance the detection performance against SS steganography.

A. FRAMEWORK OF THE PROPOSED STEGANALYSIS

Existing linguistic steganalysis methods mainly mined detection features from word frequencies, word co-occurrence analysis, the n-gram model, etc. However, these features are very low-dimensional and shallow, failing to accurately capture the semantic information and the semantic correlations between words. Thus, they cannot well perceive the semantic differences brought by synonym substitutions during embedding information.

To solve the problem of effectively and efficiently capturing the semantic information of words affected by SS-steganography, this paper introduces word embeddings for modeling the suitability of a synonym in the certain context. Then, the detection features are extracted for differentiating stego text and normal one. The detection of stego texts can be regarded as a classification problem, i.e. classifying the stego texts from normal ones, so a large-scale training stego and cover texts should be prepared for training a classifier in the training process. With the trained classifier, a test text will be determined whether it is a normal or a stego one using the same way to extract detection features. In this paper, SVM(Support Vector Machine)are employed as the classifier. Fig. 1 illustrates the framework of the proposed method. From Fig. 1, we can observe that the proposed steganalysis can mainly divided into three parts:

1) WORD REPRESENTATION

Given large-scale text corpora, a distributed word representation learning model is firstly employed to learn a real-valued low-dimensional vector for every word in the vocabulary. In this paper, continuous skip-gram language model, a type of NNLM, is selected to produce word embeddings for downstream steganalysis task. For a training or test text in our steganalysis task, we first recognize all the occurring synonyms, and then find their synonymous words in the corresponding substitution sets, and the words in their context windows. Finally, we represent these related found words into word embeddings.

2) FEATURE EXTRACTING

As word embedding can be conveniently used to measure the semantic distance between two words, the word correlation between a synonym and each word in its context windows is estimated by using their word embeddings. At the same time, the TF-IDF value of each context word will be calculated for measuring its importance to current synonym. With the acquired word correlations and TF-IDF values, the context fitness of each synonym from the same substitution set is obtained under the same context window, to sensitively perceive differences between different substitution elements. Finally, three detection features are calculated from the characteristics of the calculated context fitness values in a text.

3) CLASSIFICATION

The extracted detection features of all training texts are fed into a SVM classifier. Then the test text can be classified through the trained classifier according to its detection features.

B. WORD EMBEDDING GENERATION

SS linguistic steganography substituted a synonym with its synonymous words from the corresponding substitution set to embed information into the cover text. To analyze the influence on the text caused by synonym substitutions, not only the statistical changes, but also the linguistic changes should be effectively captured, so the influenced words should be represented into an effective semantic vector to discover the differences between a stego and cover text.

We begin our steganalysis process with leaning the word embeddings with semantic regularities. We adopt a recently developed efficient distributed word representations model called Skip-gram language model to learn word embeddings. The skip-gram model, which is an unsupervised feature learning algorithm, trains large-scale corpora to gradually converge words with similar meanings to nearby areas in the vector space.

The goal of Skip-gram model is to learn word representations which are good at predicting a center word's neighborhood words within in a certain context window. Mathematically, Skip-gram model maximizes the average log probability:

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{-k\le j\le k, j\neq 0}\log p(w_{i+j}|w_i) \tag{1}$$

where $w_1, w_2, \cdots, w_N$ is a sequence of training words, and k is the size of the training context window. The probability $p(w_{i+j}|w_i)$, which is the core, is defined by using a softmax function:

$$p(w_{i+j}|w_i) = \frac{\exp\left({V'_{w_{i+j}}}^{T} V_{w_i}\right)}{\sum_{l=1}^{M}\exp\left({V'_{w_l}}^{T} V_{w_i}\right)} \tag{2}$$

where V_w and V'_w represent the input and output representation of w, respectively, and M is the size of the vocabulary.

Essentially, the Skip-gram model is a bag-of-words model. As the representation of each word reflects a weighted bag of context words those co-occur with it, the learned word embeddings can well characterize the semantic information of a word. Compared with the previously used neural network language models for learning word embeddings, the structure of Skip-gram model is very simple, and its training is extremely efficient and effective. Especially, the Skip-gram model does a better job for infrequent words. As partial synonyms used in the SS linguistic steganography have low frequencies, to learn better word embeddings for synonyms,

