ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Sarcasm Messages Detection using Hybrid Features Extraction Deriving from Context and Content Sentences on Social Networks

Pramote Namwong[1], Panida Songram[2] and Kriangsak Rukpukdee[3]

ABSTRACT

This research aims to enhance the detection of sarcastic messages in the Thai language across social networks. It involves extracting and analyzing context-based features from messages to identify and differentiate sarcastic content. The study employs deep learning and machine learning techniques to classify these messages. The experimental findings demonstrate that a combination of context-based and content-based features yields the highest accuracy in identification. Specifically, the utilization of a bidirectional Long-Short Term Memory (Bi-LSTM) with 256 nodes, ReLU as the activation function, a dropout rate of 0.2, Sigmoid as the output activation function, binary cross-entropy as the loss function, and the Adam optimizer resulted in the highest accuracy achieved by the Bi-LSTM model, reaching 96.79%.

Article information:

Keywords: Sarcasm Detection, Machine Learning, Deep Learning, Text Classification

Article history:
Received: October 28, 2024
Revised: January 23, 2025
Accepted: March 29, 2025
Published: April 12, 2025 (Online)

DOI: 10.37936/ecti-cit.2025192.259379

1. INTRODUCTION

Communication on social networks has rapidly grown through text. This trend has increased comments expressing opinions and attitudes through online social networks. Opinions are expressions of the writer's emotions and feelings, which are categorized into three main aspects: positive, negative, and neutral. The opinions expressed through comments on social networks have been utilized for various analyses. For example, candidates running for elections used comments to survey public opinions about political parties or to present topics of interest to the public. Likewise, hotel service providers surveyed satisfaction levels to improve their services using feedback comments. Similarly, manufacturers and service providers have used feedback comments to evaluate consumer satisfaction and refine their offerings. Furthermore, educational institutions have utilized feedback comments to gauge student satisfaction with teaching methods and learning support, thereby improving teaching strategies for more effective learning management.

Computer-based techniques have been proposed for analyzing emotions, feelings, and attitudes. Among those techniques, sentiment analysis is a method for extracting emotions from extensive textual data. It has been widely studied and has various applications. For instance, Chan and Chong [1] investigated investors in the stock market by using sentiment analysis. Analyzing investor sentiment expressed through text has proven beneficial for predicting market trends. Tartir and Abdul-Nabi [2] presented sentiments, attitudes, and deep business insights extracted from Arabic Twitter social media. Examining varying sentiments aids in understanding the feelings of certain corporate entities, which are crucial for manufacturers' decision-making in planning future expansions. Gitto and Mancuso [3] analyzed airport service users' attitudes and satisfaction levels by examining customers' positive, negative, and neutral opinions to improve airport services based on the outcomes of this examination. Unfortunately, sentiment analysis may occasionally fall short of capturing genuine opinions. This limitation can introduce potential inaccuracies in data analysis. At times, the expressed text may not accurately reflect

[1]The author is with the Student in Computer Science from the Faculty of Informatics, Mahasarakham University (MSU), Thailand, Email: 60011260505@msu.ac.th
[2]The author is with the Faculty of Informatics, Mahasarakham University (MSU), Thailand. , Email: panida.s@msu.ac.th
[3]The author is with the Faculty of Business Administration and Management, Ubon Ratchathani Rajabhat University (UBRU), Thailand., Email: kriangsak.r@ubru.ac.th
[2]Corresponding author: panida.s@msu.ac.th

it's true intent, often exemplified by "sarcasm," where the text implies the opposite of its literal meaning. For example, "This weather is really, really good todayyyyyyyyyyyy!", "This hotel provides such excellent service that I do not want to come back and stay again soon.", and "I love being ignored all the time." The examples show that text written with positive words might aim to convey negative feelings, or some text using negative ones might intend to express positive emotions. Therefore, interpreting the meaning of words in these texts as strictly positive or negative is insufficient to comprehend the true sentiments.

