EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

Narcotic-related tweet classification in Asia using sentence vector of word embedding with feature extension

Narongsak Chayangkoon* and Anongnart Srivihok

Department of Computer Science, Faculty of Science, Kasetsart University, Bangkok 10900, Thailand

Received 22 December 2020
Revised 14 February 2021
Accepted 17 February 2021

Abstract

Currently, Asia faces a narcotic drug addiction problem. In social networking services, such as Twitter, some drug addicted users converse about behaviours related to narcotic drugs. This research proposes a new Narcotic-related Tweet Classification Model (NTCM) that uses data preprocessing. Two new data preprocessing methods, Sentence Vector of Word Embedding (SVWE) and Sentence Vector of Word Embedding with Feature Extension (SWEF), are introduced to prepare data for the NTCM. The proposed data preprocessing method uses the reduction of the dataset to produce an SVWE. Word embedding is generated by deep neural networks using the skip-gram model. The authors further extended some features to SVWE to produce a new dataset called SWEF; these datasets were used for the dataset in the NTCM. The authors collected data with keywords related to narcotic drugs from Twitter in Asia. The authors investigated a text classification model using a Support Vector Machine, Logistic Regression, a Decision Tree, and a Convolutional Neural Network. Logistic Regression with the SWEF provided the best approach for the NTCM compared with state-of-the-art methods. The proposed NTCM showed correctness and fitness by accuracy (0.8964), F-Measure (0.895), AUC (0.949), Kappa (0.7131), MCC (0.714), and low running time performance (1.04 seconds).

Keywords: Data mining, Data preprocessing, Feature reduction, Narcotic drug, Text classification, Text vectorization, Word embedding

1. Introduction

The United Nations Office on Drugs and Crimes (UNODC) published a report on the epidemic situation of drugs and noted that East Asia and Southeast Asia faced epidemics of methamphetamine and amphetamine use. In particular, methamphetamine production is thought to be concentrated in Indonesia, while Afghanistan and the Golden Triangle had the highest global production of cannabis and opium. Moreover, a report from UNODC stated that approximately 243 million people in the world used narcotic drugs. Outbreaks of illicit drug use have spread across all regions of the world, and notably, the arrest rate for methamphetamine and amphetamine use in Asia grew by 400% during 1998-2014 [1].

Globally, society, economic development, and technology have evolved rapidly. New channels of communication help drugs spread faster and more widely. Addicts share drug activity through social networks, with Twitter being especially popular. However, very few studies have developed models for text classification that monitor narcotic drug-related messages on social media [2], and few studies have examined the situation in Asia. The rate of drug outbreaks is increasing annually [1]. The worsening situation resulted in the development of this research on the Narcotics-related Tweet Classification Model (NTCM), which used machine learning and text classification models to address the problem.

The development of the NTCM includes an important method, data preprocessing. The most popular method for data preprocessing is Bag-of-Words (BoW) [3]. At present, BoW is the well-known text mining method that takes each word's occurrence into account as a feature in the classification; it is called a one-hot-encoded vector. A sparse vector containing each word's index and frequency is used to represent it [4]. Some datasets may have potential features that have many vectors. BoW has a scalability challenge as a high-dimensional vector [5]. Previous research introduced the New Document-Term Matrix Data (NDTMD) method, which reduced the dimensional vector of BoW [6]. However, NDTMD still has a large feature size. In this research, the authors develop a new data preprocessing method adapted from NDTMD with even fewer features.

The main objective of the research was to establish the NTCM, which relies on text classification knowledge. To do this, the authors propose a new data preprocessing method for dataset feature reduction using word embedding that calculates the average of the vector of word embedding. This method produces a new dataset called the Sentence Vector of Word Embedding (SVWE). Word embedding is generated from the corpus using the skip-gram model. This model is created using deep neural networks, which are a version of Neural Language Processing (NLP) for efficient training of word embedding [7]. The authors then extended two features into SVWE to produce a dataset called the Sentence Vector of Word Embedding with Feature Extension (SWEF). In this research, the authors gathered messages from Asian Twitter users. The messages were classified into two categories: abuse and non-abuse. According to the authors, the dataset was divided into two parts: a training set and a test set. The four classifiers used in the research experiment were

*Corresponding author. Tel.: +668 9199 0395
Email address: narongsak.chay@ku.th
doi: 10.14456/easr.2021.57

Support Vector Machine (SVM), Logistic Regression (LR), decision tree (J48), and Convolutional Neural Network (CNN). Individual model output was evaluated using accuracy, F-measure, Area under the ROC Curve (AUC), Kappa, Matthews Correlation Coefficient (MCC), and running time performance. The authors ultimately evaluated the text classification models using multiple data preprocessing methods and summarized the best approach for the NTCM. The LR algorithm gave the highest performance measurements with SWEF. This model was the best approach for the NTCM.

The contribution of this research is the proposal of new data preprocessing methods for the development of the NTCM. These methods are called SVWE and SWEF and are optimized by word embedding. These methods reduce the number of features of the dataset and produce new datasets smaller than BoW, Term Frequency-Inverse Document Frequency (TF-IDF), and NDTMD. Performance measurements of SVWE and SWEF are higher than Latent Semantic Analysis (LSA), Paragraph Vector-Distributed Memory model (PV-DM), and Paragraph Vector without Word Ordering-Distributed Bag of Words (PV-DBOW). Furthermore, NTCM might develop a prototyping tool for detecting messages about the abuse of narcotic drugs on Twitter in Asia.

2. Literature review

In today's world, drug addicts use new channels to make their way into communities. Addicts usually share drug-related activity on social media such as Twitter. Researchers have attempted to find new text classification models for identifying messages about narcotic drug use that spread on Twitter. Few text classification models address the narcotic drug problem on Twitter [2]. Further, there is scant research done on the Asia narcotic dataset. The NTCM is one form of text classification using a text mining function. Text classification assigns target categories to objects in a collection of data. In each case of data, it can be used the text classification model to predict the target category correctly. Previous researchers used various classifiers for developing NTCMs. One study using an NTCM was developed by Phan et al. to detect the distribution of narcotic drug messages on Twitter. Marijuana, cocaine, and heroin were some of the most often discussed drugs on Twitter. The Twitter data stream was accessed using an API. Based on the keywords, the authors filtered and stored narcotic-drug messages in the database. Their experts labelled tweets related to narcotic drugs that were classified into two categories: abuse and non-abuse. They experimented with models and compared their performance using three classification algorithms (SVM, J48, and Naïve Bayes). The performance measurements included recall, precision, and F-measure. According to the experimental findings, the J48 classifier with TF-IDF showed the highest F-measure (74.8%). Phan et al.'s suggestion was to use TF-IDF to improve the accuracy of the classifiers. They found that interesting aspects of their work were streaming Twitter and classified tweets into abuse and non-abuse categories [2]. Both methods have been adapted for use in this article.