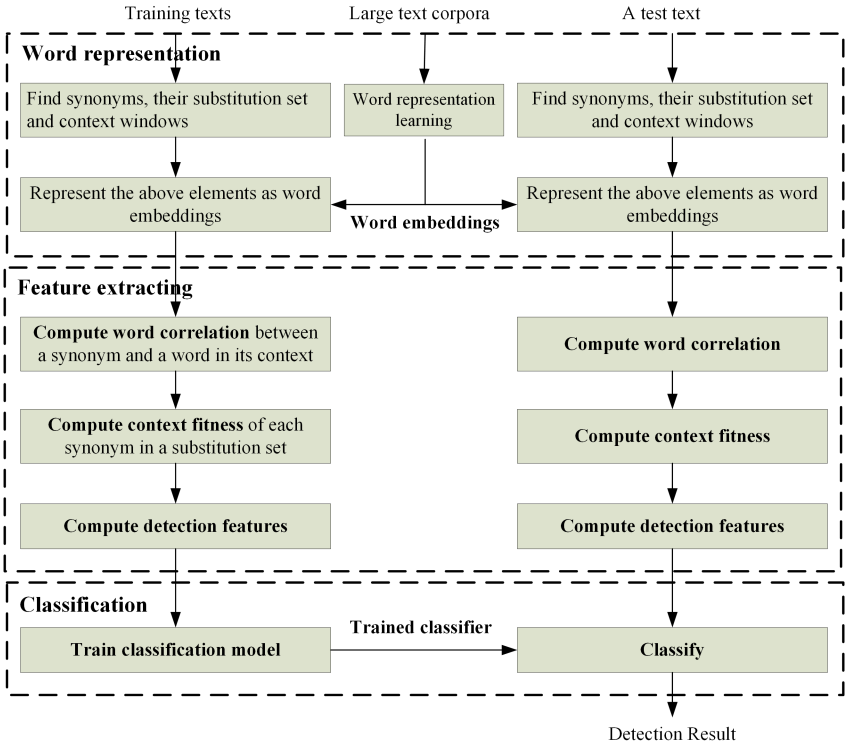


FIGURE 1. The framework of the proposed steganalysis.

the Skip-gram model is a more appropriate choose than CBOW in the task of linguistic steganalysis.

Having learned the word embeddings by using Skip-gram model to train large-scale unstructured text corpora, a word is represented to a word embedding, i.e., a low-dimensional vector. Given a word w, its word embedding in the form of vector is expressed as follows:

$$\overrightarrow{V}(w) = \{v_1, v_2, \cdots, v_m\} \tag{3}$$

where m denotes the dimension of the word embedding. As shown in [41], in general, m is suggested to be greater than 50. The larger m is, the higher the capability of capturing subtle semantics and syntactic information is. However, the computational complexity increases as the dimension of the word embedding increases. Therefore, choice of m should be overall considering several influence factors, e.g. the scale of the training corpus, computation speed, etc.

In general, each synonym substitution for embedding information in the SS steganography can only cause changes on context words, which surround the substituted synonym, thus, not all the words in the detected text are required to be represented by their word embeddings. Suppose that only the $2k$ words surrounding each substituted synonym have dependencies with the center word. These $2k$ words are defined to the context window of the center word. A synonym may be replaced by any one of its synonymous words in the process of embedding information. The synonymous words without appearing in a text are also important for steganalysis. Thus, to reduce the noise caused by useless words, only the synonyms in the text, their synonymous words in the corresponding substitution sets, and the words in their context windows are represented to their word embeddings learned by employing Skip-gram model.

By checking the words in a text using a prepared synonym dictionary, the synonyms are recognized. Given a synonym s in the detected text, its context window is denoted as $c(s)$.

$$c(s) = \{w_1, \cdots, w_k, w_{k+1}, \cdots, w_{2k}\} \tag{4}$$

where $w_1, \cdots, w_k$ is the k continuous words in front of s, while $w_{k+1}, \cdots, w_{2k}$ is the k continuous words after s. The substitution set of w_i locating in is denoted as $S(s) = \{sw_1, sw_2, \cdots, sw_x\}$, where $sw_1, sw_2, \cdots, sw_x$ are synonymous with s. The words related to s are all represented as their corresponding word embeddings illustrated by Fig. 2.

Take the sentence "He holds him with his glittering eye, the wedding guest stood still" as an example. With a prepared synonym dictionary for steganalysis, a synonym "glittering" is found, and its substitution set is $S(glittering) = \{glittering, aglitter, gliting, glittery\}$. Set the size of the context window $k = 5$, then words in its context window are "He", "holds", "him", "with", "his", "eye", "the", "wedding", "guest", "stood". The related words are all represented to word embeddings illustrated by Fig. 3.

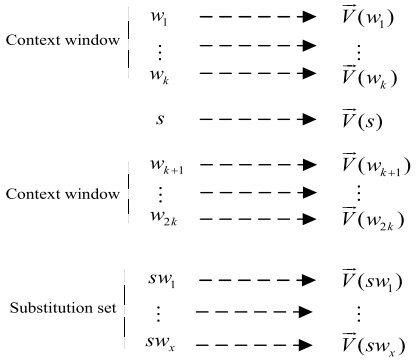


FIGURE 2. Represent all words related to a synonym as word embeddings.

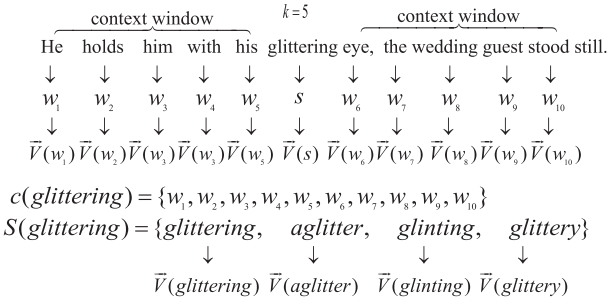


FIGURE 3. An example of representing related words as word embeddings.

C. CONTEXT FITNESS CALCULATION

Each related word is replacing with the corresponding m-dimensional word embedding. As the words whose word embeddings are close to each other have been proven to have similar semantic and syntactic properties, we directly measure the correlation between a synonym and one of its context words by cosine distance between the corresponding word embeddings.