Classifying sarcastic texts becomes a challenging and intricate task in sentiment analysis as it does not align with conventional positive or negative language patterns. Various techniques were presented to classify sarcastic texts. For example, Jasso and Meza [4] classified sarcastic texts in Spanish on Twitter using SVM and random forest. Bouazizi and Otsuki [5] utilized text patterns to detect sarcastic texts. Razali and Halin [6] showed an interest in studying the classification of sarcastic comments across various languages, which remains challenging due to the linguistic peculiarities inherent in each language. For instance, the Thai language is more complex compared to English. Classifying contentious texts in the Thai language is challenging and relatively new, lacking comprehensive studies in this domain.

In Thai, it is difficult to understand the real meaning via text, especially sarcastic text. Sarcasm text is the use of words to express something other than and especially the opposite of the literal meaning of a sentence [7, 8]. For example, "*วันนี้อากาศดีจริงเลยยยยยยยยยยยยยย*" ("This weather is really, really good todayyyyyyyyyyyy!"). The meaning of the words in the sentence shows the favorable weather, but the sentence expresses the unfavorable weather. This contradiction is emphasized through the repeated use of the consonant (ย), which adds a layer of sarcasm to the statement. In Thai, there are various formats and styles for writing sarcastic sentences, making checking for sarcasm challenging. Furthermore, sarcasm on social networks tends to be brief and lacks structure, amplifying the difficulty of discerning it. Our paper aims to address these challenges through two key contributions. 1) Context-based and content-based features are studied for detecting Thai sarcastic messages on social networks. 2) We conducted a comparative analysis that includes machine learning models (SVM, Naive Bayes, Decision tree, and KNN) and deep learning models (DNN, Bi-LSTM) to find the best model for detecting Thai sarcasm messages on social networks.

2. RELATED WORKS

Understanding sarcasm texts contributes to more accurate sentiment analysis and enhances data-driven business decision-making. Many techniques have been proposed to understand sarcastic texts. For example, Karkiner and Sert [9] explored sarcasm detection, highlighting trade-offs between accuracy and efficiency in training time. They evaluated deep learning models, including BERT, RNN, Bi-LSTM, and GRU, for classification accuracy and training on a sarcastic and non-sarcastic news headline dataset. With the models, BERT achieved the highest accuracy at 88%, while RNN demonstrated the best performance in training time. Helal *et al.* [10] addressed a challenge by incorporating specific contextual cues to improve sarcasm detection accuracy. The pre-trained transformer models, RoBERTa and DistilBERT, were fine-tuned on the news headlines and mustard datasets for sarcasm detection and achieved an F-measure of 99% and 90%, respectively.

Additionally, summarizing context into concise sentences reduced training time by 35.5%. Validating the RoBERTa model on the Reddit dataset highlighted the importance of context, with F-measure increasing from 49% (without context) to 75% (with context). Vitman *et al.* [11] combined various features to capture sarcasm using a pre-trained transformer and CNN. They applied sentiment and emotion models to extract features and fine-tuned the Transformer and CNN blocks to optimize performance. Experiments conducted on four datasets from different domains showed that the approach outperforms previous state-of-the-art models in sarcasm detection across social media and online platforms. Bari [12] studied leveraging neural networks to build a sarcasm detection model capable of distinguishing between sarcastic and non-sarcastic statements using sequences from collections of news headlines. Kumar and Sarin [13] proposed a novel approach combining word embeddings with context-aware language models to automate feature extraction and improve classification performance. Results showed that FastText embeddings paired with the BERT language model achieved the highest accuracy in identifying sarcasm across three benchmark datasets. Kunneman *et al.* [14] presented a system for detecting sarcasm in tweets and microblog posts on Twitter. Tweets marked with hashtags such as ♯sarcasm or ♯not were included in a dataset comprised of 2.25 million Dutch tweets, with 406,000 tweets. A winnow classification algorithm was employed and achieved an AUC score of 84%. Vateekul and Koomsubha [15] investigated sentiment analysis of Thai text on Twitter using LSTM and Dynamic Convolutional Neural Network (DCNN) techniques. The performance of the models was compared against that of traditional machine learning algorithms, including Naive Bayes, SVM, and maximum entropy. Both LSTM and DCNN outperformed the other algorithms. DCNN achieved the highest accuracy at 75.35%.