The authors of the current work studied text classification using messages from Twitter and techniques for data preprocessing and classification algorithms that previous researchers used in their research. Dhariyal et al. [8] proposed a model for sentiment analysis using Doc2Vec and CNN hybrids. Their proposed approach was named the Convolutional Neural Network-Probabilistic Neural Network (CNN-PNN). The datasets were movie reviews from the Stanford Network Analysis Platform (SNAP) website. Text vectorization used two algorithms, PV-DM and PV-DBOW. They applied the CNN-PNN to develop the text classification model. This model achieved an AUC of 96.63%, which is high [8]. Chen et al. [9] studied the automated detection of abusive content on social media. They compared the performance of the three classifiers: SVM, CNN, and RNN. They focused on the performance of the classifiers with an imbalanced dataset. The CNN classifier outperformed the SVM classifier. In addition, SVM classifiers with an average vector of word embedding achieved high performance with balanced datasets. SVM working with TF-IDF has particularly been a popular text classification model [9]. Ahmad et al. [10] focused their research on tuning SVM performance for sentiment analysis. The datasets used posts from micro-blogging sites. The set of features was defined using TF-IDF, and they used SVM to classify the sentiment analysis. The results showed that SVM worked best with TF-IDF features and provided the highest detection of positive and negative posts. This model achieved the highest recall, precision, and F-measure [10]. Burel and Alani [11] proposed a Crisis Event Extraction Service (CREES) that was an application for automatically identifying relevant posts. That model identified social media posts about hurricanes or floods, and the researchers experimented with TF-IDF to create text vectorization using the SVM algorithm. Another experiment used CNN, which was trained using word embedding. The findings revealed that CNN with pretrained word embedding did not outperform SVM substantially [11]. Likewise, Rameshbhai and Paulose [12] proposed opinion mining on newspaper headlines using linear SVM. They investigated text classification with an online website dataset (http://www.indianexpress.com). Their results showed that linear SVM using TF-IDF provided higher accuracy than Stochastic Gradient Descent (SGD) with TF-IDF [12]. Finally, Pimpalkar and Raj [13] compared data preprocessing methods for a Twitter content dataset. They used tweet datasets from the Internet Movie Database (IMDb). This research compared the performance of text vectorization features using BoW and TF-IDF. The classifiers chosen and compared were SVM, Multinomial Naïve Bayes (MNB), LR, decision tree, and eXtreme Gradient Boosting (XGBoost). Their results showed that SVM and BoW provided the highest accuracy with the Twitter content [13].

2.1 Word embedding

Word embedding is a set of word representations using number vectors produced using the Word2Vec program. Text vectorization is represented using unique numerical vectors calculated using the probability of words and context words in the sentence. The set of documents is the input data, and the outputs are vectors and several dimensions. A vector corresponds to each distinct term in the corpus. The context similarity is evaluated according to the corpus positions of the vectors. Word2Vec software has two architectures to produce word distribution and representation consisting of the Continuous Bag-of-Words (CBOW) model and the continuous skip-gram model. The skip-gram model does a better job for infrequent words by using the current word to analyse the window size of context words [14]. The skip-gram model is a deep learning approach used for NLP that optimizes deep neural networks and emphasizes training efficiency. This version uses the estimation method with a neural network called linear neurons. This model aims to try and predict the context words when the input is a given target word. The model consists of three layers: 1) The input layer is the conversion of input to a one-hot vector. The size of one input unit is equal to one word in the context window. Input units are distributed to all units in the hidden layer by feed-forward to calculate the word's probability in a context window. 2) The hidden layer uses the linear neurons classifier, where each hidden neuron has a weight value. Input into the hidden neuron is compressed to a smaller dimension: the size of the hidden neuron, such as 100, 300, or 600. 3) The output layer is equal to the number of words from the input. The output layer receives a value of the hidden neuron. Every unit of output has a weight value obtained from the hidden neuron of every

relationship. The output is a neuron weight vector that is multiplied against the word vector. The vectors are the text vectorization of words [15].

2.2 Data preprocessing

2.2.1 Text vectorization

BoW is a text representation model. This vector space model is a simple traditional data processing method for text mining. Text vectorization represents the occurrence of each word with regard to its position. The words are features that are used to train the classifier. Each feature is represented by a numeric value. Therefore, a document is represented by a vector of its words' values [4]. Various short text classification models represent document vectors using TF-IDF. Those models could be fitted with SVM and provided high-performance measurements [2, 10-12]. Those studies showed that TF-IDF is a popular data preparation method for text mining. The TF-IDF uses terms for features that are used for training the classifier. TF is the weighted numbers of frequency measurement that evaluates the importance of terms in the document. IDF generates the inversion weight of each feature in the corpus. TF-IDF defines the weight of the term (*t*) in the document (*d*) based on Pimpalkar and Raj [16]. TF-IDF is defined as follows:

$$TF\text{ - }IDFt, d = (TFt, d \times IDFt) \quad (1)$$

LSA is a theory and mathematical method for creating text vectorization and extracting relationships between words and documents by analyzing the context using the meaning of words. This concept assumes that closely related terms often occur in similar documents. The statistical calculation of LSA uses Singular Value Decomposition (SVD), which is a technique of feature reduction that maintains the identity of either document as much as possible [16]. The SVD is defined as follows:

$$C = U\sum V^{T} \quad (2)$$

SVD helps decompose the term-document matrix *C* into a term-concept matrix *U* and a concept document matrix *V*. $\sum$ is a singular value matrix on diagonals [16].

PV-DM uses other context words in the paragraph to predict words in an ordered sequence. This version matches words with vectors and matches each paragraph with a unique vector to display the vector in the matrix column. PV-DM has two essential processes: 1) the context provides paragraph identification, so that paragraph identification is inserted as another word in an ordered sequence of words, and 2) the input word vectors are averaged or concatenated as part of the classifier process. Text vectorization is represented using the vectors of the paragraph [17]. Later, PV-DBOW solves the problem from the opposite direction used by PV-DM. Paragraph identification predicts words in a small window that does not have any word order restrictions. This version produces paragraph identification that predicts the target in a small window of the paragraph. The vectors are the text vectorization of the paragraph [17].

2.2.2 Text vectorization using word embedding

Chen et al. used word embedding to calculate the Average Vector of Word Embedding (AVWE) representing the dataset's document attribute, where each AVWE represents one sentence in the document. They used SVM and AVWE to develop models to detect abusive content on social media [9]. Another approach is the NDTMD, which is created using the feature reduction method for BoW in short text classification. This technique uses the intersection of words in word embedding and features of BoW. The features of the NDTMD dataset include infrequent words because word embedding is generated using the skip-gram model. This feature set represents documents similar to BoW. An advantage of this dataset is a smaller number of features than BoW. The method's performance is evaluated with five open datasets from Kaggle, the data science community website. The SVM model with this dataset had good performance evaluators for classifying short messages from social media such as Twitter, US Airline Sentiment, Ironic Corpus, and Deepnlp [6].

2.3 Classification algorithms

SVM is a supervised machine-learning algorithm that solves problems involving two categories. In the hyperplane, SVM determines a decision boundary that differentiates from the two categories by using a binary linear function and the maximum margin between the support vectors will be reserved [18]. In addition, this classifier has been used extensively to generate text classification models [9-13]. LR is a statistical model of probability estimation for dichotomous variables. The dependent variable is transformed into a logit variable using LR that uses maximum likelihood estimation. A logistic response function calculates the log-odd probability using independent variables. This model can be used with two categories using the variable as one of the two possible categories [19]. J48 is a subclass of a decision tree, which is a classification algorithm. A top-down greedy search that extracts features from the root node is used to build a decision tree. The feature with the highest discriminate value is created as the root node calculated from information gain and entropy [20]. Phan et al. [2] used J48 to develop narcotic-related tweet classification. CNN is a neural network with a high number of hidden layers (typically four). Moreover, convolutional layers emulate the response of accepting fields for a specific feature. For example, a CNN can emulate part of the human skin or part of an animal's body. Pooling layers decrease the data scale by combining the output of convolution layers. Fully connected layers connect all the neurons from the previous layer to every neuron in the next layer. The loss layer indicates the training, which is the performance of predicting the actual labels. Various loss functions are suitable for different tasks. For example, softmax is used to predict the non-mutually exclusive class [21].

