electronics

MDPI

Article

A New Text Classification Model Based on Contrastive Word Embedding for Detecting Cybersecurity Intelligence in Twitter

Han-Sub Shin [1], Hyuk-Yoon Kwon [2,*] and Seung-Jin Ryu [3]

[1] Department of Industrial Engineering, Seoul National University of Science and Technology, 232 Gongneung-Ro, Nowon-Gu, Seoul 01811, Korea; sostkr@seoultech.ac.kr

[2] Department of Industrial Engineering, The Research Center for Electrical and Information Technology, Seoul National University of Science and Technology, 232 Gongneung-Ro, Nowon-Gu, Seoul 01811, Korea

[3] The Affiliated Institute of ETRI (Electronics and Telecommunications Research Institute), 1559 Yuseong-daero, Yuseong-gu, Daejeon 34044, Korea; sjryu@nsr.re.kr

* Correspondence: hyukyoon.kwon@seoultech.ac.kr

Received: 6 August 2020; Accepted: 12 September 2020; Published: 18 September 2020

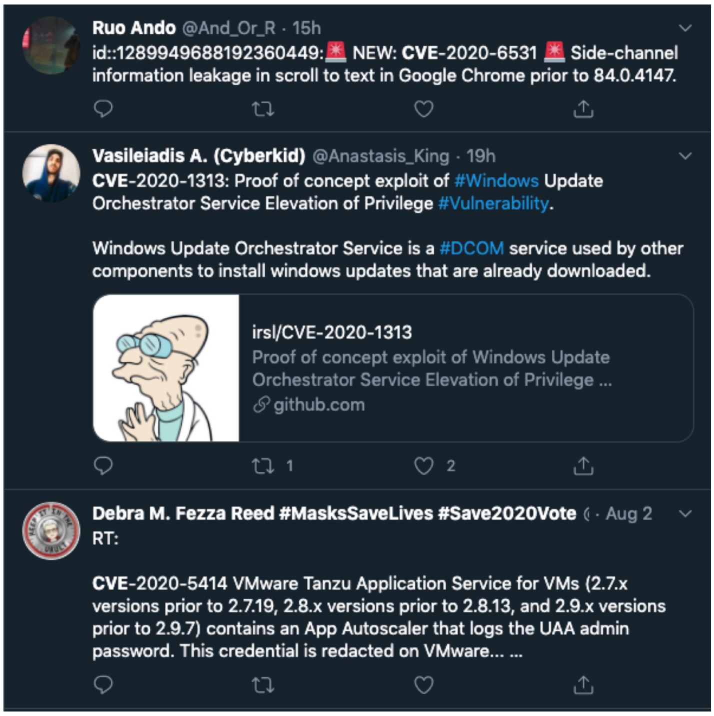
Abstract: Detecting cybersecurity intelligence (CSI) on social media such as Twitter is crucial because it allows security experts to respond cyber threats in advance. In this paper, we devise a new text classification model based on deep learning to classify CSI-positive and -negative tweets from a collection of tweets. For this, we propose a novel word embedding model, called contrastive word embedding, that enables to maximize the difference between base embedding models. First, we define CSI-positive and -negative corpora, which are used for constructing embedding models. Here, to supplement the imbalance of tweet data sets, we additionally employ the background knowledge for each tweet corpus: (1) CVE data set for CSI-positive corpus and (2) Wikitext data set for CSI-negative corpus. Second, we adopt the deep learning models such as CNN or LSTM to extract adequate feature vectors from the embedding models and integrate the feature vectors into one classifier. To validate the effectiveness of the proposed model, we compare our method with two baseline classification models: (1) a model based on a single embedding model constructed with CSI-positive corpus only and (2) another model with CSI-negative corpus only. As a result, we indicate that the proposed model shows high accuracy, i.e., 0.934 of F1-score and 0.935 of area under the curve (AUC), which improves the baseline models by 1.76∼6.74% of F1-score and by 1.64∼6.98% of AUC.

Keywords: cybersecurity intelligence; word embedding; deep learning; background knowledge; Twitter

1. Introduction

Twitter is a representative social media where users write their opinions and share events. It is known that more than 150 million users are wrote more than 500 million tweets per day as of 2019 [1]. To extract useful information from such a large number of tweets, the application of the text classification is necessary to condense the entire large-scale information into a small-scale subclass manageable for analysis. As a result, the text classification of tweets has been attempted in various domains to detect the information related to a specific topic. Typical examples are the geolocation prediction of the user by the classification of tweets containing the geolocation [2], the political affiliation prediction by the classification of tweets related to the politics [3], and the crime prediction by the classification of tweets based on the emotion [4]. However, the classification of tweets is inherently difficult because the length of tweets is limited, i.e., less than 280 characters, and various types of users are involved in writing tweets in an informal way [5].

In this paper, we aim to detect cybersecurity intelligence (CSI) from a collection of tweets. Cybersecurity is defined as the practice of defending computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks (e.g., phishing, ransomware, or data breaching), which abuse vulnerabilities in the systems [6]. To prevent the cyberattacks or minimize the damage from the attacks, it is important to collect the latest CSI from various data sources such as open-source intelligence (OSINT), human intelligence (HUMINT), or intelligence from the dark web. Among the data sources, Twitter is the representative OSINT data where CSI has been generated constantly [7]. That is, the security experts, system administrators, and hackers discuss technical details about cyber attacks and share their experiences [8]. Figure 1 shows examples of the tweets containing a commonly used CSI keyword, 'CVE' (i.e., Common Vulnerabilities and Exposures), which refers to publicly disclosed software vulnerabilities. As depicted in Figure 1, CSI-positive tweets include crucial information related to the cyber threats such as attackers, vulnerabilities, and targets. Therefore, by revealing CSI-positive tweets in advance, cybersecurity experts are able to respond to the corresponding cyber threats effectively. For this purpose, we classify the entire tweet data sets into positive and negative classes based on the relevance of CSI.



Ruo Ando @And_Or_R · 15h
id::1289949688192360449:🚨 NEW: **CVE**-2020-6531 🚨 Side-channel information leakage in scroll to text in Google Chrome prior to 84.0.4147.

Vasileiadis A. (Cyberkid) @Anastasis_King · 19h
CVE-2020-1313: Proof of concept exploit of #Windows Update Orchestrator Service Elevation of Privilege #Vulnerability.

Windows Update Orchestrator Service is a #DCOM service used by other components to install windows updates that are already downloaded.

irsl/CVE-2020-1313
Proof of concept exploit of Windows Update Orchestrator Service Elevation of Privilege ...
github.com

Debra M. Fezza Reed #MasksSaveLives #Save2020Vote · Aug 2
RT:

CVE-2020-5414 VMware Tanzu Application Service for VMs (2.7.x versions prior to 2.7.19, 2.8.x versions prior to 2.8.13, and 2.9.x versions prior to 2.9.7) contains an App Autoscaler that logs the UAA admin password. This credential is redacted on VMware... ...

Figure 1. The examples of the tweets containing cybersecurity intelligence.

There have been research efforts to detect CSI in tweets automatically based on machine learning or deep learning models. It has been known that a way of constructing an embedding model significantly affects on the classification performance [9]. As a result, two important issues have been dealt with: (1) the corpus definition used for constructing the embedding model and (2) the classifiers used for training the classification model. Ritter et al. have collected tweets containing keywords related to the cyber attack events and have proposed a weakly supervised learning model [10]. Le et al. have collected tweets containing CSI related keywords written by the selected Twitter accounts and have used CVE data set, which has been collected from National Vulnerability Database (NVD) (https://nvd.nist.gov/vuln/full-listing), for constructing the embedding model. They have trained two anomaly detectors using centroid and one-class Support Vector Machine (SVM) [11]. Chambers et al. have collected tweets containing the name of Distributed Denial of Service (DDoS) attacked organization and have performed time series analysis based on the frequency of tweets to forecast cyber threats [12]. Dionísio et al. have collected the tweets containing cyber threat intelligence and have labeled them manually. They have used a deep learning model based on convolutional neural network (CNN) to detect tweets related to cyber threats [13].

In this paper, we propose a new text classification model for classifying CSI-positive and -negative tweets by devising a novel embedding model. To constructing the embedding model, we consider two kinds of corpora: (1) CSI-positive corpus (e.g., tweets related to CSI) and (2) CSI-negative corpus (e.g., tweets unrelated to CSI). We do not have clear criteria to define the CSI-negative corpus because the subjects are widely distributed. In contrast, we can clearly define the CSI-positive corpus using a keyword set related to CSI such as "RCE", "CVE", or "Exploit" defined in data curated from cyber threat intelligence specialized companies (e.g., Recorded Future) or adopting well-defined public knowledge (e.g., CVE data set). Due to these different characteristics, when we use only CSI-positive corpora in constructing the embedding model, CSI-positive tweets are well classified, but CSI-negative tweets are not; when we use only CSI-negative corpus, it cannot extract distinct features for a specific class, and consequently, the overall classification performance is not satisfactory. These characteristics are supported by the experimental results conducted in this paper (See Section 6).