Furthermore, many studies investigated feature extraction for sarcasm detection. For example, Bouazizi and Ohtsuki [16] studied sarcasm detection in microblogging, explicitly focusing on Twitter. They proposed a pattern-based approach for identifying sarcasm with their dataset collected through Twitter's streaming API. The dataset included 58,609 tweets. This study examines four categories of features: 1) Sentiment-related features: two lists of words with positive and negative emotional content were created using the SentiStrength database. 2) Punctuation-related features: punctuation-based features were also extracted, such as the number of exclamation marks, question marks, ellipses, all-capital words, quotation marks, and vowels repeated more than twice (e.g., "loooooove"). 3) Syntactic and semantic features: features were reflected in syntactic and semantic sarcasm indicators, including the presence of uncommon words, the number of such words, and common sarcastic expressions, interjections, and laughing expressions. These features, combined with punctuation-related data, help in detecting sarcasm. 4) Pattern features: part-of-speech tagging was used to define word patterns in each tweet and then each tweet was transformed" into a vector of words according to predefined rules. The four categories were used to classify sarcasm with various algorithms: random forest, SVM, KNN, and maximum entropy. The experimental results indicated that the random forest classifier achieved the highest accuracy of 83.1% with an F-measure of 81.3%.

Rajadesingan *et al.* [17] proposed a framework to enhance sarcasm detection through behavioral analysis. The data collection process involved retrieving sarcastic tweets using the keywords "sarcasm" and "not" with Twitter's Streaming API. During data pre-processing, non-English tweets, retweets, and tweets containing mentions or URLs were filtered out. Additionally, tweets tagged with "sarcasm" and "not" were stripped of hashtags before the evaluation stage. This tested various class distributions to evaluate the framework's performance, including ratios of 1:1, 10:90, and 20:80, where a 1:1 distribution means each sarcastic tweet in the dataset is matched with one non-sarcastic tweet. This tested the framework by executing sarcasm classification multiple times, achieving an accuracy of 79.38%, which outperformed all baseline models. Moreover, feature analysis identified the top 10 features critical to sarcasm detection, listed in order of importance: 1) percentage of emotions in the tweet, 2) percentage of adjectives in the tweet, 3) percentage of past words with a sentiment score of 3, 4) number of polysyllables per word in the tweet,5) lexical density of the tweet, 6) percentage of part words with a sentiment score of 2, 7) percentage of past words with a sentiment score of -3, 8) number of past sarcastic tweets posted by the user, 9) percentage of positive-to-negative sentiment transitions made by the user, 10) percentage of capitalized hashtags in the tweet.

Jasso and Meza [4] introduced two approaches, a word-based and a character-based method, for irony detection in Spanish tweets. The word-based approach utilized word n-grams for classification. On the other hand, the character-based approach employs character n-grams, capturing features such as average word length and relevant characteristics like punctuation marks and emojis. Experiments were conducted using both SVM and random forest classifiers, with the term frequency-inverse document frequency (TF-IDF) applied to both word and character n-grams. Experimental results showed that the character level achieved F-measures of 87% on a balanced dataset and 80% on an unbalanced dataset when using a random forest classifier.

Hiai and Shimada [18] presented an approach to identifying patterns for evaluating sarcastic texts, particularly in the context of product reviews. This study extracted sarcastic sentences from product reviews by categorizing sentences into eight distinct classes; 1) expressing both positive and negative sentiments in the same sentence, typically with low ratings, 2) positive sentiment in the review title contrasted by negative sentiment in the review body, emphasizing the negative aspect, 3) using "ii" (meaning "good" in English), where the combination with phrases like "nedan" (price) alters the meaning to "too expensive," creating a sarcastic positive expression, 4) focusing on delivery aspects rather than the product, implying dissatisfaction with the product through positive language, 5) positive phrases used in negative contexts, such as "ii (better)" followed by "losing money again," conveying an overall negative sentiment, 6) positive statements about other products, suggesting the product in question is inferior, 7) use of uncommon positive expressions imply a negative view of the product, 8) positive statements that fail to highlight genuine advantages, suggesting the product lacks notable strengths.