Several related works were studied in Subchapters 2.1, 2.2, and 2.3. These studies, reviewed above, adapted traditional classifiers for text classification using social media data, such as SVM, LR, and J48. SVM has been widely used to develop text classification models because SVM is a strong classifier for a binary class dataset. Deep learning algorithms, such as CNN, are new recent classifiers used for text mining. PV-DM, PV-DBOW, and word embedding are the trending algorithms used to prepare data for text mining. BoW and TF-IDF are also still widely used for text vectorization. However, BoW and TF-IDF methods have some limitations. Neither

method can analyse the words with even a slight change in message. For example, both methods treat “king” and “emperor” as two different independent words. This effect makes both methods provide a high-dimensional vector dataset. Therefore, this research invents new data preprocessing methods (SVWE and SWEF) that eliminate the BoW and TF-IDF method weaknesses.

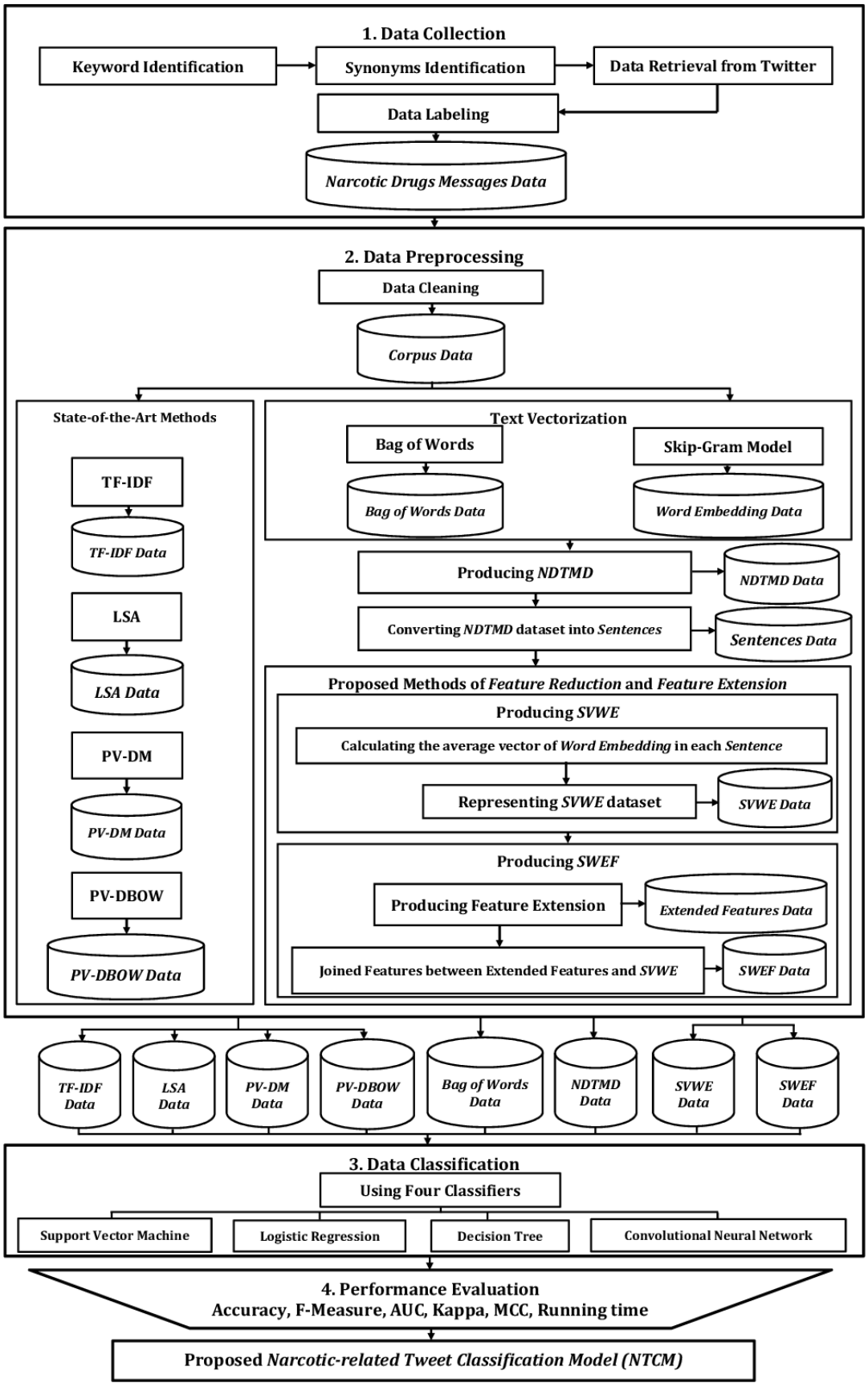

Figure 1 Overview of the research framework

3. Methodology

The authors developed a narcotic drug-related message classification model in this research for short messages on Twitter in Asia. This model is called the NTCM. Furthermore, the authors developed a new data preprocessing method that involved a feature reduction of the dataset created from word embedding. This method calculates the average vector of word embedding and produces SVWE, a new dataset. SVWE was extended by additional features to produce another dataset named SWEF. The research methodology used to develop the NTCM consisted of four steps: 1) data collection, 2) data preprocessing, 3) data classification, and 4) performance evaluation. The research framework is shown in Figure 1.

3.1 Data collection

Data collection focused on narcotic drug-related messages from Twitter and, specifically, messages delivered in Asia from February to June 2019. The data collection process included keyword identification, synonym identification, collecting data from Twitter using identified keywords, and data labelling by experts.

3.1.1 Keywords identification

Keyword identification finds keywords related to narcotic drug names, consisting of general names, drug slang names, drug street names, and drug combination keywords. The authors collected these keywords from reliable significant sources, including the National Center on Addiction and Substance Abuse (NCASA), American Addiction Centers, and The Telegraph. Keywords are as follows: 1) General names: The common names used for narcotic drugs, including marijuana, amphetamine, methamphetamine, MDMA, ketamine, opium, heroin, cocaine, Rohypnol, hydroxybutyrate, Salvia divinorum, LSD, and inhalants [22]. 2) Drug slang names: The 3,000 drug slang names were published in The Telegraph. The polices of the United Kingdom identified and gave these drugs slang keywords [23]. 3) Drug street names: Drug street names collected and published by American Addiction Centers [24]. 4) Drug combination keywords: Drug combinations were collected and published by the National Center on Addiction and Substance Abuse [22].

3.1.2 Synonym identification

Synonym identification is the process of identifying words that are similar to narcotic drug names. There are three steps. 1) Defining query keywords: Common narcotic drug names use query keywords. 2) Finding synonyms: The GoogleNews-Vectors-Abuse300 Model was used to find synonyms by calculating cosine similarity. This model includes three million words collected from Google News Corpus [25]. 3) Word filtering: The authors selected words with a similarity value higher than 0.55 to a narcotic drug name as measured using cosine similarity. The authors did not select words that had the same slang name and street name.

3.1.3 Data retrieval

Twitter provides an API program that allows users to access Twitter messages. Streaming connections can display data before all the files are packaged into the computer. There are five steps. 1) Register Twitter API key: The Twitter API requires API keys, consisting of an access token, access token secret, consumer key, and consumer secret, to gain access. 2) Limiting collection scope: This step identifies messages associated with narcotic drug use data in Asia. 3) Accessing Twitter public streaming API: The R programming language [26] collects and filters messages by keywords. 4) Collecting messages using keywords: Drug-related messages are collected by keywords from general names. 5) Storing messages: The data collection period spanned five months, from February to June 2019, and included 33,334 messages.

3.1.4 Data labelling

Experts from the Royal Thai Police were contacted to label the dataset. The narcotic drug message data were labelled either non-abuse or abuse. 1) Non-abuse messages are related to narcotic drugs, but no use is involved, such as a message referring to the drug's effect. 2) Abuse messages include messages supporting narcotic drugs, messages inviting users to use drugs, messages mentioning the use of narcotic drugs, and buying or selling illegal drugs. Examples are shown in Table 1.