The contributions of the paper are summarized as follows:

1. We propose a novel word embedding model, contrastive word embedding, that enables to maximize the difference between base embedding models. We construct each embedding model using CSI-positive corpus or -negative corpus, which have completely different characteristics. Here, we utilize the background knowledge, which is well-defined public data set for a specific class, to supplement the imbalance of each tweet corpus. Hence, we use two kinds data sets for the CSI-positive corpora: (1) CSI-positive tweet data set and (2) CVE data set as the background knowledge; we use two kinds of data sets for the CSI-negative corpora: (1) CSI-negative tweet data set and (2) Wikitext data set as the background knowledge.
2. We devise a new text classification model based on the proposed word embedding model. We adopt deep learning models such as CNN or LSTM to extract adequate feature vectors from the embedding model and integrate the feature vectors into one classifier. To the best of our knowledge, none of the previous methods have considered both CSI-positive and -negative embedding models in one integrated classifier.
3. To validate the effectiveness of the proposed classification model, we compare it with two baseline models: (1) a model based on a single embedding model constructed with CSI-positive corpus only and (2) another model with CSI-negative corpus only. In the experiment, we use 70,000 tweets for CSI-positive and CSI-negative corpora as training data set, respectively, and 30,000 tweets for each corpus as testing data set, respectively. As a result, we indicate that the proposed model shows high accuracy, i.e., 0.934 of F1-score and 0.935 of AUC, which improves the baseline models by 1.76∼6.74% of F1-score and by 1.64∼6.98% of AUC.

The paper is organized as follows. In Section 2, we explain the background for the tweet classification. In Section 3, we review the related work. In Section 4, we present data sets and corpus used in this paper. In Section 5, we propose a new classification model based on contrastive word embedding. In Section 6, we present the experimental results to show the effectiveness of the proposed model. In Section 7, we discuss the characteristics of the proposed model. In Section 8, we conclude the paper.

2. Background

Text classification methods have been studied for various applications because they can be used to enhance the speed of decision-making and the automation of the analyzing process for the text data [14]. Over the past years, machine learning techniques (e.g., *k*-NN [15] and SVM [16]) have been investigated for the text classification. Recently, due to the huge success of deep learning in the field of natural language processing, there have been many research efforts that proposed text classifiers based on deep learning. Here, two considerable factors affecting the performance of the text classifier are as follows: (1) word representation with a corpus and (2) the deep learning model used for the classifier. In the following sections, we explain the details of each factor.

2.1. Word Representation

2.1.1. Word Embedding

Word embedding represents the words in the text in an R-dimensional vector space. It enables us to capture both the semantic and syntactic information of words from the vector space [17,18]. In the deep learning model, the effective word embedding model is essential as well for improving the accuracy of the text classification [9]. Term Frequency-Inverse Document Frequency (TF-IDF) has been widely used as a measure to represent the importance of the words in a document based on the simple statistics (i.e., frequency for each word) [19]. However, TF-IDF does not capture the semantic of the words in the document. Therefore, Mikolov et al. have proposed Word2Vec [20] that can represent the semantic similarity of words. It trains word embedding model considering similarity of words in local context windows. They have also devised a method of representing the text corpus by the phase instead of the word in Word2Vec [21]. Glove is another text representation using global word–word co-occurrence statistics to address the problem in Word2Vec, which does not consider the global co-occurrence [22]. FastText has been proposed to overcome the problem of Word2Vec, which cannot represent out of vocabulary words and infrequent words [23]. To this purpose, FastText breaks each word into a bag of character n-grams and use the sum of all the n-grams as a vector for the word. Li et al. have proposed a new embedding model where each element is composed of a pair of contradicting words (e.g., a pair of unimpressive and impressive words) [24]. The proposed model performs well in recognizing a contradicting relation between sentences. Liu et al. have shown that it is more effective to construct word embedding using the corpus related to a specific target topic than using the general corpus [25].

2.1.2. Multiple Word Embedding

To improve the performance of a single embedding model, there have been research efforts to utilize multiple embedding models. Bruni et al. have proposed multiple embedding models defined from different data models, i.e., textual models and visual models [26]. Luo et al. have shown that the performance of text similarity measurement becomes better when we use multiple word embedding instead of a single word embedding [27]. Zhang et al. have shown that the performance of multiple word embedding from various text representations, i.e., Word2Vec, Glove, and Synthetic, is better than that of each single embedding [28]. Ren et al. have proposed multiple word embedding where one is constructed with original texts and the other with the background knowledge [29].

2.1.3. Background Knowledge for Embedding

The performance of deep learning models tends to be degraded when only a target data set is used as the corpus for training the embedding model because the training data set cannot encompass all the characteristics of the target class due to its imbalanced characteristic [29]. To resolve this imbalance of a target data set, a method of incorporating the background knowledge, which utilizes the external reliable data set for supplementing the target class, has been proposed [30,31]. In particular, tweets tend to contain insufficient information for the classification because it has informal expression and a limited length of the text [5]. In this paper, we also define the background knowledge for CSI-positive and -negative data set and show their effectiveness compared to a target data set (i.e., tweets).

2.2. Text Classification with Deep Learning

In the text classification based on deep learning, once the text corpus is represented in the embedding model, its outputs are fed into the classifier based on deep learning. In the following sections, we introduce the representative deep learning models, which will be used in the proposed classification model.

2.2.1. CNN

LeCun et al. have proposed CNN for aiming on image recognition [32]. The basic idea of CNN is capturing a feature of data by moving the kernel, a convolution matrix, to a region in the image. Generally, while neural networks do not maintain spatial information in the image, CNN can maintain it by applying the kernel into each region of the image. For natural language processing (NLP), we can also apply the convolutional layer of CNN to the vector space converted from the text corpus. Because each kernel can learn the embedding on a region (i.e., one sentence in NLP) and capture the semantic and structural features in the sentence, CNN performs well in the text classification. Xu et al. have adopted a CNN model in classifying news data using two kinds of embedding models: topic-based word embedding and Word2Vec [33]. In addition, Kim et al. have shown the effectiveness of CNN in short text classification on movie reviews [34]. Hu et al. have presented a new method, SVMCNN, for the short text classification, which combines CNN and SVM [35]. Specifically, they have used CNN as the feature extractor of short texts and SVM as the classifier, and SVMCNN shows a better performance than each of CNN and SVM. Liu et al. have proposed an attention-gated CNN for the sentence classification by generating attention weights from the feature's context windows before the pooling layer [36], which shows a better performance than standard CNN models.

2.2.2. RNN and LSTM

Elman et al. have proposed recurrent neural network (RNN) for sequential data processing such as voice and text processing [37]. The distinguishing feature of RNN, which is different from general neural networks, is the introduction of the hidden state vector. The hidden state describes the summary of the previous input data, and it is updated whenever the new input comes in. Eventually, after processing all the input data, the hidden state is a summarization of the entire sequences, which is quite similar to the processing of a sequence performed by a human being. Naturally, RNN has the advantage when processing the sentences that are read by a person. However, as the layer becomes deep, gradient exploding and vanishing problems occur, which degrades the performance. To avoid them, Long Short-term Memory (LSTM) has been proposed [38]. To prevent gradient exploding and vanishing problems, LSTM adds the cell state so as to adjust previous information. LSTM has been widely used for the text classification because it can learn high-level representation using a deeper layer due to the cell state while preserving the sequence order of representations, which is provided by RNN. Wang et al. have applied LSTM to the sentiment classification of short texts on social media [39]. Ding et al. have proposed a densely connected Bi-LSTM consisting of multiple Bi-LSTM layers [40], which shows a better performance than Bi-LSTM.