Given a synonym s, its ith context word w_i, and their corresponding word embeddings $\overrightarrow{V}(s)$, $\overrightarrow{V}(w_i)$, the word correlation of s and w_i is calculated as follows:

$$cor(s, w_i) = \frac{\overrightarrow{V}(w_i) \cdot \overrightarrow{V}(s)}{\|\overrightarrow{V}(w_i)\| \cdot \|\overrightarrow{V}(s)\|} \tag{5}$$

The calculated correlation between a synonym and its context word can be exploited for effectively measuring the suitability of a synonym in the certain context. However, the importance of context words are unbalanced for the central word. For example, the function word and part of notional words, such as "a", "the", "of" and "who", which have high frequencies, always have less useful information than medium-frequency words such as "enhance", "refuse" and "stand". Thus, word correlations from different context words would make different contribution to measure the suitability of a synonym. Sometimes, an infrequently used word, which appears more frequently in the current document, can better express the meaning of the document than other frequently used word in the corpus, which has low frequency in the document.

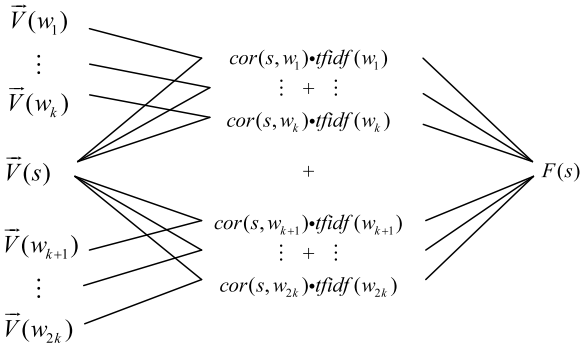


FIGURE 4. The process of context fitness calculation.

In order to reduce the interference caused by high-frequency words that have very less information, we use the TF-IDF method to assess the importance of a context word to the certain synonym. Accordingly, the context fitness of a synonym in a certain context is defined in this paper as:

$$F(s) = \sum_{i=1}^{2k} cor(s, w_i) \cdot tfidf(w_i) \tag{6}$$

where $tfidf(w_i)$ denotes the TF-IDF value of the word w_i in the text.

In any corpus and document, some words such as "the" and "a" have extremely high frequency, but they have very little useful information. These stopwords are insignificant for evaluating the context fitness. Therefore, we filter stopwords out of the context window by setting their TF-IDF values as 0. The process of calculating the context fitness of a synonym is illustrated by Fig. 4.

With the learned word embeddings, the context fitness of "glittering" in the above-mentioned example is calculated as follow:

$$\begin{aligned}
&F(glittering) = cor(glittering, he) \cdot tfidf(he) \\
&+cor(glittering, holds) \cdot tfidf(holds) \\
&+ cor(glittering, him) \cdot tfidf(him) \\
&+ cor(glittering, with) \cdot tfidf(with) \\
&+ cor(glittering, his) \cdot tfidf(his) \\
&+ cor(glittering, eye) \cdot tfidf(eye) \\
&+ cor(glittering, the) \cdot tfidf(the) \\
&+ cor(glittering, wedding) \cdot tfidf(wedding) \\
&+ cor(glittering, guest) \cdot tfidf(guest) \\
&+ cor(glittering, stood) \cdot tfidf(stood) \\
&= 0 + 0.1431 \times 0.0417 + 0 + 0 + 0 + 0.3869 \times 0.3418 + \\
&0 + 0.0997 \times 0.2491 + 0.0252 \times 0.1834 + 0.3275 \times 0.184 \\
&= 0.2279
\end{aligned}$$

D. FEATURE EXTRACTION

A cover synonym, which originally used in the cover text, may be replaced by its any synonymous words in the corresponding substitution set for embedding information. However, the context fitness values of different synonymous words are different in the same context. Thus, we should

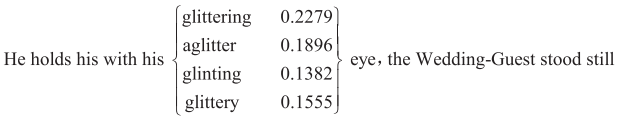


FIGURE 5. The context fitness results of synonymous words in the example.

calculate the context fitness of each substitution element in the same substitution set for extracting detection features.

Given the substitution set of synonym s is $S(s) = \{sw_1, sw_2, \cdots, sw_x\}$ and the context window is $c(s)$, then the context fitness values of the words in $S(s)$ are $F(sw_1), F(sw_2), \cdots, F(sw_x)$ respectively. By comparing these x context fitness values, we denote their maximum as $F^{max}(s)$.

$$F^{max}(s) = max(F(sw_i)) \tag{7}$$

where $1 \leq i \leq x$. For the above-mentioned sentence in the example, it is a normal sentence without being modified, and the synonym ''glittering'' is original. After calculating the context fitness of ''glittering'', we replace ''glittering'' with a synonymous word in its substitution set, and calculate the context fitness of each substituted synonym in the current same context, respectively. The results are shown in Fig. 5.

As shown in Fig. 5, in the same context, the context fitness will vary with the synonymous words. In general, the original word in a normal text should be the most appropriate word for the current context, its context fitness will be largest compared with those of its synonymous words. Therefore, when an original synonym in the cover text is replaced by its synonymous word for embedding information, the new word would not be very appropriate to the current context compared with the original word, leading to a relatively small context fitness. For example, the original ''glittering'' has the highest context fitness 0.2279 in the example sentence. When it is replaced by ''aglitter'',''glinting'', or ''glittery'', the context fitness would be significant reduced, which are consistent with human judgments of the suitability of a word.