Table 1 Examples of twitter messages

Categories	Messages
Non-Abuse	Bangladesh security forces have seized nearly 9 million methamphetamine pills in less than three months as a massive.
Non-Abuse	Half the Thai army high? Soldiers seize almost 8m speed pills and 50 kg of crystal methamphetamine in Chiang Rai.
Abuse	I do make people smoke pot or meth so coke put pills or tabs in their mouths or shoot themselves up.
Abuse	I liked a video of people smoking crystal meth for the first time.
Abuse	Have you ever smoke weed? Yes, to clarify marijuana is a plant and not a drug.

3.1.5 Final dataset

The dataset in this study was a social media data collection dataset. It was collected from tweets related to narcotic drug use on Twitter in Asia. The 33,334 messages consisted of 223,165 words. The completed 4,200 messages were selected. These messages involved the three-drug names amphetamine, methamphetamine, and marijuana, which are the major epidemic drugs in Asia. Finally, experts from the Royal Thai Police labelled this data collection. The characteristics of a final dataset are shown in Table 2.

Table 2 Characteristics of narcotic drug messages in the Twitter dataset

Dataset	Messages	Number of categories	Category members	Average sentence length (characters)	Features (words)
Narcotic drugs messages data in Asia	4,200	2	963 Abuse 3,237 Non-Abuse	94	42,429

Definition 1: Messages are *n* narcotic drug tweets in the dataset. The *tweet* is each tweet in the dataset. *Messages* = {$tweet_1$, $tweet_2$, $tweet_3$, …, $tweet_n$}.

3.2 Data preprocessing

Data preprocessing is a set technique for preparing data from the dataset of narcotic drug messages. This is the second step in NTCM development. Data preprocessing consists of four stages: 1) data cleaning, 2) text vectorization, 3) feature reduction, and 4) feature extension.

3.2.1 Data cleaning

Narcotic drug tweet data (*Messages*) are cleaned by transforming all words to lowercase and then removing punctuation, numbers, tabs, stop words, blank spaces at the beginning, and stemming words to a common base form. Data cleaning produced the *Corpus*.

Definition 2: The *Corpus* is the input dataset for state-of-the-art methods and text vectorization processes. The *doc* is each document in the dataset. *Corpus* = {doc_1, doc_2, doc_3, …, doc_n}.

3.2.2 Text vectorization

Text vectorization is the conversion of a text document into numerical form. The represented vector describes the text document using numerical features. The features depend on the text vectorization algorithm.

BoW is the most widely used data preprocessing for text mining because of its advantage in explaining the document's content because BoW has a large amount of training data [5]. In this research, BoW was created using a vectorizer function in the R programming package [26].

Definition 3: *Bag of Words* is the set of words in the *bow*. Then, the *bow* is each bag of words that are created from *Corpus. Bag of Words* = {bow_1, bow_2, bow_3, …, bow_n}.

The skip-gram model produces word embedding using the Word2Vec function [14, 25] with the R Programming package [26]. This model generates word embedding represented by 100 features.

Definition 4: *Word embedding* is the set of words and vector so that $\vec{V}$ is the vector of words in word embedding. The w_j is each word in word embedding. The *j* is the index of each word embedding. The *e* is the number of frequent words (words that appear more than five times in *Corpus*). *Word Embedding* = {w_1, w_2 , w_3 ,…, w_e}. The *v* is each feature of w_j, and each word w_j in word embedding has 100 features *v*. $\vec{V}(w_j)$ is the vector $\vec{V}$ of word w_j, and $\vec{V}(w_j)$ = {$v_{j,1}$, $v_{j,2}$, $v_{j,3}$, …, $v_{j,100}$}.

3.2.3 Feature reduction

Feature reduction reduces the number of features by creating a new vector with a smaller number of features. The proposed methods were SVWE and SWEF. The SVWE performs this using the average vector of the word embedding. It is based on NDTMD. In previous research, NDTMD reduced the dimensions of BoW using the intersection of features in BoW and words in word embedding (word selections). Therefore, the features in BoW are reduced by keeping only the features in word selections. This is done by calculating the sum of all word frequencies in each row of BoW, and then, if the sum is zero, that row is removed. The previous method generated a dataset called NDTMD, a text vectorization similar to BoW, but the number of features and instances is smaller [6]. NDTMD is defined as follows:

Definition 5: *NDTMD* is the set of bow'_i and *m* is the number of remaining instances in the *bag of words*. The w_j is a set of words in each bow'_i in *NDTMD*. The *t* is the number of words in each bow'_i. Then, bow'_i is each bag of words in *NDTMD*, where *i* is the index of each bow'_i. Then, bow'_i = {w_1, w_2, w_3, …, w_t}. *NDTMD* = {*bow'1, bow'2, bow'3, …, bow'*$_m$}.

However, the NDTMD dataset still included high-dimensional features. Therefore, this research proposed SVWE methods that produced a smaller dataset than NDTMD. The represented features of SVWE are created using the 100 features of word embedding. The text vectorization of the dataset is computed from the average vector of the word embedding, and the new dataset is called SVWE, which is represented by 100 features. Producing SVWE involves three steps.

Definition 6: *Sentences* are the set of arrays simplified from the bag of words bow'_i in *NDTMD*. S_i is each array of words, and *i* is the index of each s_i. Then, *Sentences* = {s_1, s_2, s_3, …, s_m}.

Definition 7: s_i is the array of words transformed from each bag of words bow'_i.
The w_j is each word in s_i. The *t* is the number of words in s_i. Then, s_i = {w_1, w_2 , w_3 ,…, w_t}.

Definition 8: $AVWE_i$ is the average vector $\vec{V}$ of word w_j, where w_j is in s_i and *word embedding*. *Avgv* is the average feature *v* at index 1 to 100 of each vector $\vec{V}$. $AVWE_i$ = {$Avgv_{i,1}$, $Avgv_{i,2}$, $Avgv_{i,3}$,…, $Avgv_{i,100}$}.

Step 1: The features of *NDTMD* are converted into a set of arrays and stored as *sentence* variables.

Step 2: If word w_j appears in each s_i, then $\vec{V}(w_j)$ is calculated as the average vector $\vec{V}$of word w_j in *word embedding*. The variable is stored as *AVWE*. The *AVWE* is defined as follows:

$$AVWE = \frac{1}{t}\sum \forall w_j \epsilon s_i \vec{V}(w_j) \quad (3)$$

where w_j is a word in s_i and *Word Embedding*, *v* is each feature of w_j, and each word w_j in *a word embedding*. $\vec{V}(w_j)$ is vector $\vec{V}$ of word $w_j = \{v_{j,1}, v_{j,2}, v_{j,3}, \ldots, v_{j,100}\}$.

Step 3: The representation of SVWE is a dataset consisting of $AVWE_1$ to $AVWE_m$. SVWE represents the features with the average feature *v* at index 1 to 100 and a dimension equal to 100.

Definition 9: *SVWE* is a set of $AVWE_1$ to $AVWE_m$. $SVWE = \{AVWE_1, AVWE_2, AVWE_3..., AVWE_m\}$. The algorithm of SVWE is shown in Figure 2.

SVWE 's Algorithm

Input: $NDTMD = \{bow'_1, bow'_2, bow'_3, ..., bow'_m\}$
Output: $SVWE$
$Sentences = \{\}$
For each bag of words bow'_i in $NDTMD$
  s_i = Array of words transformed from each bag of words bow'_i
  $Sentences = Sentences \cup \{s_i\}$
End for
$SVWE = \{\}$
For each array s_i in $Sentences$
  For word w_j in *Word Embedding*
    If w_j appears in each s_i
      $AVWE_i$ = Average vector $\vec{V}$ of word w_j in *Word Embedding*
      $SVWE = SVWE \cup \{AVWE_i\}$
    End if
  End for
End for

Figure 2 Algorithm 1: SVWE algorithm

3.2.4 Feature extension

Feature extension is the addition of relevant features that can benefit a short text classification model's performance. The extended features are used to train the model so that those features can influence the model's performance measurements. The feature extension adds significant features to SVWE. They consist of numerical digits in each tweet and the number of words in each sentence in the *Sentences* set. The combination of SVWE and both features is called SWEF, the second proposed method in this article.