3. Related Work

3.1. Tweet Classification

A number of studies have been proposed to apply machine learning and deep learning techniques to the tweet classification. First, for the machine learning techniques, Sriram et al. have proposed the classification method based on the Naïve Bayes classifier using the account information additionally to improve the classification accuracy in the tweet [41]. Alsmaddi et al. have proposed a new term weighting scheme and have evaluated nine term weighting schemes containing the proposed scheme for extracting characteristics of tweets [42]. Then, they have applied machine learning techniques such as SVM and k Nearest Neighbor (k-NN) to the corpus made based on each term weighting. Second, for the deep learning techniques, Wang et al. have applied LSTM to classify Internet Movie Database (IMDB) reviews by the sentiment [39]. They have shown that the performance of the deep learning model is better than that of the machine learning method such as Naïve Bayes and Extreme

Learning Machine (ELM). Zhou et al. have attempted to concatenate the CNN and LSTM for utilizing their strengths [43]. That is, CNN is used to extract a sequence of the phase representation, and then, LSTM is used to learn the sentence representation from the output of CNN. Graph Convolutional Network (GCN) has been proposed, which is one of the effective graph neural networks that can capture neighbor information in the graph representation [44]. Yao et al. have proposed a new GCN for the text classification using a single large graph consisting of word nodes and document nodes. Because it can jointly learn the embedding of words and documents, it shows a robust classification performance even with a small-labelled data [45]. Yang et al. have applied the capsule networks [46] to the text classification [47]. For this, they have devised dynamic routing strategies to alleviate the disturbance of some noise capsules.

3.2. Classification Using Background Knowledge

There have been research efforts utilizing the background knowledge in training embedding models in the text classification. Yang et al. have proposed a new topic model that combines lexical features obtained from two training data sets, i.e., Google Snippet and Ohsumed, and semantic features obtained from Wikitext, which is used as the background knowledge, for the topic classification of the short text [48]. Qureshi et al. have constructed a large graph of categories and articles in Wikipedia, which is used as the background knowledge [49]. This graph-based background knowledge shows a better performance than Word2Vec, FastText, and GloVe. Ren et al. have proposed a new text representation model that calculates the similarity between text data at Fundan University, which is the target data set, and the Chinese encyclopedia, which is the background knowledge [50]. Furthermore, Ren et al. have performed the feature fusion and decision fusion of multiple embedding models made from the target data set and the background knowledge [29].

Based on the investigations of these previous studies, we adopt background knowledge to improve the performance of classifying CSI-positive and -negative tweets. Especially, we focus on the definition of the background knowledge in building multiple word embedding models based on CSI-positive and -negative tweets. We also investigate the effects according to different fusion strategies (i.e., feature fusion and decision fusion) in designing the proposed embedding model (See Section 5.5).

3.3. Classification by the Cybersecurity Intelligence

Ritter et al. have proposed weakly supervised learning to detect cybersecurity events using tweets [10]. For the weakly supervised learning, they have annotated tweets containing the keyword "DDoS" out of the tweets written on the dates when the DDos attacks occurred as cybersecurity intelligence (CSI)-positive tweet data. Chambers et al. have proposed a framework to analyze the DDoS attack using tweets according to the following steps: (1) collection of tweets written on the day when the DDoS attack occurred, (2) training them using the basic neural network, (3) detection of attack events from the trained model, and (4) extraction of the attack topics by analyzing the user's response to the attack using an LDA-based model [12]. Zong et al. have collected tweets containing a keyword, "vulnerability", and have applied logistic regression to detect the existence of threats. They have also applied logistic regression and CNN to analyze the extent of threats and the user's opinions on social media about cyber threats [51].

Le Sceller et al. have proposed a new framework for discovering cybersecurity events in real time, called SONAR [52]. SONAR collects tweets containing CSI and clusters them into several groups using the first story detection algorithm [53] where each cluster is regarded as one event. It also identifies new keyword sets related to CSI considering the co-occurrence between the tweets in the Glove embedding model.

Dionísio et al. have proposed a method to classify tweets related to CSI in real time [13]. They have used CNN as the classifier to determine if a given tweet is related to CSI and have used BiLSTM to classify the tweets as one of six entities. They have shown that the proposed method outperforms the machine learning techniques such as SVM and Multi-Layer Perceptron (MLP).

Alves et al. have proposed a Twitter streaming threat monitor that continuously updates the summary of the threats related to a target infrastructure [54]. They have collected CSI-positive tweets based on the accounts and have extracted features using TF-IDF. Then, they have used MLP and SVM as the classifier for CSI-positive tweets.

Le et al. have proposed a new machine learning-based method for detecting CSI [11]. They have used CVE data set as the background knowledge and have adopted two novelty classifiers (i.e., centroid, one-class SVM) to detect tweets related to CSI. For extracting the features to the classifiers, Le et al. have used TF-IDF values obtained from the CVE data set. In contrast, in this paper, we use two representative deep learning models, i.e., CNN and LSTM, for both the feature extractor and the classifier, because the machine learning-based classifier requires the preceding feature extractor and its performance greatly depends on the feature extractor [55]. The detailed architecture of the proposed method is explained in Section 5.5. Especially, in our method, we focus on integrating multiple baseline learning models into one classifier. This implies that our framework can be easily extended to support the other deep learning or machine learning methods including this method by incorporating a new baseline learning model into the framework. As presented in Section 4, the F1-score of the proposed model is observed 0.932∼0.934 at maximum, which indicates quite high accuracy compared to the method proposed by Le et al. [11] where F1-score has been only 0.643, even considering different data sets.

Table 1 summarizes the comparison of previous studies on the classification by the CSI. It indicates that they have focused on the usage of the effective classifiers to improve the classification accuracy. None of them have used multiple word embedding models; only one study [11] has used the background knowledge for a single word embedding. In this paper, we focus on the design of a novel multiple word embedding, which is built in a contrastive way in terms of the relevance of CSI, and the definition of the background knowledge for the multiple word embedding.

Table 1. Comparison of previous studies on the classification by the cybersecurity intelligence (CSI).

Methods	Classifiers	Data Sets	Background Knowledge	Multiple Word Embedding
Ritter et al. [10]	Weakly supervised learning	Tweets containing "DDoS"	N/A	X
Chambers et al. [12]	Basic neural network	Tweets written on attack day	N/A	X
Zong et al. [51]	Logistic regression	Tweets filtered by keywords	N/A	X
Le Sceller et al. [52]	First story detection	Streaming Tweets	N/A	X
Dionísio et al. [13]	SVM, MLP, CNN, BiLSTM	Tweets filtered by keywords	N/A	X
Le et al. [11]	Centroid, One-class SVM, CNN, LSTM	Streaming Tweets	CVE descriptions	X
Alves et al. [54]	SVM, MLP	Tweets filtered by accounts and keywords	N/A	X

4. Data Sets and Corpus

In this paper, we use three kinds of data sets: (1) curated data, (2) OSINT data, and (3) background knowledge. Curated data are provided by the companies specialized in the field of the cybersecurity such as Recorded Future (https://www.recordedfuture.com/), Sensecy (https://www.sensecy.com/), and Surfwatch (https://www.surfwatchlabs.com/). In this paper, we obtain them from Recorded Future, and they are used to (1) identify Twitter accounts related to CSI and (2) filter tweets related to CSI from the collected tweets. We use tweets for OSINT data, which are publicly opened and can be crawled [56]. By leveraging the curated data, we classify a collection of the tweets into CSI-positive and

-negative classes. In addition, we use CVE data set and Wikitext data set as the background knowledge to construct embedding models. Table 2 shows the type, source, data set name, usage, and data size of each data set.

Table 2. The used data sets.

Types	Sources	Data Set Name	Usage	Data Size
Curated data	Recorded Future	RF Keyword Set	Filtering	639 keywords
		RF Account Set	Filtering	130 accounts
OSINT	Twitter	CSI-Positive Tweet Data Set	Embedding, Training, Testing	100,000 tweets
		CSI-Negative Tweet Data Set	Embedding, Training, Testing	100,000 tweets
Background knowledge	NVD	CVE Data Set	Embedding	134,166 identifiers
	Wikipedia	Wikitext Data Set	Embedding	28,475 articles

4.1. Curated Data

We acquire 1935 total accounts who have been written tweets contained in the “exploit” category analyzed by Recorded Future. The list of accounts is sorted by the number of tweets containing “exploit.” Among 1935 accounts, we define RF account set as the top 130 accounts. In addition, we define RF keyword set as the keywords in “industrial term” analyzed by Recorded Future. The total number of keywords included in the RF keyword set is 639, and the examples are as follows: “Internet-security”, “flaw”, “PoC”, “Exploit”, “RCE”, “Online Security Blog”, “CVE”, “Flash”, “Sandworm”, “Shellshock”, “Neutrino”, “Samba”, “Stagefright”, “Bin,” which clearly show the relevance to CSI.

4.2. CSI-Positive Tweet Data Set

We use cybersecurity intelligence (CSI)-positive tweet data set for word embedding, training, and testing of the model. Figure 1 shows an example of a CSI-positive tweet data set. To collect this data set, we use a web crawling framework, Scrapy (https://scrapy.org/). We collect all tweets written by the accounts in the RF account set. Since some of them could not be related to CSI, we filter only the tweets containing at least one keyword in the RF keyword set and define them as the CSI-positive tweet data set.