Meanwhile, the results of Google searching these four synonymous words are as following: ''glittering'': 39,900,000; ''aglitter'': 7,580,000; ''glinting'': 2,290,000; ''glittery'': 43,500,000. If we take these results to measure the suitability of a word in a certain context, it may mistakenly regard ''glittery'', which appears in most results than other three synonymous words, as the original used word in the example sentence. However, it is not the most appropriate word in the current context. It can be easy found that word embeddings can more accurately measure the context fitness than using word frequencies.

With the above analysis, if the context fitness of a synonym in the detected text is not the maximum of context fitness values of all the words in its substitution set, then it is doubtful that this word may not be the original one, which may be a stego word employing for embedding secret message. If there are much more synonyms in the detected text are in this case, then the detected text is a stego text with a very high probability.

On the other hand, the context fitness values of words in the substitution set have significant differences, the larger the context fitness of a synonym is, the smaller the probability of a synonym being a stego one is. In most cases, a synonym in the cover text has larger context fitness value than its stego synonymous word in the corresponding stego text. Therefore, we extract features to detect the stego texts by comparing the context fitness of each synonym appearing in the detected text with the maximum context fitness of the words in its substitution set, and analyzing the mean of the calculated context fitness values.

Suppose all the synonyms sequentially appearing in a text are $\{s_1, s_2, \cdots, s_n\}$. For convenience, the context fitness of the ith synonym s_i and the maximum context fitness of the words in the substitution set of s_i are denoted as F_i and F_i^{max}, respectively. Then, three detection features: Context Fitness Maximum Rate (CFMR), Context Fitness Maximum Deviation (CFMD), and Context Fitness Mean(CFM) are defined by the following equations:

$$CFMR = \frac{1}{n}\sum_{i=1}^{n}[F_i = F_i^{max}] \tag{8}$$

$$CFMD = \frac{1}{n}\sum_{i=1}^{n}(F_i - F_i^{max})^2 \tag{9}$$

$$CFM = \frac{1}{n}\sum_{i=1}^{n}F_i \tag{10}$$

where

$$[F_i = F_i^{max}] = \begin{cases} 0 & F_i = F_i^{max} \\ 1 & F_i \neq F_i^{max} \end{cases}$$

Generally speaking, the context fitness values from the same synonym set is unbalance. We find that the CFMR of a normal text tends to be greater than that of its stego text, since an original synonym always has the highest context fitness than its synonymous words. In most cases, in the normal text, $F_i = F_i^{max}$ holds. While for embedding information into a normal text, more synonyms with highest context fitness values have been replaced by their synonymous words with lower context fitness values, while fewer synonyms with lower context fitness values are replaced by their synonymous words with highest context fitness values. Thus, the amount of synonyms which have maximum context fitness in its synonym set is reduced by synonym substitutions for embedding secret message. We then infer that the CFMR of a stego text is always smaller than that of its corresponding cover text. Obviously, the CFM has the similar trend as the CFMR.

On the other hand, if the context fitness of a synonym is different from the maximum context fitness of its substitution set, this synonym is likely a substituted stego word. By synonym substitutions, the amount of this kind of synonym in the stego text is more than that in the corresponding cover

text, thus the stego text should have a larger CFMD than its corresponding cover text.

In a word, the cover text should have larger CFMR, CFM and smaller CFMD than the stego text.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we demonstrate the effectiveness of the proposed method by extensive experiments. We compared our method with two baseline steganalysis methods on the same datasets for fair comparison. In addition, we investigate the influences made by the corpus and language model of learning word embeddings.

A. EXPERIMENTAL SETUP

1) THE TOOLKIT OF LEARNING WORD EMBEDDINGS

In experiments, we employ word2vec[1] for learning word embeddings. It is an effective and efficient toolkit to implement Skip-gram model and CBOW model. In the Word2vec, there has a parameter called min_count using for filtering out unusual words. If the frequency of a word is less than the preset min_count, it will be filtered out for learning word embeddings. In fact, some synonyms used in steganography and steganalysis have very low frequencies. In order to acquire the word embeddings of these infrequent used synonyms, we set the min_count to 1.

2) THE CORPORA FOR LEARNING WORD EMBEDDINGS

In the task of natural language processing, the corpus always plays an important role. There are several corpora publicly available, while the efficient and quality of the word embedding vary with the employed language model and training corpora. As the word embedding has a big influence on detection performance of our proposed method, the involved experiments of learning word embeddings are carried out on two corpora.

- *GoogleNews corpus*. Google has a tremendous Google News dataset and pre-trained word embeddings on part of this dataset, which is about 100 billion words. By training the Skip-gram model with Word2vec, 300-dimensional word embeddings for 3 million words and phrases were obtained and published, which can be directly downloaded from the Internet.[2]
- *Gutenberg corpus*. we downloaded more than 40 thousand texts from Project Gutenberg[3] on March 21, 2016 to form a Gutenberg corpus for our experiments. Gutenberg corpus consists of 20.5GB English free eBooks, which are famous literatures in the world. The size, style and theme of Gutenberg corpus are different from GoogleNews corpus.