Here, f_{nd} is the list of numerical digits generated from all tweets in the narcotic-drug tweet data (*Messages*). This is calculated from the original data before the data cleaning process. This feature increases classification efficiency because news reports of arrests related to narcotic drugs often mention the numbers of drug doses. For example, "***Bangladesh security forces have seized nearly 9 million methamphetamine pills in less than three months…***" is a non-abuse tweet that contains numerical digits. The f_{nd} is defined as follows:

Definition 10: f_{nd} is the list of nd_i (the number of numerical digits), where nd_i is calculated from each $tweet_i$ in *Messages* that is not removed and is still in *SVWE*. nd_i is each number of numeric digits in each $tweet_i$. $f_{nd} = \{nd_1, nd_2, nd_3, \ldots, nd_m\}$.

f_{nw} is the list of numbers of words in s_i calculated by counting words in each s_i from the *Sentences* set. This feature increases the classification efficiency because drug-related messages are short. Most of the long messages include news on drugs featuring arrest, articles on amphetamines, and research on the benefits of marijuana. The f_{nw} is defined as follows:

Definition 11: f_{nw} is the list of nw_i (numbers of words) calculated from each s_i from the *Sentences* set. nw_i is each number of words in each s_i. $f_{nw} = \{nw_1, nw_2, nw_3, \ldots, nw_m\}$.

SWEF is a dataset that represents the outer joined features between *SVWE* and additional features f_{nd} and f_{nw}.

Definition 12: *SWEF* is generated using the outer join function between *SVWE* and additional features f_{nd} and f_{nw}. Finally, *SWEF* is a combination of the 100 feature vectors of *SVWE* and two features from extended features. The *SWEF* feature set has 102 features, and the number of instances is *m*. The SWEF algorithm is shown in Figure 3.

SWEF 's Algorithm

Input: $SVWE = \{AVWE_1, AVWE_2, AVWE_3, ..., AVWE_m\}$
  $Messages = \{tweet_1, tweet_2, tweet_3, ..., tweet_n\}$
  $Sentences = \{s_1, s_2, s_3, ..., s_m\}$
Output: $SWEF$
$f_{nd} = \{\}$
For each tweet $tweet_i$ in *Messages*
  nd_i = Each number of numeric digits in each $tweet_i$
  $f_{nd} = f_{nd} \cup \{nd_i\}$
End for
$f_{nw} = \{\}$
For each array s_i in $Sentences$
  nw_i = Each number of words in each s_i
  $f_{nw} = f_{nw} \cup \{nw_i\}$
End for
$SWEF = SVWE$ join with f_{nd}
$SWEF = SWEF$ join with f_{nw}

Figure 3 Algorithm 2: SWEF algorithm

3.2.5 State-of-the-art methods

Four state-of-the-art methods for text vectorization (TF-IDF, LSA, PV-DM, and PV-DBOW) were used to convert the Corpus to text vectorization. The represented vector numbers were prepared as data for the text classification model.

3.3 Data classification

In this study, split tests divided the datasets into two subsets (training set and test set). The training set was defined as 80%, and the test set was defined as 20% of the dataset. This technique has a low running time [27]. Consequently, the four classifiers, SVM, LR, J48, and CNN, were used to develop a narcotic drug prediction model as follows: SVM solves the two categories problem and works well with unstructured data such as text or a document. SVM has the advantage of less chance of overfitting [18]. LR is a popular algorithm for binary classification problems. LR has advantages, as the feature value is used to calculate the log-odd probability, and it does not require a normal distribution for the input data [19]. For certain large datasets, J48 is a fitting decision tree algorithm. This version requires a low running time to develop a model [20]. CNN has advantages because it does not require an excessive feature selection process [21]. The classification models are implemented using the WEKA program. This freeware program is commonly used in data mining research [28].

3.4 Performance evaluation

Evaluation of feature reduction uses the feature reduction rate (FRR), which is a performance measurement of the feature reduction method. The reduction selects the most critical features to lower the number of features [29]. The equation of *FRR* is defined as follows:

$$FRR = \frac{(OF-FS)}{OF} \tag{4}$$

where *OF* represents the number of traditional features, and *FS* represents the number of features remaining after using the reduction method. When the *FRR* is close to one, the reduction rate of features is highly effective.

The evaluation of the classification performance uses six performance evaluators: accuracy, F-measure, AUC, Kappa, MCC, and running time.

Accuracy is a measurement of a classification model's correction that considers the expected correct measure from a classification model. The values in a confusion matrix are used to measure accuracy that itself is generated in classification modelling. An accuracy value closer to one means that the classification model's accuracy has a high percentage of correctness [30]. The equation of accuracy is defined as follows:

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{5}$$

The number of correctly and incorrectly accepted cases is known as *TP* (True Positive) and FP (False Positive), respectably. The number of correctly denied cases is called *TN* (True Negative) and the number of incorrectly denied cases is called *FN* (False Negative) [31].

The F-measure is a measurement considering the classification model performance based on recall and precision. The F-measure value is determined as the harmonic mean of recall and precision. The performance of the F-measure is calculated using both recall and precision. An F-measure value closer to one means that the classification model has near-perfect precision and recall [32]. The equation of F-measure is defined as follows:

$$Precision = \frac{TP}{TP+FP}, Recall = \frac{TP}{TP+FN}$$

$$F-measure = \frac{2 \times Precision \times Recall}{Precision \times Recall} \tag{6}$$

The AUC is a measurement of the entire two-dimensional area below the receiver operating characteristic curve (ROC) curve. An AUC value closer to one indicates a high rate of correctly accepted instances [33]. The equation for AUC is defined as follows:

$$AUC = \frac{1+TP-FP}{2} \tag{7}$$

Kappa is a nonparametric statistic used to assess the accuracy of two groups' classification (ground truth values and predicted values). The Kappa coefficient does not require the dataset of interest to have a normal distribution. A high Kappa coefficient indicates a high degree of consistent classification, whereas a low value indicates a low degree of consistent classification [34]. The Kappa equation is defined as follows:

$$Kappa = \frac{P_{observe}-P_{change}}{1-P_{change}} \tag{8}$$

where $P_{observe}$ is the agreement observed among assessors and P_{change} is the probability hypothesis of opportunity agreement.

In machine learning, MCC is a performance measurement of the quality of binary class classification. The MCC is considered to balance measurements that can be used even with an extremely imbalanced dataset. The MCC is a correlation coefficient between binary classes. The MCC provides a number between −1 and +1. TP, TN, FP, and FN can be used to measure MCC [35]. The equation for MCC is defined as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{9}$$

Running time is a classification model performance measurement. The proposed model was measured and compared to various classification models. The running time is composed of three indicators: 1) *Train Time* is time spent on training the training set, 2) *Test Time* is the time spent testing the test set, and 3) *Model Time* is the time spent with the NTCM choosing the predicting class. Specifically, *Train Time* is when the NTCM uses to classify the full training set, and *Test Time* is the time that the NTCM uses to classify the full test set [36].