4.3. CSI-Negative Tweet Data Set

We use CSI-negative tweet data set for word embedding, training, and testing of the model. For this data set, we collect the tweets written by random users using the Twitter Streaming API (https://developer.twitter.com/en/docs/tweets/filter-realtime/overview) and define them as the CSI-negative tweet data set. Tweets written by random users are widely distributed in various topics. Thus, it is an adequate data set that has contrastive characteristics against CSI-positive tweet data set.

4.4. CVE Data Set

Common Vulnerabilities and Exposures (CVE) is a database for describing disclosed software flaws managed by the National Vulnerability Database (NVD) (https://nvd.nist.gov/vuln/full-listing). A CVE has the brief description of a cyber threat that includes the information about the affected product, versions, vendor, threat type, impact, method, and inputs of an attack. It is a formal document that has been officially verified. In this paper, we use the CVE data set as the background knowledge to supplement the imbalance of the CSI-positive tweet data set.

4.5. Wikitext Data Set

Wikitext data set (https://creativecommons.org/licenses/by-sa/3.0/) is a collection of tokens extracted from a set of verified articles on Wikipedia. For this, we use wikitext data with over 100 million tokens. One existing study has verified the performance of the Wikitext data set by showing that the performance of the language modeling using the Wikitext data set is superior to that of using Penn Treebank, which is a target data set [57]. In this paper, we use the Wikitext data set as the background knowledge to supplement the imbalance of the CSI-negative tweet data set.

5. The Proposed Classification Model for Detecting Cybersecurity Intelligence in Twitter

5.1. Basic Idea

We propose a new text classification model for detecting cybersecurity intelligence in Twitter based on a novel embedding model, contrastive word embedding. Figure 2 shows the overall framework of the proposed model based on contrastive word embedding. To construct the proposed embedding model, we define two completely different corpora according to the relevance of CSI: (1) CSI-positive data set and (2) CSI-negative data set. To confirm the conjecture that these two corpora have contrastive characteristics in terms of CSI, we investigate the CSI-positive ratio for each corpus using actual tweet data sets. Specifically, with CSI-positive corpus (i.e., 50,000 tweets) and -negative corpus (i.e., 50,000 tweets), we calculate the ratio of tweets containing the CSI related keywords (See the definition of RF keyword set in Section 4.1) out of the former corpus and that out of the latter corpus, respectively. As a result, we observe that 39.75% of tweets in the CSI-positive corpus and only 4.88% of tweets in the CSI-negative corpus have the CSI related keywords. This implies that the defined two corpora are contrastive in terms of CSI.

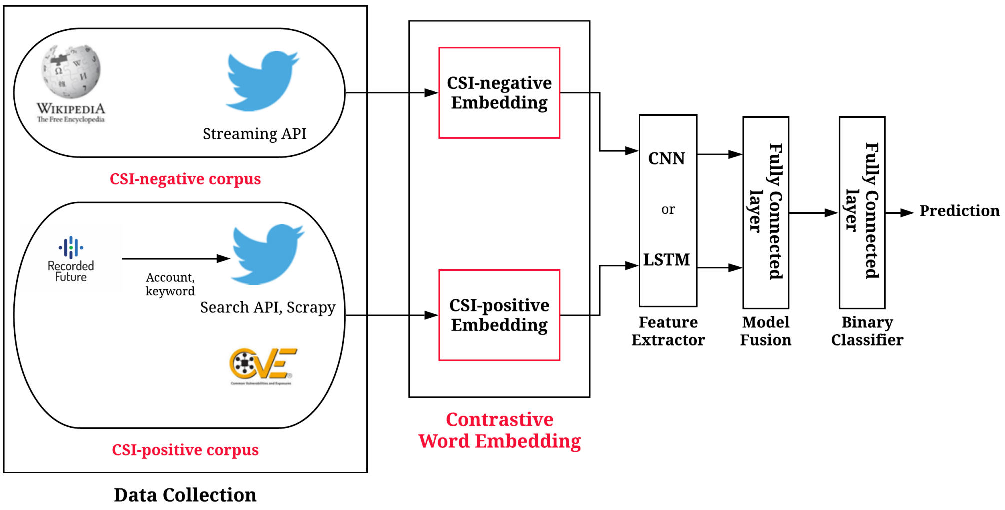


Figure 2. The overall framework of the proposed classification model based on contrastive word embedding.

5.2. Pre-Processing of Data Sets

Because tweets usually contain noises, pre-processing of the collected raw tweets is essential. In this paper, we adopt the existing pre-processing methods that have been commonly used in previous SNS text mining studies [13,58]. The used pre-processing techniques are as follows: (1) all characters except for those used in English are eliminated, (2) uppercase letters are converted to lowercase letters, and (3) tweet dependent properties such as @RT, tag, link, and punctuation are removed. In the case of the CVE and Wikitext data set, we also eliminate all characters except for those used in English.

When we construct the embedding model from a corpus, we use the top 5000 words based on the frequency in each corpus for every embedding model.

5.3. Annotation

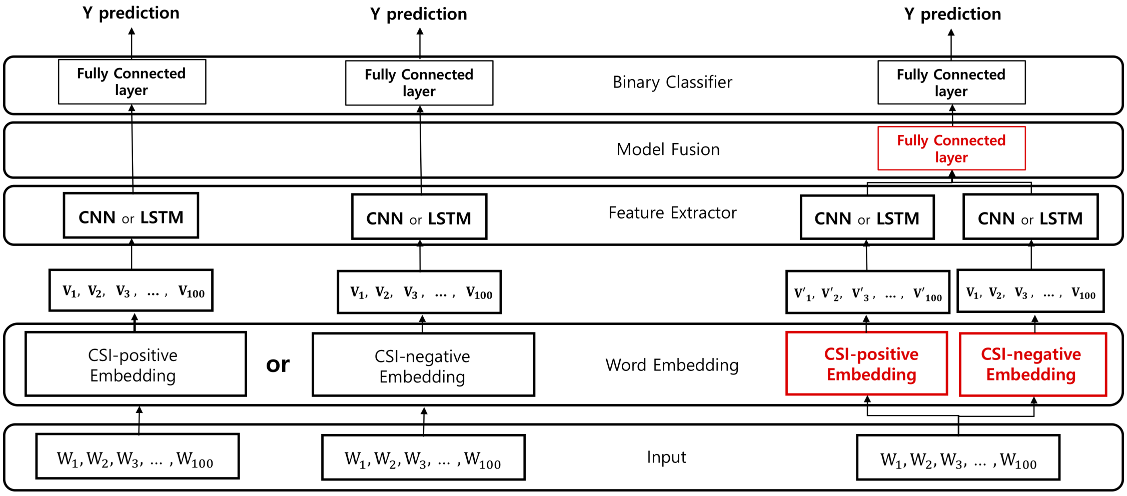
Table 3 shows the training and testing data sets used for the proposed method. We use a total of 200,000 tweets for training and testing. Specifically, for CSI-positive tweet data set, we use 70,000 tweets from the top 100 accounts in RF account set for training. We use 30,000 tweets from the remaining 30 accounts in RF account set for testing, separating the testing data set from the training data set. For CSI-negative tweet data set, among a total of 100,000 tweets collected from the Twitter streaming API, we use 70,000 tweets for training and the remaining 30,000 tweets for testing.

Table 3. Training, testing data.

Data Sets	Target Accounts	Number of Tweets	Usage
CSI-positive tweet data set	100 accounts	70,000	Training
	30 accounts	30,000	Testing
CSI-negative tweet data set	Random accounts	70,000	Training
	Random accounts	30,000	Testing

5.4. The Baseline Model

In this section, we present the baseline model based on each single word embedding that will be used as the basis of the proposed classification model. Figure 3a shows the architecture of the baseline model.



(**a**) (**b**)

Figure 3. The architectural difference between the baseline model and the proposed model. (**a**) The baseline model. (**b**) The proposed model.

Input layer: We conduct one-hot encoding of the input tweet text. We fix the length of the encoded bits as 100. That is, if the encoded bit length is larger than 100, we use the front 100 bits; otherwise, we pad 0.

Word embedding: For the baseline model, we use a single embedding layer, i.e., CSI-positive or -negative embedding on which we construct CNN or LSTM layers. We use four types of corpora (i.e., CSI-positive tweet data set and CVE data set for CSI-positive corpora; CSI-negative tweet data set and Wikitext data set for CSI-negative corpora). To construct all the embedding models, we randomly initialize output vectors of the model with the size of 100 dimensions.