Karoui *et al.* [19] presented methods for classifying sarcastic texts in Arabic on social media networks. A dataset comprised 5,479 texts, with an equal distribution of 1,733 sarcastic and 1,733 non-sarcastic texts. It was extracted into two categories of features: 1) Surface features were extracted from symbols and punctuation, such as emoticons, quotation marks, contrasting words (e.g., "but"), the frequency and sequence of exclamation and question marks, their combinations, non-conflicting quotes, interjections, and the number of emoticons. 2) Sentiment features were extracted from a lexicon-based approach, such as Bing Liu's lexicon, MPQA subjectivity lexicon, and Arabic emoticon lexicon. The experiments were conducted using the Weka toolkit, testing algorithms including SVM, Naive Bayes, logistic regression, linear regression, random tree, and random forest. The random forest achieved the best performance, with an accuracy of 72.36%, precision of 72.90%, Recall of 73.50%, and an F-measure of 72.70%.

The review of related works indicates that sarcasm detection remains a challenging task. Studies are ongoing to enhance sarcasm classification performance across various languages, each with unique linguistic structures. However, no research has focused on detecting sarcastic texts in Thai social media messages. Consequently, this study aims to study context and content-based features and explore methods for identifying sarcasm in Thai-language social media texts.

3. METHODOLOGY

This paper examines context-based and content-based features for detecting sarcasm in text data obtained from Facebook. The overall process is illustrated in Fig. 1 and consists of five main components: data collection, pre-processing, feature extraction, model training, and performance evaluation. The details of each component are as follows.

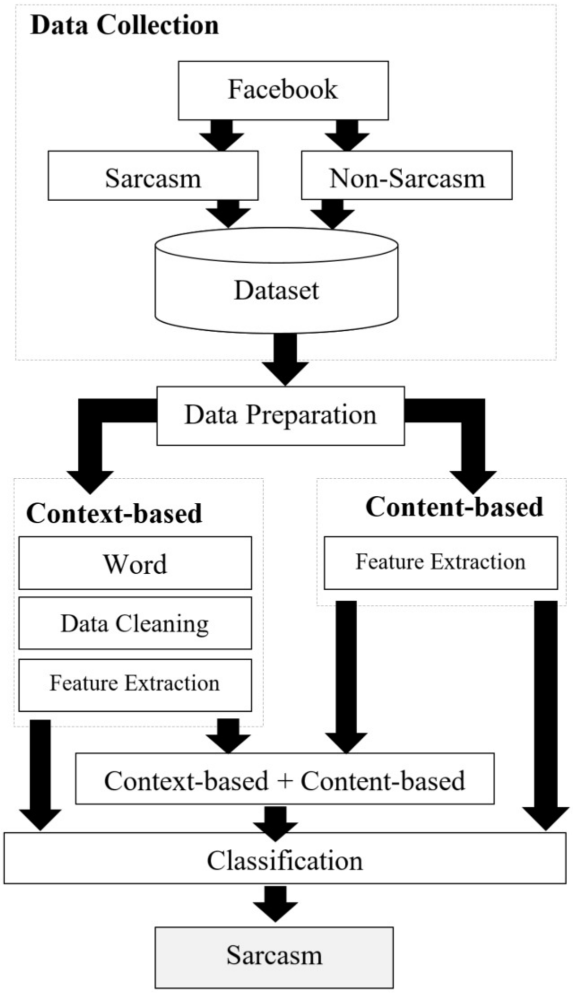


Fig.1: *The proposed methodology.*

3.1 Data Collection

This paper collected text data from Facebook containing the hashtag as in previous studies [5, 14, 17, 20-22]. The previous studies found that data aggregation on conflict was carried out using keyword-based data collection, specifically through hashtag searches. For example, English text utilized keyword searches ♯sarcasm, ♯sarcastic, and ♯irony to collect sarcasm text. Dutch text [14] utilized the keyword ♯sarcasm. Therefore, this research employed a data aggregation method utilizing hashtags, specifically collecting data with the following hashtags. Messages with hashtag ♯ประชด(♯Sarcasm) and ♯ประชดประชัน(♯Sarcasm) were collected as sarcasm and non-sarcasm messages were gathered using hashtags ♯สิ่งดีดี(♯“GoodThings”), ♯คิดดี (♯“PositiveThinking”), ♯มีสุข (♯“BeHappy”), ♯ความสุข (♯“Happiness”), ♯โชคดีจัง (♯“SoLucky”). Finally, the dataset comprises 5,400 sarcasm messages and 5,400 non-sarcasm messages. Examples of those messages are depicted in Fig. 2.