4. Results

4.1 Feature reduction

The feature reduction performances by SVWE and SWEF were compared with the NDTMD, baseline (BoW), and state-of-the-art methods consisting of TF-IDF, LSA, PV-DM, and PV-DBOW. SVWE and SWEF produced high performance in reducing the number of features of the narcotic-drug dataset and had the highest FRR (0.9976), close to one. This result meant that not only did both methods provide high performance to reduce features, but they were also highly effective. However, SVWE and SWEF had the same reducing-feature performance as LSA, PV-DM, and PV-DBOW. The experimental results for feature reduction and FRR performance using SVWE and SWEF preprocessing are shown in Table 3.

Table 3 Features reduction and FRR performance

Data preprocessing methods	Numbers of remaining features	Feature reduction rate (FRR)
SVWE	100	0.9976
LSA	100	0.9976
PV-DM	100	0.9976
PV-DBOW	100	0.9976
SWEF	102	0.9976
NDTMD	1,555	0.9634
TF-IDF	6,768	0.8405
Bag of Words	8,028	0.8108

4.2 Classification performance

Table 4 presents the performance comparisons of the classification models generated by SVM, LR, J48, and CNN. The predictive results of the four classifiers used eight data preprocessing methods and were measured based on accuracy, F-measure, AUC, Kappa, MCC, and running time.

Table 4 Performance measurements of different data preprocessing methods

Data preprocessing methods	Classifier	Accuracy	F-measure	AUC	Kappa	MCC	Running time (s)
SWEF	SVM	0.8274	0.821	0.735	0.4998	0.504	3.97
	LR	0.8964	0.895	0.949	0.7131	0.714	1.04
	J48	0.8333	0.834	0.761	0.5528	0.553	0.87
	CNN	0.8285	0.822	0.854	0.5006	0.505	24.84
SVWE	SVM	0.8262	0.820	0.732	0.4955	0.500	4.66
	LR	0.8738	0.871	0.943	0.6438	0.646	1.04
	J48	0.8429	0.845	0.791	0.5878	0.589	0.83
	CNN	0.8488	0.848	0.877	0.5826	0.583	24.54
NDTMD	SVM	0.8833	0.884	0.846	0.6859	0.686	1.50
	LR	0.7369	0.750	0.739	0.3677	0.378	24.96
	J48	0.8893	0.892	0.923	0.7137	0.717	16.25
	CNN	0.8226	0.824	0.796	0.5248	0.525	28.35
TF-IDF	SVM	0.8691	0.865	0.782	0.5950	0.598	2.14
	LR	0.7821	0.784	0.738	0.3743	0.375	1,040.68
	J48	0.8810	0.881	0.871	0.6493	0.649	90.18
	CNN	0.8167	0.814	0.731	0.4468	0.447	127.98
Bag of words	SVM	0.8798	0.880	0.827	0.6493	0.649	2.43
(baseline)	LR	0.7667	0.779	0.798	0.3962	0.405	1,208.67
	J48	0.8774	0.882	0.900	0.6731	0.682	114.33
	CNN	0.8119	0.810	0.747	0.4392	0.439	266.14
LSA	SVM	0.8214	0.782	0.614	0.3029	0.374	5.60
	LR	0.8381	0.833	0.869	0.4951	0.498	1.11
	J48	0.7738	0.770	0.606	0.3147	0.315	1.75
	CNN	0.7941	0.797	0.781	0.4130	0.414	21.92
PV-DM	SVM	0.7821	0.687	0.500	0	0	4.92
	LR	0.7821	0.687	0.500	0	0	0.42
	J48	0.7429	0.688	0.462	0.0014	0.002	1.94
	CNN	0.6833	0.669	0.478	0.0087	0.009	26.99
PV-DBOW	SVM	0.7821	0.687	0.500	0	0	5.51
	LR	0.7702	0.727	0.690	0.1282	0.150	1.12
	J48	0.7750	0.693	0.671	0.0158	0.032	0.40
	CNN	0.7107	0.723	0.706	0.2342	0.238	25.93

The proposed method showed that SWEF with the LR classifier had the highest accuracy rate (0.8964). This combination also had the highest percentage of correctness than the baseline, state-of-the-art model, NDTMD, and SVWE. SWEF with the LR classifier had the highest F-measure (0.895), and that result showed near-perfect precision and recall. SWEF with the LR classifier had the highest AUC (0.949). The greater the AUC value, the higher the number of correctly accepted instances. SWEF with the LR classifier had a high Kappa (0.7131), which was higher than that of the baseline and state-of-the-art models. SWEF with the LR classifier had a high MCC (0.714), which was higher than that of the baseline and state-of-the-art models. Furthermore, the LR model from SVWE had a high accuracy rate (0.8738). However, NDTMD with the J48 classifier had the highest Kappa (0.7137) with the narcotic-drug dataset. NDTMD with the J48 classifier had the highest MCC (0.717), the value closest to one.

SVWE with the LR classifier, NDTMD with the SVM classifier, NDTMD with the J48 classifier, TF-IDF with the SVM classifier, TF-IDF with the J48 classifier, BoW with the SVM classifier, and BoW with the J48 classifier all had an accuracy rate of more than eighty-five percent (Table 4). For PV-DM and PV-DBOW with SVM, the LR classifier did not fit the narcotic drug messages in the Twitter dataset.

Thus, the best performance was for SWEF with the LR classifier, which provided the highest accuracy rate and F-measure and high Kappa and MCC values. In particular, SWEF with the LR classifier produced a low running time of 1.04 seconds (Table 4).

4.3 Running time performance

Figure 4 presents the running time performance of the top 8 best models. These models provided an accuracy rate of more than 85% (Table 4). Figure 4 shows that the proposed model (SWEF with the LR classifier) runs faster than competitive models, including NDTMD with the SVM classifier, TF-IDF with the SVM classifier, TF-IDF with the J48 classifier, BoW with the SVM classifier, and BoW with the J48 classifier. No model was faster than SVWE with the LR classifier. SWEF with the LR classifier gave better performance measurements than SVWE with the LR classifier (Table 4). Thus, the overall running time performance of SWEF with the LR classifier was better than that of the other paired models.

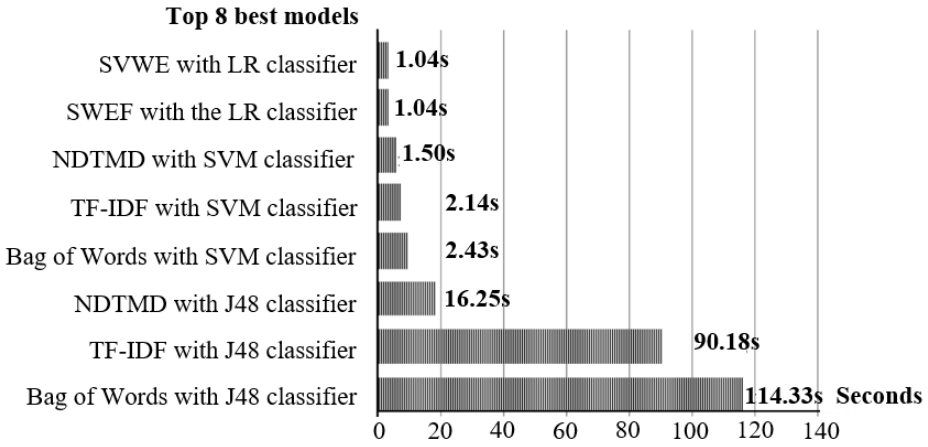

Figure 4 Running time performance of the top 8 best models

5. Discussion

The proposed SVWE effectively produced vectors because the optimization of the skip-gram model approximates the noise-contrastive estimation with a rough approximation. This technique subsamples frequent words and applies negative sampling, which reduces the computational burden of the training process and produces quality resulting vectors. SVWE applied by Chen et al. [9] used word embedding and calculated its average vector. In contrast, the proposed method 1) reduced the dimension size before calculating the average of the vector and 2) converted the NDTMD to sentences before finding the average of the vector.