Feature extractor: In the proposed model, we employ the feature fusion strategy for integrating feature vectors extracted from multiple learning models into a classifier (See Section 5.5). For this, we need to separate the feature extractor and the classifier because the concatenation of the features occurred between the feature extractor and the classifier. We also design each baseline model with the same strategy for comparison. That is, we use the deep learning model (i.e., CNN and LSTM) for the feature extractor. Then, we apply the classifier to the feature vectors extracted from the feature extractor. For constructing a CNN model, we use a Conv1D layer with 128 filters and a global max-pooling layer; For a LSTM model, we use 128 nodes in a hidden layer and tanh activation.

Binary classifier: We design a binary classifier based on a fully connected layer with sigmoid activation. We train the model so as to minimize the binary cross entropy.

5.5. Contrastive Word Embedding and Model Fusion

The main idea of the proposed model is devising a new classification model based on contrastive word embedding that can effectively integrate multiple word embedding models into one classifier. Figure 3b shows the architecture of the proposed model compared to the baseline model. Here, the distinguishing properties of the proposed model from the baseline model stem from two aspects: (1) the used embedding model and (2) the model fusion. In the embedding model, we use two multiple word embedding models having contrastive characteristics in terms of CSI, i.e., both CS-positive embedding and -negative embedding, for one classification model. In the model fusion, we concatenate the feature vectors extracted from the deep learning model based on each single embedding model and integrate them into one classifier. To focus on the effectiveness due to the contrastive embedding model and its fusion, we use the same base embedding models and deep learning models as in the baseline model.

In designing the model fusion, we consider two commonly-used fusion methods of multiple learning models: (1) the feature fusion and (2) the decision fusion [59]. The feature fusion is the concatenation of the feature values extracted from multiple models, and the entire feature values will be delivered into the succeeding step; the decision fusion determines the final decision values considering the feature values extracted from multiple models. In this paper, we employ the feature fusion because it maintains all the features extracted from two base embedding models and the succeeding step (i.e., the classifier) can utilize the interaction based on the difference between them. The decision fusion cannot consider the interaction between features from multiple learning models.

Figure 4 shows the concept of the feature fusion used in the proposed model. Specifically, the result of each embedding model is delivered into the deep learning model for the feature extraction. Then, we concatenate the features generated from multiple learning models. For this, we use a fully connected layer, as presented in Figure 3b, where ReLu activation is used and the unit (i.e., the dimensionality of the output space) is 8. As a result, the feature vectors are expanded so as to consider both CSI-positive and -negative corpora, maintaining both information in an integrated vector space. This leads to improving the accuracy of the classification between CSI-positive and -negative tweets.

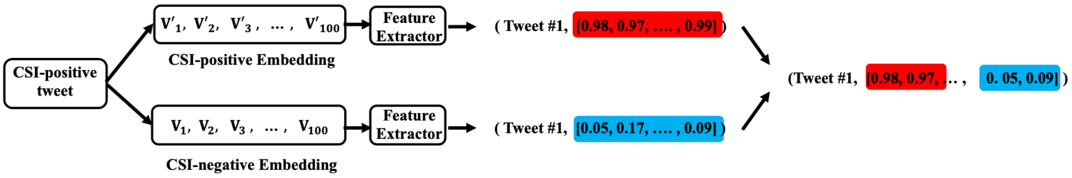


Figure 4. The concept of the feature fusion used in the proposed method.

6. Performance Evaluation

6.1. Experimental Methods and Evaluation Metrics

In the experiment, we aim to measure the classification accuracy of the baseline models and the proposed model. In the baseline model, if we choose CNN as the classifier, one of existing methods [13] is mapped to the methodology. To measure the accuracy of the classification, we use four evaluation metrics: (1) *Precision*, (2) *Recall*, (3) *F*1-*score*, and (4) AUC. *Precision* and *recall* are defined by four components in the confusion matrix: true positive (*TP*), false positive (*FP*), false negative (*FN*), and true negative (*TN*). *TP* and *TN* mean correct prediction; *FP* and *FN* wrong prediction. In our problem, *TP* (or *FP*) means that the model predicts a given tweet is positive to CSI, and it is actually positive (or negative); *FN* (or *TN*) the model predicts a given tweet is negative to CSI, but it is actually positive (or negative). The precision represents the ratio of which model is true (i.e., *TP*) to what is true (i.e., *TP* + *FP*). Equation (1) shows the equation for *precision*. *Recall*, which indicates sensitivity, represents the ratio of what the model is true (i.e., *TP*) to be true (i.e., *TP* + *FN*). Equation (2) shows the equation for *recall*. *F*1-*score* is a harmonic mean of *precision* and *recall*. It has been known that, even when the data is imbalanced, *F*1-*score* can accurately evaluate the performance of the model [60]. Equation (3) is the equation for obtaining *F*1-*score*.

$$Precision = \frac{TP}{TP + FP}. \tag{1}$$

$$Recall = TPR = \frac{TP}{TP + FN} \tag{2}$$

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

AUC is another commonly used criterion to evaluate the accuracy of the classification and is obtained by calculating the area of the receiver operating characteristic (ROC) curve as shown in Figure 5 [61]. ROC curve is represented by *TPR*, y-axis, and *FPR*, x-axis. *TPR* is the same as *recall* in Equation (2). *FPR* is the proportion of the results that are incorrectly predicted as positive among the negative as shown in Equation (4). AUC means how much model is capable of binary classification [62]; the higher the AUC, the higher the accuracy of the model.

$$FPR = \frac{FP}{FP + TN}. \tag{4}$$

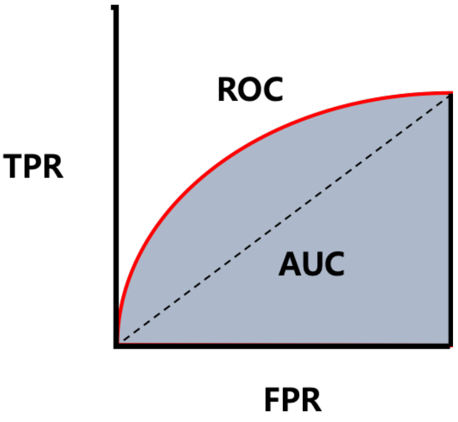


Figure 5. Area under the curve (AUC)-receiver operating characteristic (ROC) curve [61].

We performed all experiments on a machine with GeForce RTX 2080 Ti, Intel Core i7 7800X 3.50 GHz CPU, and 64GM RAM running Ubuntu 18.04. We used Python (version 3.7.3) for implementing the baseline and proposed models and Keras (version. 2.3.1) for CNN and LSTM.

6.2. Evaluation Result

6.2.1. The Accuracy of the Baseline Models

We measured the accuracy of the baseline model based on a single embedding model. Table 4 shows the evaluation result of the baseline model by varying the corpus (i.e., CSI-positive tweet data set, CSI-negative tweet data set, CVE data set, and Wikitext data set) and classifiers (i.e., CNN and LSTM).

Table 4. Evaluation result of the baseline models.

Embedding Model	Corpus	Measurement Learning Model	Precision	Recall	F1-Score	AUC
CSI-positive embedding	CSI-positive tweet data set	CNN	0.535	0.853	0.658	0.557
		LSTM	0.526	0.865	0.654	0.543
	CVE data set	CNN	0.958	0.856	0.904	0.909
		LSTM	0.952	0.869	0.909	0.912
CSI-negative embedding	CSI-negative tweet data set	CNN	0.890	0.860	0.875	0.874
		LSTM	0.887	0.870	0.878	0.879
	Wikitext data set	CNN	0.907	0.851	0.878	0.882
		LSTM	0.905	0.867	0.886	0.888

In CSI-positive embedding, when we used the CVE data set (i.e., background knowledge) as the corpus, it showed much better accuracy than the CSI-positive tweet data set. This result demonstrated the necessity of using the background knowledge for detecting CSI from tweets. We analyzed the result for each corpus in detail. (1) When the CSI-positive tweet data set is used as the corpus, it shows the lowest accuracy out of all the corpora: 0.654∼0.658 of F1-score and 0.543∼0.557 of AUC as the learning models are varied. We speculate that CSI-positive tweets contain much information related to CSI, at the same time, contain information with general expressions used in tweets. As a result, because CSI-positive tweets are relatively correctly classified (i.e., FN is low), the recall is high (i.e., 0.853∼0.865); however, because the ratio of actual positive tweets out of the tweets predicted as positive is low (i.e., FP is high), the precision is low (i.e., 0.526∼0.535). Table 5 shows four components (i.e., TP, FN, FP, and TN) used for calculating precision and recall to analyze the accuracy of the baseline models. Specifically, among 60,000 testing data sets, FN is only 4046∼4403, but FP is 22,204∼23,396. (2) When the CVE data set is used as the corpus, it shows the highest accuracy out of all the corpora: 0.904∼0.909 of F1-Score and 0.909∼0.912 of as the learning models are varied. In this case, as shown in Table 5, the recall is as high as the case of CSI-positive tweet data set; in addition, the precision is much higher than the case of CSI-positive tweet data set because FP becomes much lower than the case of CSI-positive tweet data set (i.e., only 1129∼1301).