[1] https://code.google.com/archive/p/word2vec/

[2] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit

[3] https://www.gutenberg.org/

3) THE DATASETS FOR STEGANALYSIS TASK

In order to evaluate the performance of the involved steganalysis methods, several synonym substitution-based steganographic methods are employed to generate stego texts. Thus, we generate 6 stego text datasets from a same cover text dataset with different embedding rates or different steganograpyhic methods. The employed datasets for testing the involved steganalysis methods are described as follows:

- *cover text set*.
  5000 texts were selected randomly from our Gutenberg corpus as cover texts. Cover texts are embedded random secret information by different linguistic steganographic methods.
- *stego text sets 1-4: Tlex-25%, Tlex-50%, Tlex-75% and Tlex-100%*.
  The T-Lex steganographic tool [18] is the only available SS steganography system on the Internet, thus to our knowledge, all the current linguistic steganalysis against SS steganography mainly attacked T-Lex or its variants. Note that in our implementation, instead of embedding secret message with arbitrary length, we modify the code of T-Lex using specified embedding rate. This is because detection performance varies with embedding rates. A steganalysis method always achieves stronger capability of detecting stego texts with high embedding rate than those with low embedding rate. The first stego text set denoted as Tlex-25% is generated by using T-Lex to embed random secret message into each cover text with the embedding rate of 25%. In a similar way, the stego text sets of Tlex-50%, Tlex-75% and Tlex-100% are generated by T-Lex with embedding rate of 50%, 75% and 100%, respectively.
- *stego text set 5: MC*.
  We implement the Matrix-coding-based(MC-based) steganographic method in [22] for verifying the detection performance of steganalysis methods. MC-based method can achieve higher embedding efficiency than T-Lex tool, so that the security of stego texts generated by MC-based method are improved. However, MC-based method must embed message with certain embedding rates, which should be $k/(2^k\text{-}1)$, (k=2,3,...). Namely, every 2^k-1 cover synonyms are embedded into k bits secret message. In our experiments, we fix $k = 3$ to generate stego texts. The generated stego text set is denoted as MC.
- *stego text set 6: STC*.
  Moreover, the STC-based method in [23] is also implemented with STC simulator based on the publicly available codes for comparison. STC-based method minimized distortion caused by synonym substitutions to achieve high secure level. The stego text set generated by STC-based method with random secret message is denoted as STC.

TABLE 1. The detection results of different steganalysis methods.

	PP			NRF			WES-Google-skip		
	Precision	Recall	F-value	Precision	Recall	F1 value	Precision	Recall	F-value
Tlex-25%	85.22%	69.20%	76.38%	90.82%	76.86%	83.26%	93.26%	88.57%	90.85%
Tlex-50%	94.12%	84.43%	89.01%	97.62%	94.56%	96.07%	98.95%	97.03%	97.98%
Tlex-75%	97.64%	92.53%	95.02%	99.02%	98.00%	98.51%	99.90%	98.87%	99.38%
Tlex-100%	98.46%	96.53%	97.49%	99.52%	98.66%	99.09%	100.00%	99.40%	99.70%
MC	95.63%	70.90%	81.43%	94.39%	74.03%	82.98%	98.55%	90.37%	93.98%
STC	60.66%	80.48%	69.18%	81.84%	90.36%	85.89%	94.46%	90.97%	92.68%
Average	88.62%	82.35%	84.75%	93.87%	88.75%	90.96%	97.57%	94.07%	95.76%

TABLE 2. The detection performance of our steganalysis for different trained corpora.

	WES-Google-skip			WES-Gutenberg-skip			WES-Gutenberg-CBOW		
	Precision	Recall	F-value	Precision	Recall	F-value	Precision	Recall	F-value
Tlex-25%	93.26%	88.57%	90.85%	93.54%	88.87%	91.15%	94.28%	87.50%	90.76%
Tlex-50%	98.95%	97.03%	97.98%	98.98%	97.40%	98.18%	98.95%	97.13%	98.03%
Tlex-75%	99.90%	98.87%	99.38%	100.00%	99.23%	99.61%	99.92%	98.59%	99.25%
Tlex-100%	100.00%	99.40%	99.70%	100.00%	99.83%	99.91%	100.00%	99.69%	99.84%
MC	98.55%	90.37%	93.98%	98.88%	91.00%	94.78%	98.54%	88.43%	93.21%
STC	94.46%	90.97%	92.68%	95.24%	93.30%	94.26%	95.49%	90.85%	93.11%
average	97.57%	94.07%	95.76%	97.77%	94.94%	96.32%	97.86%	93.70%	95.70%

4) PERFORMANCE MEASURE

In order to accurately evaluate the reliability of the steganalysis methods, we measure the detection results of the involved methods through three performance measures: *precision*, *recall* and *F-value* [44], which are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F - value = (1 + \beta^2) \cdot \frac{Recall \cdot Precision}{\beta^2 \cdot Recall + Precision} \tag{13}$$

where TP is the number of stego texts correctly predicted, FP is the number of cover texts incorrectly predicted as stego ones, FN is the number of stego texts incorrectly predicted as cover ones, TN is the numbers of cover texts correctly predicted. Obviously, in the field of steganalysis, we should pay more attention to *recall*, which measures the fraction of stego texts correctly predicted over all stego texts. However, the goals of *recall* and *precision* can be often conflicting. Thus, *F-value* metric is employed to represent the trade-off between *recall* and *precision*. The parameter *beta* corresponds to the relative importance of *recall* vs *precision*. In this paper, it is set to 1.