Table 3 shows that the proposed SVWE had the highest FRR with high effectiveness of feature reduction. The SVWE dataset had the same number of features as LSA, PV-DM, and PV-DBOW. However, SVWE with the LR classifier had higher accuracy, F-measure, and AUC scores than LSA, PV-DM, and PV-DBOW (Table 4). Consequently, the SVWE dataset achieved a high performance when used with the LR classifier. This result indicated that the SVWE dataset works well with the LR classifier because LR is a mathematical function that supports binary class problems. The LR classifier function could be provided with high accuracy when used with the low-dimensional dataset.

Furthermore, SVWF with a CNN classifier was applied to develop the text classification model. It provided a high performance because convolutional features found an abuse message pattern in the SVWE dataset. Then, the pooling process detected messages involving drug abuse in the SVWE dataset. Abuse or non-abuse features appeared in the vectorization of the SVWE dataset.

The performance of the proposed SWEF had a high FRR, indicating a high performance of feature reduction. However, the SWEF dataset still had more features than LSA, PV-DM, PV-DBOW, and SVWE (Table 3). Subsequently, SWEF was used in the classification model. The running time of SWEF with the classification model was faster than BoW, TF-IDF, and NDTMD because the input data size of SWEF was smaller than BoW, TF-IDF, and NDTMD. Thus, assigning a smaller number results in classification efficiency. Table 4 also shows that the accuracy of SWEF with the LR classifier was better than other preprocessing techniques (BoW, TF-IDF, LSA, PV-DM, PV-DBOW, NDTMD, and SVWE). SWEF with the LR classifier had the highest performance evaluation based on the F-measure, AUC, and a low running time. SWEF with the LR classifier provided the highest accuracy value, that is, the highest percentage of prediction and the highest F-measure. This model provided excellent recall and precision. Additionally, high recall indicated an accurate prediction of the true positive abuse class.

The narcotic-drug messages dataset was imbalanced. However, SWEF with the LR classifier provided an AUC value close to one. The high AUC value suggested a high classification performance for a TP. When considering the TP value, the experimental results showed that the model could classify narcotic drug messages as abuse or non-abuse. There were no significant differences in Kappa and MCC values between SWEF with the LR classifier and NDTMD with the J48 classifier. SWEF with the LR classifier also provided a high Kappa value, indicating that this model had a high degree of consistent classification. The SWEF with the LR classifier provided a high MCC value, indicating the correlation coefficient between binary classes.

Figure 4 shows that SWEF with the LR classifier required less running time than other competitive models because the SWEF feature set had a smaller number of features than BoW, TF-IDF, and NDTMD. SWEF had the best performance because of additional features. The number of numerical digits and the number of words in each sentence increase the classification performance. This might have been because the news reports about drug arrests also mentioned the amounts of drugs seized, and drug-related messages were short. SWEF with the LR classifier provided higher performance than SVWE with the LR classifier because the additional features of the SWEF dataset could predict abuse and non-abuse classes. Additionally, *fnd* indicates potential features for the non-abuse class prediction, and *fnw* indicates potential features for the prediction of the abuse class.

Phan et al. [2] examined TF-IDF with J48 for developing narcotic drug classification and found that TF-IDF with J48 produced high-performance evaluators. However, SWEF with the LR classifier outperformed TF-IDF with J48. In addition, research on tuning SVM performance for sentiment analysis [10] showed that SVM worked best with the TF-IDF feature for posts from microblogging sites. However, Table 4 shows that LR using SWEF outperformed the SVM model with TF-IDF. In particular, the SVM classifier model from TF-IDF had a high performance in research on the CREES [11]. Table 4 shows that the LR classifier model from SWEF was superior to the SVM classifier model from TF-IDF. Rameshbhai and Paulose [12] investigated opinion mining using newspaper headlines. Their research reported the highest accuracy for linear SVM with TF-IDF. For the current study, Table 4 shows that the LR and SWEF provided a higher F-measure than the SVM and TF-IDF. Likewise, research comparing the data preprocessing methods for the Twitter content dataset [13] compared various classifier performance using BoW and TF-IDF. SVM with BoW had the highest accuracy. Table 4 shows that LR using SWEF has higher accuracy than the SVM classifier using BoW. Last, in the automated detection of abusive content on social media [9], the SVM model from the AVWE represented text vectorization by applying the SVWE from Chen et al. [9]. Table 4 shows that the LR classifier's performance from the SWEF method was better than the SVM model from SVWE.

SWEF was used to develop the NTCM because SWEF with the LR classifier provided the highest correctness percentage. LR is a better approach when considering binary-class problems. The narcotic drug messages are a binary class dataset having 4,200 instances. When preparing data with SWEF, the number of features was reduced to 102, and LR was fast because of this dataset's small size. Accordingly, the LR classifier model's function is strictly decreased by the smaller size of the dataset. Consequently, because of the smaller input text vectorization size, the classifier model from SWEF ran faster than the other methods.

6. Conclusion

This article proposed new data preprocessing methods for developing the NTCM. The proposed methods were SVWE and SWEF, which used feature reduction on the dataset using word embedding and feature extension. Both methods were used to prepare data for NTCM. This model focused on checking narcotic drug messages on Twitter in Asia. NTCM development consisted of four key processes as follows: 1) data were collected using keywords extracted from narcotic drug messages and Twitter data streams to retrieve narcotic drug messages on Twitter in Asia. The keywords were based on major sources consisting of the National Center on Addiction and Substance Abuse, American Addiction Centers, and *The Telegraph* newspaper. Synonym keywords were identified by measuring the cosine similarity with the GoogleNews-Vectors-Abuse 300 Model. The dataset was categorized using two labels (abuse and non-abuse) by experts in narcotic drug messages, and 2) data preprocessing consisted of data cleaning and data preparation for the classification model. Word embedding was generated from the corpus using the skip-gram model. The data preprocessing methods consisted of SVWE, SWEF, NDTMD, TF-IDF, BoW, LSA, PV-DM, and PV-DBOW. SVWE used the average vector of the word embedding. SWEF was performed by combining additional features with the SVWE feature set. The extended features were the set of numbers of words and the set of numbers of numerical digits calculated from the original data. Data preparation using SVWE and SWEF converted the narcotic drug messages into numerical vectors. 3) This study used SVM, LR, J48, and CNN to experiment with text classification models. Thirty-two text classification models were created from four classification algorithms using the above eight data preprocessing methods. 4) Comparing different models used accuracy, F-measure, AUC, Kappa, MCC, and running time as performance evaluators. The results showed that SWEF with the LR classifier had the highest performance evaluators and lowest running time. Thus, the LR algorithm with SWEF can be used for NTCM development to represent text vectorization in the SWEF data.

The research contributions of this article are data-preprocessing methods for SVWE and SWEF datasets. The methods produced small dimensional datasets. LR provided high performance based on both datasets. Moreover, this proposed method could prepare data for short text classification using short messages from Twitter. Furthermore, the NTCM can be used to develop a prototyping tool that might be used to detect narcotic drug messages on Twitter. This prototyping tool can be used by the Royal Thai Police and the Office of the Narcotics Control Board of Thailand.

For future work, the authors are interested in developing text classification models in more focused areas related to Thailand's drug problems and developing a data preprocessing method using the Thai language. This research problem is a challenge because a Thai sentence does not have spaces between words. The authors are also interested in investigating the time series classification model using Long Short-Term Memory Networks (LSTMs), a recent deep learning classifier for time series classification problems. Government officials want to record the timeline and location of their population due to the COVID-19 pandemic. The timeline record could be used to track the at-risk groups when COVID-19 cases are identified. Currently, people like to check in their events on Facebook, Twitter, and Instagram. Therefore, a time series classification model for various locations from messages is an interesting research problem. This model could help government officials track people when a COVID-19 patient is found at a check-in location.