ข้อความ	คลาส
ยุติธรรมจริงๆๆๆๆ	Sarcasm
ชื่นชมในความคิดสร้างสรรค์🤣🤣🤣	Sarcasm
เมื่อสั่งให้เพื่อนไปเด็ดกระถินมากินกับส้มตำโถเด็ดมาทีกินได้ทั้งหมู่บ้าน😒😒😒5555	Sarcasm
ย..ยายทองดีเอ้ยนี่ถ้าไปด้วยนะให้ใส่หมวกครอบผมอาบน้ำไปเลย🤣🤣🤣🤣🤣🤣	Sarcasm
อยู่คนเดียวไม่เหงานะจริงๆแล้วการอยู่คนเดียวมันอิสระและโคตรจะมีความสุขเลย😅😅😅	Sarcasm
มีความสุขต้อนรับวันแรกของปีพ่อแม่ลูก	Non-Sarcasm
ยิ้มชิยิ้มมมมมมมีความสุข	Non-Sarcasm
ทำบุญวันปีใหม่มีความสุข	Non-Sarcasm
มีความสุขอย่างบอกไม่ถูกทุกอย่างอยู่ที่ตัวเรา	Non-Sarcasm
มีความสุขทุกครั้งที่ได้อยู่ด้วยขี้ดื้อน้อย	Non-Sarcasm
ร๊ากน่ะคนดีมีความสุข	Non-Sarcasm
มีความสุขจังถ้าชีวิตมีแต่ความทุกข์ต้องหาความสุขใส่ตัวเรามีความสุขที่ได้อยู่กับครอบครัวลูกหลาน	Non-Sarcasm

Fig.2: *Example of dataset.*

3.2 Data Pre-processing

The dataset was pre-processed before extracting features to perform a classification. Each message was segmented to find a collection of words. Separating words is carried out by using Thai word segmentation methods [23, 24]. Stop word rejection is then performed to remove common words that do not contain important information to be used for processing tests in the language. The remaining words in messages will be used to extract features. In this work, two featured-based techniques are used for the classification. i.e., 1) context-based and 2) content-based. In context-based, special characters (! “ ‘ @ ♯ $ % ^ * ( ) − + = ? . / < > _) were removed as they are no important fragments to represent messages in interns of the context description.

3.3 Feature extraction

Two feature-based approaches are implemented to classify messages into sarcasm and non-sarcasm. The context-based technique considers the information around a single word, while content-based features examine the characteristics of a special pattern of characters. The feature extraction of these two approaches is explained below.

1) Context-based approach: Contextual information can be used to describe messages to identify sarcasm. All distinct words in the dataset were collected and used as a set of features containing 12,973 features. A message was pre-processed to a vector containing 12,973 features. Given a vector $m = \{f_1, f_2, \ldots, f_n\}$, f_i is a feature, and n is the number of features. The weight of feature f_i is assigned using the following methods:

- Boolean weighting: the weight of feature f_i is assigned to 1 if f_i is in the message and zero otherwise.

- TF weighting: the weight of feature f_i is assigned to the number of the occurrence of f_i in the message.

- TF-IDF weighting: the weight of feature f_i is assigned as equation (1), where tf is the number of the occurrence of f_i in the message, and IDF is calculated by using equation (2), where N is the overall number of messages, and n is the number of messages containing f_i.

$$w(f_i) = tf * idf \tag{1}$$

$$idf = \log\left(\frac{1+N}{1+n}\right) + 1 \tag{2}$$

- Word embedding [25, 26] is a method for transforming a feature in a message into a vector. This involves encoding each word in a message into a numerical format that computers can process and understand. A key advantage of word embedding is its ability to capture semantic similarities between words within different contexts, enabling meaningful calculations based on word relationships. This work adopted the CBOW model (Continuous Bag-of-Words) to generate word embedding. It typically starts with encoding each word using one-hot encoding, and then embedding is learned by predicting the target word based on context words. An example of word embedding used to transform text into vector form is shown in Fig. 3.

2) Content-based approach: This work also investigates content-based features. The content-based features were extracted from the indicators (uncommon words) relating to sarcasm. In this work, 15 indicators are deployed to detect sarcasm messages. The indicators are as follows, and examples are shown in Fig 4.