B. DETECTION PERFORMANCE COMPARISON FOR DIFFERENT STEGANOGRAPHIC METHODS

For convenience, the proposed steganalysis based on word embedding is abbreviated to WES. First, WES employs word embeddings learned by Skip-gram model from GoogleNews corpus(WES-Google-skip) for detecting stego texts in different stego text sets. For each stego text set, 3000 stego texts are selected and combined with 3000 cover texts for training, and the remaining are used for testing. In the process of classifying stego and cover texts, WES utilizes SVM with RBF kernel as the classifier. 5-fold cross validation is chosen for measuring classification performance. For comparison, the steganalysis methods in [32] (named NRF) and in [30] (named PP) are implemented as baseline methods.

The detection results are listed in Table 1. The experimental results demonstrate that WES-Google-skip can capture the subtle semantic changes of the stego texts successfully, thanks to the effectiveness of the learned word embeddings. As shown in Table 1, WES-Google-skip performs best for all stego text sets. It can more effectively detect different SS steganographic methods than other two baseline steganalysis methods PP and NRF.

WES-Google-skip can accurately detect Tlex stego texts with various embedding rates, especially when the rate is greater than 50%. The performances of PP and NRF are not satisfactory when the embedding rate of Tlex stego texts is 25%. Meanwhile, for detecting MC and STC stego texts, WES also perform very well. Its precision, recall and F-value are all larger than those of PP and NRF. For all stego text sets in the experiments, WES-Google-skip achieves over 90% F-value. It improves the average F-value of PP by 11.01%, that of NRF by 4.8%. While the average precision and recall are oustrips PP method by 8.95% and 11.72%, NRF method by 3.7% and 5.32%, respectively. Moreover, from the results in Table 1, we can see that different SS steganographic methods do not have a significant impact on our method, our method has good generalization.

C. DETECTION PERFORMANCE COMPARISON FOR DIFFERENT LEARNED WORD EMBEDDINGS

The same language model will learn word embeddings with different qualities from different corpora. For the same corpus, different word embeddings will be learned by different models. In order to investigate the effectiveness of the word embeddings for steganalysis, some more experiments

are conducted. Firstly, we learn word embeddings by word2vec with Skip-gram model from the Gutenberg corpus. The proposed WES using these new word embeddings is denoted as WES-Gutenberg-skip. As the Skip-gram model is better for infrequent words and negative sampling with low dimensional vectors, word embeddings with only 100 dimensions are learned from Gutenberg corpus. Secondly, the CBOW model in word2vec is instead of Skip-gram model for learning word embeddings with the same parameters. The steganalysis with word embeddings generated by CBOW model from Gutenberg corpus is denoted as WES-Gutenberg-CBOW.

Since the used cover texts in the experiments are all selected from the Gutenberg corpus, semantic and statistical characteristics captured by word embeddings from Gutenberg corpus may be consistent with those in the stego texts. In terms of distinguishing stego texts, better word embeddings are learned from Gutenberg corpus than from GoogleNews corpus, thus WES-Gutenberg-skip will achieve better detection performance than WES-Google-skip, which can be demonstrated by the detection results listed in Table 2. For all the kinds of stego texts, WES-Gutenberg-skip have larger precision, recall, and F-value thanWES-Google-skip. WES-Gutenberg-skip outperforms WES-Google-skip by 0.2%, 0.87%, and 0.56% in terms of average precision, recall and F-value, respectively. It can demonstrate that WES-Gutenberg-skip can capture more statistical differences between stego and cover texts than WES-Google-skip, namely, WES-Gutenberg-skip works better than WES-Google-skip. Therefore, we can deduce that the detection performance of the proposed method can be further improved by using particular corpus to learn better word embeddings, whose derivation is consistent with the cover texts.

In order to study the detection performance over different embeddings from different language models, we compare WES-Google-skip with WES-Google-CBOW. The involved experimental results are shown in Table 2. As we can see from Table 2, CBOW model does not perform better than Skip-gram model on the steganalysis task, which confirms to our previous conclusions.

Overall, high quality word embeddings will lead to good detection performance under the same steganalysis model. We can further improve our proposed steganalysis method by acquiring better word embeddings.

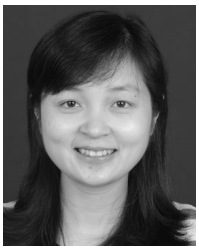
V. CONCLUSION

In this paper, we propose a linguistic steganalysis method base on word embedding for detecting the stego texts generated by synonym substitutions. The word embeddings are learned by Skip-gram language model for measuring the correlation between a synonym and its context. Then, the fitness of a synonym in the context is evaluated by the correlations of the synonym and its context words, while the correlations are weighted by TF-IDF method. The context fitness values of related synonyms are employed to extract detection features for discriminating stego texts and normal ones by using SVM as a classifier. The experimental results show that the proposed steganalysis method can effectively detect the synonym substitution-based steganography and has good generalization. Especially, using particular corpus or good language model to learning word embeddings with high quality, the detection performance can be further improved.