7. Acknowledgement

The authors thank the Department of Computer Science, Faculty of Science, Kasetsart University, Bangkok, Thailand, for financial support and Professor Dr. Chia-Hui Chang at the National Central University, Taiwan, for providing many useful comments.

8. References

[1] United nations office on drugs and crime [UNODC]. Independent in-depth cluster evaluation of global research projects of the research and trend analysis branch. Vienna, Austria: United Nations Publication; 2018.
[2] Phan N, Chun SA, Bhole M, Geller J. Enabling real-time drug abuse detection in tweets. 2017 IEEE 33rd international conference on data engineering (ICDE); 2017 Apr 19-22; San Diego, USA. New York: IEEE; 2017. p. 1510-4.
[3] Deng X, Li Y, Weng J, Zhang J. Feature selection for text classification: a review. Multimed Tools Appl. 2019;78(3):3797-816.
[4] Kowsari K, Jafari Meimandi K, Heidarysafa M, Mendu S, Barnes L, Brown D. Text classification algorithms: a survey. Inform. 2019;10(4):1-68.
[5] Johns BT, Jamieson RK. A large-scale analysis of variance in written language. Cogn Sci. 2018;42(4):1360-74.
[6] Chayangkoon N, Srivihok A. Feature reduction of short text classification by using bag of words and word embedding. Int J Control Autom. 2019;12(2):1-16.
[7] Fathi E, Shoja BM. Deep neural networks for natural language processing. In: Gudivada VN, Rao CR, editors. Handbook of statistics. Netherlands: Elsevier; 2018. p. 229-316.
[8] Dhariyal B, Ravi V, Ravi K. Sentiment analysis Via Doc2Vec and convolutional neural network hybrids. 2018 IEEE symposium series on computational intelligence (SSCI); 2018 Nov 18-21; Bangalore, India. New York: IEEE; 2018. p. 666-71.
[9] Chen H, McKeever S, Delany SJ. A comparison of classical versus deep learning techniques for abusive content detection on social media sites. In: Staab S, Koltsova O, Ignatov D, editors. International conference on social informatics; 2018 Sep 25-28; Petersburg, Russia. Berlin: Springer; 2018. p. 117-33.
[10] Ahmad M, Aftab S, Bashir MS, Hameed N, Ali I, Nawaz Z. SVM Optimization for sentiment analysis. Int J Adv Comput Sci Appl. 2018;9(4):393-8.
[11] Burel G, Alani H. Crisis Event Extraction Service (CREES)-automatic detection and classification of crisis-related content on social media. In: Boersma K, Tomaszewski B, editors. Proceedings of 15th international conference on information systems for crisis response and management; 2018 May 20-23; Rochester, USA. p. 1-13.
[12] Rameshbhai CJ, Paulose J. Opinion mining on newspaper headlines using SVM and NLP. Int J Electr Comput Eng. 2019;9(3):2152-63.
[13] Pimpalkar AP, Raj RJ. Influence of pre-processing strategies on the performance of ML classifiers exploiting TF-IDF and BOW features. ADCAIJ. 2020;9(2):49-68.
[14] Soleimani BH, Matwin S. Spectral word embedding with negative sampling. Thirty-second AAAI conference on artificial intelligence; 2017 Feb 2-7; Louisiana, USA. California: AAAI Press; 2018. p. 5481-7.
[15] Grzegorczyk K, Kurdziel M. Disambiguated skip-gram model. Proceedings of the 2018 conference on empirical methods in natural language processing; 2018 Oct 31 – Nov 4; Brussels, Belgium. Stroudsburg: Association for Computational Linguistics; 2018. p. 1445-54.
[16] Merchant K, Pande Y. NLP based latent semantic analysis for legal text summarization. Proceedings of international conference on advances in computing, communications and informatics (ICACCI); 2018 Sep 19-22; Bangalore, India. New York: IEEE; 2018. p. 1803-7.
[17] Park EL, Cho S, Kang P. Supervised paragraph vector: distributed representations of words, documents and class labels. IEEE Access. 2019;7:29051-64.
[18] Guo H, Wang W. Granular support vector machine: a review. Artif Intell Rev. 2019;51(1):19-32.
[19] Zhang Z, Mo L, Huang C, Xu P. Binary logistic regression modeling with TensorFlow™. Ann Transl Med. 2019;7(20):591.
[20] Adnan M, Sarno R, Sungkono KR. Sentiment analysis of restaurant review with classification approach in the decision tree-J48 Algorithm. 2019 international seminar on application for technology of information and communication (iSemantic 2019); 2019 Sep 21-22; Semarang, Indonesia. New York: IEEE; 2019. p. 121-6.
[21] Georgakopoulos SV, Tasoulis SK, Vrahatis AG, Plagianakos VP. Convolutional neural networks for toxic comment classification. Proceedings of the 10th Hellenic conference on artificial intelligence; 2018 Jul 9-12; Greece. New York: Association for Computing Machinery; 2018. p. 1-6.
[22] National center on addiction and substance abuse, NCASA. Commonly used illegal drugs [Internet]. 2019 [cited 2019 Feb 2]. Available from: https://www.centeronaddiction.org/addiction/ commonly-used-illegal-drugs.
[23] Telegraph Media Group. Police given 3,000 Word 'a to z of drugs slang' to stay ahead of criminals [Internet]. 2019 [cited 2019 Jan 5]. Available from: http://www.telegraph.co.uk/news/uknews /law-and-order/6519172/Police-given-3000-word-A-to-Z-of-drugs-slang-to-stay-ahead-of-criminals.html.
[24] American Addiction Centers. Slang and nicknames for meth [Internet]. 2019 [cited 2019 Feb 6]. Available from: https://americanaddictioncenters.org/meth-treatment/slang-names.
[25] Google Code Project. Word2vec-Googlenews-Vectors [Internet]. 2018 [cited 2019 Jan 9]. Available from: https://www.kaggle.com/leadbest/googlenewsvectorsnegative300.
[26] R Core Teams. R: a language and environment for statistical computing, R Foundation for Statistical Computing [Internet]. 2019 [cited 2019 Jan 2]. Available from: http://www.R-project.org/.
[27] Xu Y, Goodacre R. On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. J Anal Test. 2018;2(3):249-62.
[28] Malkawi R, Saifan AA, Alhendawi N, BaniIsmaeel A. Data mining tools evaluation based on their quality attributes. Int J Adv Sci Tech. 2020;29(3):13867-90.

[29] Too J, Abdullah AR, Mohd Saad N, Tee W. EMG feature selection and classification using a Pbest-guide binary particle swarm optimization. Comput. 2019;7(1):1-20.
[30] Larner AJ. New unitary metrics for dementia test accuracy studies. Prog Neurol Psychiatry. 2019;23(3):21-5.
[31] Jabbar MA. Breast cancer data classification using ensemble machine learning. Eng Appl Sci Res. 2021;48(1):65-72.
[32] Hand D, Christen P. A note on using the F-Measure for evaluating record linkage algorithms. Stat Comput. 2018;28(3):539-47.
[33] Mingote V, Miguel A, Ortega A, Lleida E. Optimization of the area under the ROC curve using neural network supervectors for text-dependent speaker verification. Comput Speech Lang. 2020;63:1-16.
[34] De Raadt A, Warrens MJ, Bosker RJ, Kiers HA. Kappa coefficients for missing data. Educ Psychol Meas. 2019;79(3):558-76.
[35] Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC genom. 2020;21(1):1-13.
[36] Kasemtaweechok C, Suwannik W. Adaptive geometric median prototype selection method for K-Nearest neighbors classification. Intell Data Anal. 2019;23(4):855-76.