In CSI-negative embedding, both the Wikitext data set and CSI-negative tweet data set show comparable accuracy in the classification. That is, in the case of CSI-negative tweet data set, F1-score was 0.875∼0.878 and AUC is 0.874∼0.879; in the case of Wikitext data set, F1-score was 0.878∼0.886 and AUC was 0.882∼0.888.

Table 5. Accuracy analysis of the baseline models.

Corpus	Measurement / Learning Model	TP	FN	FP	TN
CSI-positive tweet data set	CNN	25,597	4403	22,204	7796
	LSTM	25,954	4046	23,396	6604
CVE data set	CNN	25,674	4326	1129	28,871
	LSTM	26,056	3943	1301	28,699
CSI-negative tweet data set	CNN	25,799	4201	3177	26,823
	LSTM	26,098	3902	3330	26,670
Wikitext data set	CNN	25,528	4472	2623	27,377
	LSTM	25,987	4013	2729	27,271

6.2.2. The Accuracy of the Proposed Model

Because the CVE data set significantly outperformed the CSI-positive tweet data set, we chose the CVE data set as the only candidate for CSI-positive corpus in constructing the embedding model; however, because the CSI-negative tweet data set and Wikitext data set show comparable accuracy, we chose both of them for CSI-negative corpus. Table 6 shows the evaluation result of the proposed model by the possible fusion of CSI-positive and -negative corpus. We indicate that the fusion of the CVE data set and CSI-negative tweet data set shows better accuracy than that of the CVE data set and Wikitext data set.

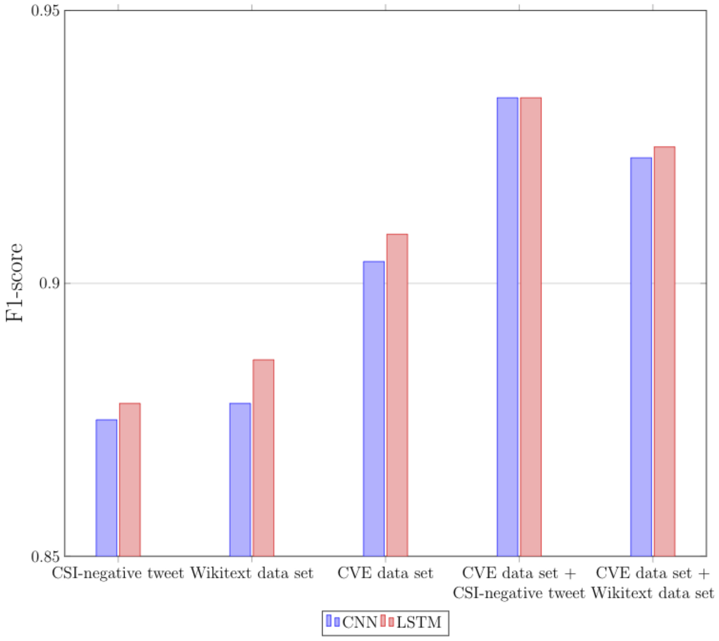
Table 6. Evaluation result of the proposed model.

Corpus	Measurement / Classifier	Precision	Recall	F1-Score	AUC
Fusion of CVE data set and CSI-negative tweet	CNN	0.957	0.912	0.934	0.935
	LSTM	0.955	0.911	0.932	0.934
Fusion of CVE data set and Wikitext data set	CNN	0.894	0.923	0.925	0.925
	LSTM	0.949	0.902	0.925	0.927

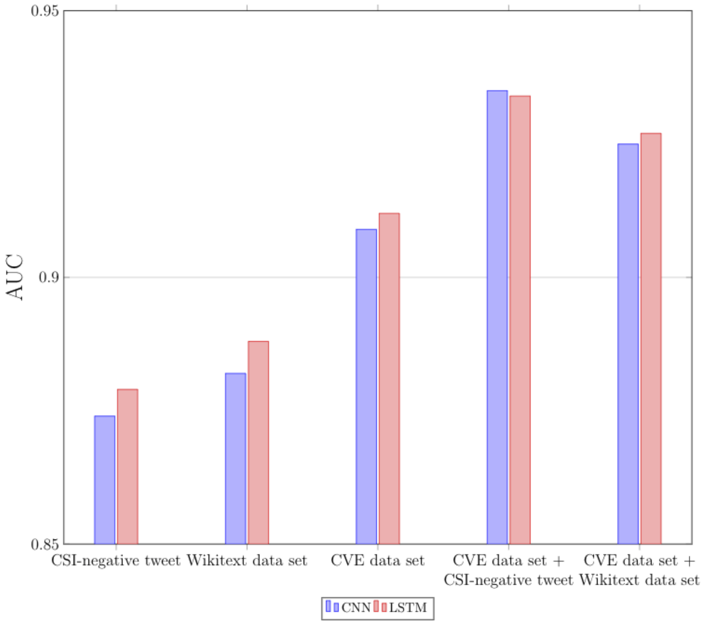
Figure 6a,b show the comparison of the accuracy between the baseline models and the proposed models using F1-Score and AUC, respectively. We indicate that the proposed models based on contrastive word embedding achieve meaningful improvement compared to the baseline models based on a single embedding model in both F1-score and AUC. Specifically, the proposed model based on the fusion of CVE data set and CSI-negative tweet data set improves the accuracy of the baseline model using only CVE data set by 2.53∼3.32% of F1-score and by 2.41∼2.86% of AUC; it improves using only CSI-negative tweet data set by 6.15∼6.74% of F1-score and 6.26∼7.00% of AUC. Table 7 shows four components (i.e., TP, FN, FP, and TN) used for calculating precision and recall to analyze the accuracy of the proposed model. By comparing Tables 5 and 7, we indicate that FP and TN in the proposed model are quite similar to them in the baseline model where only the CVE data set is used, which shows the best accuracy in the baseline model; however, in the proposed model, TP becomes increase and FN decrease (i.e., 1260∼1680), respectively, compared to the baseline model, improving the overall accuracy. In addition, the proposed model based on the fusion of CVE data set and Wikitext data set improves the accuracy of the baseline model using only CVE data set by 1.76∼2.10% of F1-score and 1.64∼1.76% of AUC; it improves that using only Wikitext data set by 4.40∼5.12% of F1-score and by 4.39∼4.99% of AUC. Here again, we indicate that FP and TN in the proposed model are quite similar to them in the baseline model where only the CVE data set is used; however, in the proposed model, TP increases and FN decreases (i.e., 1016∼1156), respectively, compared to the baseline model.

Table 7. Accuracy analysis of the proposed model.

Corpus	Measurement Classifier	TP	FN	FP	TN
Fusion of CVE data set and CSI-negative tweet	CNN	27,354	2646	1231	28,769
	LSTM	27,316	2684	1284	28,716
Fusion of CVE data set and Wikitext data set	CNN	26,830	3170	1305	28,695
	LSTM	27,072	2928	1453	28,547



(**a**)



(**b**)

Figure 6. Comparison of the baseline models and the proposed model. (**a**) F1-score. (**b**) AUC.

Here, we analyze the performance improvement of the proposed method compared to the best baseline model (i.e., using only CVE data set). As shown in Tables 4–7 the precision of the proposed model using the fusion of the CVE data set and CSI-negative tweet data set is similar to that of the baseline model using only the CVE data set. However, the proposed model significantly improves the recall by 4.83∼6.54% compared to the baseline model. As a result, the overall performance of the proposed method shows a meaningful improvement compared to the baseline model.

7. Discussion

The goal of the classification: when we work on the cyber threat intelligence, it is difficult to clearly determine the tweets related to cyber threats from public SNS media. In this paper, we propose a classification model for detecting tweets related to cybersecurity, which can be more clearly determined than tweets related to cyber threats by using the curated data sets without manual efforts (See Section 4). In this paper, to focus on showing the effectiveness of the proposed model, we define the corpus using the clearer criteria. If we can clearly define the corpus for cyber-threat-related tweets, we can also apply the proposed model to the corpus.