REFERENCES

[1] J. Torres-Sospedra *et al.*, "Enhancing integrated indoor/outdoor mobility in a smart campus," *Int. J. Geograph. Inf. Sci.*, vol. 29, no. 11, pp. 1955–1968, 2015.
[2] R. Meng, S. G. Rice, J. Wang, and X. Sun, "A fusion steganographic algorithm based on faster R-CNN," *CMC-Comput. Mater. Continua*, vol. 55, no. 1, pp. 1–16, May 2018.
[3] Y. Zhang, C. Qin, W. Zhang, F. Liu, and X. Luo, "On the fault-tolerant performance for a class of robust image steganography," *Signal Process.*, vol. 146, pp. 99–111, May 2018.
[4] L. Wang, C. Yao, Y. Yang, and X. Yu, "Research on a dynamic virus propagation model to improve smart campus security," *IEEE Access*, vol. 6, pp. 20663–20672, 2018.
[5] Y. L. Liu, H. Peng, and J. Wang, "Verifiable diversity ranking search over encrypted outsourced data," *CMC-Comput. Mater. Continua*, vol. 55, no. 1, pp. 37–57, May 2018.
[6] C. Yang, X. Luo, J. Lu, and F. Liu, "Extracting hidden messages of MLSB steganography based on optimal stego subset," *Sci. Chi. Inf. Sci.*, vol. 61, p. 119103, Nov. 2018, doi: 10.1007/s11432-017-9328-2.2018.
[7] W. Li, H. Liu, J. Wang, L. Xiang, and Y. Yang, "An improved linear kernel for complementary maximal strip recovery: Simpler and smaller," *Theor. Comput. Sci.*, to be published, doi: 10.1016/j.tcs.2018.04.020.
[8] Q. Feng, J. Hu, N. Huang, and J. Wang, "Improved PTAS for the constrained k-means problem," *J. Combinat. Optim.*, to be published, doi: 10.1007/s10878-018-0340-4.
[9] L. Xiang, G. Zhao, Q. Li, W. Hao, and F. Li, "TUMK-ELM: A fast unsupervised heterogeneous data learning approach," *IEEE Access*, vol. 6, pp. 35305–35315, 2018.
[10] X. Shen, F. Shen, Q.-S. Sun, Y. Yang, and Y.-H. Yuan, and H. Shen, "Semi-paired discrete hashing: Learning latent hash codes for semi-paired cross-view retrieval," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4275–4288, Dec. 2018.
[11] L. Xiang, X. Shen, J. Qin, and W. Hao, "Discrete multi-graph hashing for large-scale visual search," *Neural Process. Lett.*, to be published, doi: 10.1007/s11063-018-9892-7.
[12] X. Shen, W. Liu, I. Tsang, Q. Sun, and Y. Ong, "Multilabel prediction via cross-view search," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4324–4338, Sep. 2018, doi: 10.1109/TNNLS.2017.2763967.
[13] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2545–2557, Nov. 2017.
[14] R. Bergmair, "A comprehensive bibliography of linguistic steganography," *Proc. SPIE*, vol. 6505, pp. 65050W-1–65050W-6, Mar. 2007.
[15] W. Mazurczyk and K. Szczypiorski, "Trends in steganography," *Commun. ACM*, vol. 57, no. 3, pp. 86–95, 2014.
[16] A. Wilson, P. Blunsom, and A. Ker, "Detection of steganographic techniques on Twitter," in *Proc. Conf. Emp. Meth. Nat. Lang. Proc. (EMNLP)*, Lisbon, Portugal, Sep. 2015, pp. 2564–2569.
[17] R. Din *et al.*, "Performance analysis on text steganalysis method using a computational intelligence approach," in *Proc. Int. Conf. Electr. Eng., Comput. Sci. Inf. (EECSI)*, Palembang, Indonesia, Aug. 2015, pp. 67–73.
[18] K. Winstein. (1998). *Lexical Steganography Through Adaptive Modulation of the Word Choice Hash*. [Online]. Available: http://www.imsa.edu/~keithw/tlex
[19] C. Y. Chang and S. Clark, "Practical linguistic steganography using contextual synonym substitution and a novel vertex coding method," *Comput. Linguistic*, vol. 40, no. 2, pp. 403–448, 2014.
[20] A. Barmawi, "Linguistic based steganography using lexical substitution and syntactical transformation," in *Proc. Int. Conf. Converg. Secur. (ICS)*, 2016, pp. 1–6.
[21] L. Xiang, X. Wang, C. Yang, and P. Liu, "A novel linguistic steganography based on synonym run-length encoding," *IEICE Trans. Inf. Syst.*, vol. 100, no. 2, pp. 313–322, 2017.

[22] X. Yang, F. Li, and L. Xiang, "Synonym substitution-based steganographic algorithm with matrix coding," *J. Chin. Comput. Syst.*, vol. 36, no. 6, pp. 1296–1300, 2015.
[23] H. Hu, X. Zuo, W. Zhang, and N. Yu, "Adaptive text steganography by exploring statistical and linguistical distortion," in *Proc. IEEE 2th Int. Conf. Data Sci. Cyberspace (DSC)*, Shenzhen, China, Jun. 2017, pp. 145–150.
[24] L. Xiang, Y. Li, W. Hao, P. Yang, and X. Shen, "Reversible natural language watermarking using synonym substitution and arithmetic coding," *CMC-Comput. Mater. Continua*, vol. 55, no. 3, pp. 541–559, 2018.
[25] L. Xiang, W. Wu, X. Li, and C. Yang, "A linguistic steganography based on word indexing compression and candidate selection," *Multimedia Tools Appl.*, vol. 77, no. 21, pp. 28969–28989, 2018, doi: 10.1007/s11042-018-6072-8.
[26] Y. Ma, X. Luo, X. Li, Z. Bao, and Y. Zhang, "Selection of rich model steganalysis features based on decision rough set α-positive region reduction," *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: 10.1109/TCSVT.2018.2799243.
[27] C. M. Taskiran, M. Topkara, and E. J. Delp, "Attacks on lexical natural language steganography systems," *Proc. SPIE*, vol. 6072, pp. 607209-1–607209-9, Feb. 2006.
[28] Z. Yu, L. Huang, Z. Chen, and L. Li, "Detection of synonym-substitution modified articles using context information," in *Proc. Int. Conf. Future Gener. Commun. Netw. (FGCN)*, Hainan, China, 2008, pp. 134–139.
[29] Z. Chen, L. Huang, H. Miao, W. Yang, and P. Meng, "Steganalysis against substitution-based linguistic steganography based on context clusters," *Comput. Elect. Eng.*, vol. 37, no. 6, pp. 1071–1081, 2011.
[30] L. Xiang, X. Sun, G. Luo, and B. Xia, "Linguistic steganalysis using the features derived from synonym frequency," *Multimedia Tools Appl.*, vol. 71, no. 3, pp. 1893–1911, 2014.
[31] Z. Yu, L. Huang, Z. Chen, L. Li, X. Zhao, and Y. Zhu, "Steganalysis of synonym-substitution based natural language watermarking," *Int. J. Multimedia Ubiquitous Eng.*, vol. 4, no. 2, pp. 21–34, 2009.
[32] Z. Chen, L. Huang, and W. Yang, "Detection of substitutionbased linguistic steganography by relative frequency analysis," *Digit. Invest.*, vol. 8, no. 1, pp. 68–77, 2011.
[33] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[34] C. Zhu, L. Cao, Q. Liu, J. Yin, and V. Kumar, "Heterogeneous metric learning of categorical data with hierarchical couplings," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1254–1267, Jul. 2018.
[35] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. (NACACLHL)*, 2016, pp. 260–270.
[36] S. Poria, E. Cambria, and A. Gelbukh, "Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Lisbon, Portugal, Sep. 2015, pp. 2539–2544.
[37] G. E. Hinton, "Learning distributed representations of concepts," in *Proc. 8th Conf. Cognit. Sci. Soc. (CSS)*, Amherst, MA, USA, 1986, pp. 1–12.
[38] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003.
[39] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Chiba, Japan, Sep. 2010, pp. 1045–1048.
[40] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Lake Tahoe, NV, USA, 2013, pp. 3111–3119.
[41] T. Mikolov, K. Chen, G. Corrado, and J. Dean. (2013). "Efficient estimation of word representations in vector space." [Online]. Available: https://arxiv.org/abs/1301.3781
[42] B. Zadrozny, "Learning character-level representations for part-of speech tagging," in *Proc. Int. Conf. Mach. Learn. (ML)*, 2014, pp. 1810–1818.
[43] D. Zeng, Y. Dai, F. Li, R. S. Sherratt, and J. Wang, "Adversarial learning for distant supervised relation extraction," *CMC-Comput. Mater. Continua*, vol. 55, no. 1, pp. 121–136, 2018.
[44] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Proc. Eur. Conf. Princ. Pract. Knowl. Discovery Data*, Cavtat-Dubrovnik, Croatia, 2003, pp. 107–119.

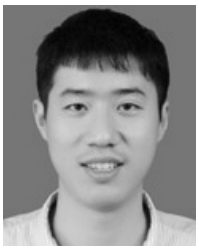


LINGYUN XIANG received the B.E. degree in computer science and technology and the Ph.D. degree in computer application from Hunan University, Hunan, China, in 2005 and 2011, respectively. She is currently a Lecturer with the School of Computer and Communication Engineering, Changsha University of Science and Technology. Her research interests include information security, steganography, steganalysis, machine learning, and pattern recognition.

JINGMIN YU received the B.E. degree from the Hunan Institute of Science and Technology, Hunan, China, in 2011. He is currently pursuing the M.S. degree in software engineering from the Changsha University of Science and Technology. His research interests include linguistic steganography and steganalysis.

CHUNFANG YANG received the B.S., M.S., and Ph.D. degrees from the Zhengzhou Information Science and Technology Institute in 2005, 2008, and 2012, respectively. He is currently a Lecturer with the Zhengzhou Science and Technology Institute. His current research interests include image steganography and steganalysis technique.

DAOJIAN ZENG received the Ph.D. degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, in 2015. He is currently a Lecturer with the School of Computer and Communication Engineering, Changsha University of Science and Technology. His research interests include natural language processing and information extraction. He received the Best Paper Award by COLING 2014 and CCL 2017.



XIAOBO SHEN received the B.Sc. and Ph.D. degrees from the School of Computer Science and Engineering, Nanjing University of Science and Technology, in 2011 and 2017, respectively. From 2015 to 2016, he visited The University of Queensland and the University of Technology Sydney for two years. He is currently a Research Fellow with the Rolls-Royce@NTU Corporate Lab, Nanyang Technological University. His primary research interests are multi-view learning, multi-label learning, and hashing.