Fusion of learning models: to integrate the multiple learning models into a classifier, we have considered two approaches: (1) model fusion, which has been finally used in the proposed model, and (2) data fusion, which have been used in the existing studies for other problems: measuring the semantic relevance of short texts [63] and detection of the adverse drug reaction [64]. We have also considered data fusion for the model fusion. Figure 7 shows the architecture of the data fusion we have considered. That is, when we construct the embedding model, we integrate both CSI-positive and -negative corpus as one corpus. However, we have observed that the data fusion is not effective for classifying CSI-positive and -negative tweets as shown in Figure 8a,b where CVE data set and CSI-negative tweet data set are used as the corpora. That is, when two corpora are combined, the result model even degrades the classification accuracy based on a single embedding model.

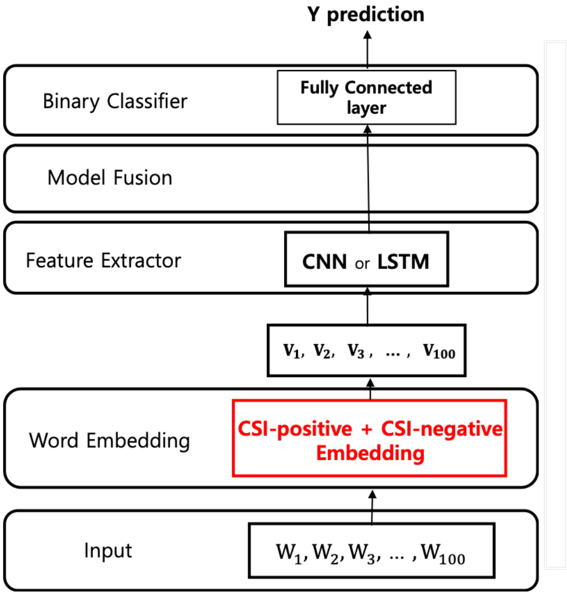
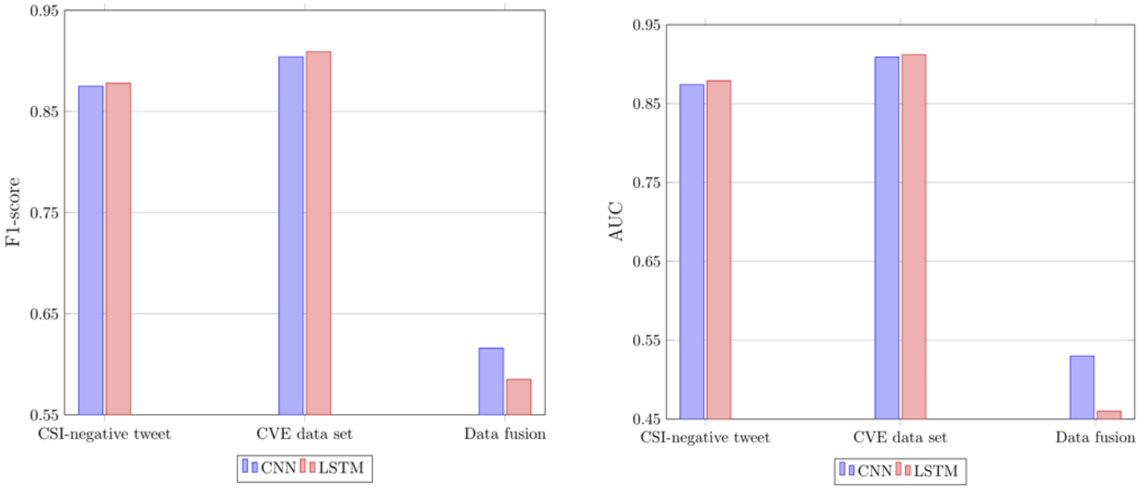


Figure 7. The data fusion model.



(**a**) (**b**)

Figure 8. Comparison between the baseline models and the data fusion. (**a**) F1-score. (**b**) AUC.

8. Conclusions

In this paper, we have devised a new text classification model based on deep learning to classify CSI-positive and -negative tweets from a collection of tweets. For this, we have proposed a novel word embedding model, called contrastive word embedding, that enables us to maximize the difference between base embedding models. First, we have defined CSI-positive and -negative corpora, which are used for constructing embedding models. Here, to supplement the imbalance of tweet data sets, we have additionally employed the background knowledge for each tweet corpus: (1) CVE data set for CSI-positive corpus and (2) Wikitext data set for CSI-negative corpus. Second, we have adopted the deep learning models such as CNN or LSTM to extract adequate feature vectors from the embedding models and integrate the feature vectors into one classifier.

To validate the effectiveness of the proposed embedding model, we have compared the proposed model with two baseline classification models: (1) a model based on a single embedding model constructed with CSI-positive corpus only and (2) another model with CSI-negative corpus only. In the experiment, we used 70,000 tweets for CSI-positive and CSI-negative corpora as the training data set, respectively, and 30,000 tweets for each corpus as testing data set, respectively. As a result, we have indicated that the proposed model shows high accuracy, i.e., 0.934 of F1-score and 0.935 of AUC, which improves the baseline models by 1.76∼6.74% of F1-score and by 1.64∼6.98% of AUC.

In this paper, we proposed the concept of contrastive word embedding, and it has been actually used for classifying the CSI-positive and -negative tweets. The concept of contrastive word embedding can be more widely used because it can be generalized by defining base multiple embedding models and by effectively integrating them. Hence, this is applicable for improving the classification performance of other domains by defining embedding models suitable for the problem and optimizing its combination.

Author Contributions: conceptualization, H.-S.S., H.-Y.K., and S.-J.R.; methodology, H.-S.S. and H.-Y.K.; validation, H.-S.S. and H.-Y.K.; data curation, H.-S.S. and S.-J.R.; writing—original draft preparation, H.-S.S.; writing—review and editing, H.-Y.K. and S.-J.R.; supervision, H.-Y.K.; project administration, H.-Y.K.; funding acquisition, H.-Y.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2018R1C1B5084424) and in part by NSR grant (2020-107).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CSI	Cyber Security Intelligence
OSINT	Open Source Intelligence
HUMINT	Human Intelligence
CVE	Common Vulnerabilities and Exposures
NVD	National Vulnerability Database
IMDB	Internet Movie Database
CNN	Convolutional Neural Network
TF-IDF	Term Frequency-Inverse Document Frequency
GCN	Graph Convolutional Network
NLP	Natural Language Processing
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
SVM	Support Vector Machine
MLP	Multi-Layer Perceptron
k-NN	*k* Nearest Neighbor
LDA	Latent Dirichlet Allocation
ReLu	Rectified Linear Unit
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
BiLSTM	Bidirectional Long Short Term Memory
DDoS	Distributed Denial of Service

References

1. Twitter by Numbers: Stats, Demographics & Fun Facts. Available online: https://www.omnicoreagency.com/twitter-statistics/ (accessed on 14 July 2020).
2. Han, B.; Cook, P.; Baldwin, T. Text-based twitter user geolocation prediction. *J. Artif. Intell. Res.* **2014**, *49*, 451–500. [CrossRef]
3. Conover, M.D.; Gonçalves, B.; Ratkiewicz, J.; Flammini, A.; Menczer, F. Predicting the political alignment of twitter users. In Proceedings of the 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing, Boston, MA, USA, 9–11 October 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 192–199.
4. Wang, X.; Gerber, M.S.; Brown, D.E. Automatic crime prediction using events extracted from twitter posts. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 231–238.
5. Ramage, D.; Dumais, S.; Liebling, D. Characterizing microblogs with topic models. In Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media, Washington, DC, USA, 23–26 May 2010.
6. What Is Cyber Security? Available online: https://www.kaspersky.com/resource-center/definitions/what-is-cyber-security (accessed on 31 August 2020).
7. Campiolo, R.; Santos, L.A.F.; Batista, D.M.; Gerosa, M.A. Evaluating the utilization of Twitter messages as a source of security alerts. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra, Portugal, 18–22 March 2013; pp. 942–943.
8. Sabottke, C.; Suciu, O.; Dumitraș, T. Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits. In Proceedings of the 24th USENIX Security Symposium (USENIX Security 15), Washington, DC, USA, 12–14 August 2015; pp. 1041–1056.
9. Arora, S.; Liang, Y.; Ma, T. A simple but tough-to-beat baseline for sentence embeddings. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
10. Ritter, A.; Wright, E.; Casey, W.; Mitchell, T. Weakly supervised extraction of computer security events from twitter. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 896–905.

11. Le, B.D.; Wang, G.; Nasim, M.; Babar, M.A. Gathering cyber threat intelligence from Twitter using novelty classification. In Proceedings of the 2019 International Conference on Cyberworlds (CW), Kyoto, Japan, 2–4 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 316–323.
12. Chambers, N.; Fry, B.; McMasters, J. Detecting denial-of-service attacks from social media text: Applying nlp to computer security. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; Volume 1, pp. 1626–1635.
13. Dionísio, N.; Alves, F.; Ferreira, P.M.; Bessani, A. Cyberthreat detection from twitter using deep neural networks. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–8.
14. Jindal, N.; Liu, B. Review spam detection. In Proceedings of the 16th International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 1189–1190.
15. Bijalwan, V.; Kumar, V.; Kumari, P.; Pascual, J. KNN based machine learning approach for text and document mining. *Int. J. Database Theory Appl.* **2014**, *7*, 61–70. [CrossRef]
16. Haddoud, M.; Mokhtari, A.; Lecroq, T.; Abdeddaïm, S. Combining supervised term-weighting metrics for SVM text classification with extended term representation. *Knowl. Inf. Syst.* **2016**, *49*, 909–931. [CrossRef]
17. Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.
18. Lai, S.; Liu, K.; He, S.; Zhao, J. How to generate a good word embedding. *IEEE Intell. Syst.* **2016**, *31*, 5–14. [CrossRef]
19. Joachims, T. *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization*; Technical Report; Carnegie-Mellon Univ Pittsburgh pa Dept of Computer Science: Pittsburgh, PA, USA, 1996.
20. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
21. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *2*, 3111–3119.
22. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
23. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
24. Li, L.; Qin, B.; Liu, T. Contradiction detection with contradiction-specific word embedding. *Algorithms* **2017**, *10*, 59. [CrossRef]
25. Liu, P.; Qiu, X.; Huang, X. Learning context-sensitive word embeddings with neural tensor skip-gram model. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
26. Bruni, E.; Boleda, G.; Baroni, M.; Tran, N.K. Distributional semantics in technicolor. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju, Korea, 8–14 July 2012; Volume 1, pp. 136–145.
27. Luo, Y.; Tang, J.; Yan, J.; Xu, C.; Chen, Z. *Pre-Trained Multi-View Word Embedding Using Two-Side Neural Network*; AAAI: Menlo Park, CA, USA, 2014; pp. 1982–1988.
28. Zhang, Y.; Roller, S.; Wallace, B. MGNC-CNN: A simple approach to exploiting multiple word embeddings for sentence classification. *arXiv* **2016**, arXiv:1603.00968.
29. Ren, F.; Deng, J. Background knowledge based multi-stream neural network for text classification. *Appl. Sci.* **2018**, *8*, 2472. [CrossRef]
30. Annane, A.; Bellahsene, Z.; Azouaou, F.; Jonquet, C. Building an effective and efficient background knowledge resource to enhance ontology matching. *J. Web Semant.* **2018**, *51*, 51–68. [CrossRef]
31. Li, C. Text Classification Based on Background Knowledge. Ph.D. Thesis, Department Advance Technology Science Information, Alcorn State University, Alcorn, MS, USA, 2017.
32. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]

33. Xu, H.; Dong, M.; Zhu, D.; Kotov, A.; Carcone, A.I.; Naar-King, S. Text classification with topic-based word embedding and convolutional neural networks. In Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, Seattle, WA, USA, 20–23 October 2016; pp. 88–97.
34. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.
35. Hu, Y.; Yi, Y.; Yang, T.; Pan, Q. Short text classification with a convolutional neural networks based method. In Proceedings of the 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 18–21 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1432–1435.
36. Liu, Y.; Ji, L.; Huang, R.; Ming, T.; Gao, C.; Zhang, J. An attention-gated convolutional neural network for sentence classification. *Intell. Data Anal.* **2019**, *23*, 1091–1107. [CrossRef]
37. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [CrossRef]
38. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
39. Wang, J.H.; Liu, T.W.; Luo, X.; Wang, L. An LSTM approach to short text sentiment classification with word embeddings. In Proceedings of the 30th Conference on Computational Linguistics and Speech Processing (ROCLING 2018), Hsinchu, Taiwan, 4–5 October 2018; pp. 214–223.
40. Ding, Z.; Xia, R.; Yu, J.; Li, X.; Yang, J. Densely connected bidirectional lstm with applications to sentence classification. In *CCF International Conference on Natural Language Processing and Chinese Computing*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 278–287.
41. Sriram, B.; Fuhry, D.; Demir, E.; Ferhatosmanoglu, H.; Demirbas, M. Short text classification in twitter to improve information filtering. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, 19–23 July 2010; pp. 841–842.
42. Alsmadi, I.; Hoon, G.K. Term weighting scheme for short-text classification: Twitter corpuses. *Neural Comput. Appl.* **2019**, *31*, 3819–3831. [CrossRef]
43. Zhou, C.; Sun, C.; Liu, Z.; Lau, F. A C-LSTM neural network for text classification. *arXiv* **2015**, arXiv:1511.08630.
44. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
45. Yao, L.; Mao, C.; Luo, Y. Graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence,Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7370–7377.
46. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2017; pp. 3856–3866.
47. Yang, M.; Zhao, W.; Ye, J.; Lei, Z.; Zhao, Z.; Zhang, S. Investigating capsule networks with dynamic routing for text classification. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 3110–3119.
48. Yang, L.; Li, C.; Ding, Q.; Li, L. Combining lexical and semantic features for short text classification. *Procedia Comput. Sci.* **2013**, *22*, 78–86. [CrossRef]
49. Qureshi, M.A.; Greene, D. EVE: Explainable vector based embedding technique using Wikipedia. *J. Intell. Inf. Syst.* **2019**, *53*, 137–165. [CrossRef]
50. Ren, F.; Li, C. Hybrid Chinese text classification approach using general knowledge from Baidu Baike. *IEEJ Trans. Electr. Electron. Eng.* **2016**, *11*, 488–498. [CrossRef]
51. Zong, S.; Ritter, A.; Mueller, G.; Wright, E. Analyzing the perceived severity of cybersecurity threats reported on social media. *arXiv* **2019**, arXiv:1902.10680.
52. Le Sceller, Q.; Karbab, E.B.; Debbabi, M.; Iqbal, F. Sonar: Automatic detection of cyber security events over the twitter stream. In Proceedings of the 12th International Conference on Availability, Reliability and Security, Reggio Calabria, Italy, 29 August–2 September 2017; pp. 1–11.
53. Petrović, S.; Osborne, M.; Lavrenko, V. Streaming first story detection with application to twitter. In Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, CA, USA, 2–4 June 2010; pp. 181–189.
54. Alves, F.; Bettini, A.; Ferreira, P.M.; Bessani, A. Processing tweets for cybersecurity threat awareness. *arXiv* **2019**, arXiv:1904.02072.
55. Dong, X.; Qian, L.; Huang, L. Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach. In Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju, Korea, 13–16 February 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 119–125.

56. Steele, R.D. Open source intelligence: What is it? why is it important to the military? *Am. Intell. J.* **1996**, *8*, 457–470.
57. Merity, S.; Xiong, C.; Bradbury, J.; Socher, R. Pointer sentinel mixture models. *arXiv* **2016**, arXiv:1609.07843.
58. Aphinyanaphongs, Y.; Lulejian, A.; Brown, D.P.; Bonneau, R.; Krebs, P. Text classification for automatic detection of e-cigarette use and use for smoking cessation from twitter: A feasibility pilot. *Pac. Symp. Biocomput.* **2016**, *21*, 480–491.
59. Baltrušaitis, T.; Ahuja, C.; Morency, L.P. Multimodal machine learning: A survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 423–443. [CrossRef]
60. Musicant, D.R.; Kumar, V.; Ozgur, A. Optimizing F-Measure with Support Vector Machines. In Proceedings of the FLAIRS Conference, St. Augustine, FL, USA, 12–14 May 2003; pp. 356–360.
61. Hanley, J.A.; McNeil, B.J. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* **1982**, *143*, 29–36. [CrossRef]
62. Narkhede, S. Understanding AUC-ROC Curve. Available online: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 (accessed on 4 August 2020).
63. El-Deeb, R.; Al-Zoghby, A.M.; Elmougy, S. Multi-corpus-based model for measuring the semantic relatedness in short texts (SRST). *Arab. J. Sci. Eng.* **2018**, *43*, 7933–7943. [CrossRef]
64. Sarker, A.; Gonzalez, G. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *J. Biomed. Inform.* **2015**, *53*, 196–207. [CrossRef] [PubMed]