ชอบ (like): [0.12, -0.34, 0.56, -0.78, 0.89, -0.23, 0.45, -0.67, 0.11, -0.56]
ไม่ชอบ (dislike): [-0.45, 0.67, -0.12, 0.34, -0.78, 0.89, -0.23, 0.56, -0.11, 0.45]
ดี (good): [0.78, -0.56, 0.23, -0.89, 0.45, -0.67, 0.12, -0.34, 0.56, -0.78]
ไม่ดี (bad): [-0.89, 0.45, -0.67, 0.12, -0.34, 0.56, -0.78, 0.89, -0.23, 0.45]

Fig.3: *Example of vectors using word embedding encoding.*

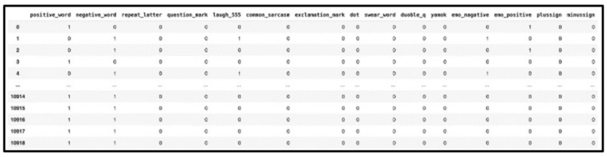
Indicator 1: having a message that shows a positive emotions.
Indicator 2: having a message that shows a negative emotions.
Indicator 3: having a message that shows an abnormal usage of words.
Indicator 4: number of question marks (???).
Indicator 5: having a laughing word.
Indicator 6: having general sarcasm words.
Indicator 7: number of exclamation marks (!).
Indicator 8: number of dots (...).
Indicator 9: using rude words in sentences.
Indicator 10: number of quotation marks (“”).
Indicator 11: number of the repeat symbol of Thai writing system represented by ๆ.
Indicator 12: having icons that show negative emotions.
Indicator 13: having icons that show positive emotions.
Indicator 14: number of plus sign (+ + +).
Indicator 15: number of a minus sign (− − −).

Indicator 1: กตเวที | กตัญญู | กล้า | กรุณา | กันเอง | แข็งแกร่ง | แข็งขัน
Indicator 2: งอแง | โง่ | ชั่ว | ทราม | เฉื่อย | ตอแหล | ต่ำ
Indicator 3: มากกกกก | ยังงงงงง | เลยยยยยย | เหรอออออออ | แล้วววววว | เรียบร้อยยยยยย
Indicator 4: สวยจัง????? | หล่อ????? | ใช่???
Indicator 5: 5555 | ฮ่าฮ่า .
Indicator 6: เกิ๊น | จริ๊ง | เลี๊ย | ใช่สิ๊ | จริงจริ๊ง | เริ้ย.
Indicator 7: รัก!!! | น่ารัก!!!!!
Indicator 8: จริง..... | ใช่.... | นั่นสิ...
Indicator 9: ฟาย | สฤล | แรด | เชี่ย | ระยำ
Indicator 10: ””””” | “““““
Indicator 11: ๆๆๆๆ.
Indicator 12:
Indicator 13:
Indicator 14: +++++.
Indicator 15: -----.

Fig.4: *Example of 15 indicators.*

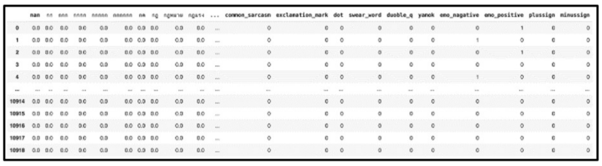
To create a vector generated by the 15 indicators, the weights of indicators 1, 2, 3, 5, 6, 9, 12, and 13 are assigned to 1 if the indicator is found in the message and zero otherwise. Fig. 5 shows an example of a vector.

In addition, the feature fusion was performed to detect sarcasm messages. The context-based vector and content-based vector of the same messages were concatenated to a vector, as shown in Fig. 6.



	positive_word	negative_word	repeat_latter	question_mark	laugh_555	common_sarcasm	exclamation_mark	dot	swear_word	duoble_q	yamok	emo_nagative	emo_positive	plussign	minussign
0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10914	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
10915	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
10916	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
10917	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
10918	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig.5: *Example of vectors generated by15 indicators.*



	nan	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	common_sarcasn	exclamation_mark	dot	swear_word	duoble_q	yamok	emo_nagative	emo_positive	plussign	minussign
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	0	0	0	0	0	1	0	0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	0	0	0	0	1	0	0	0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	0	0	0	0	0	1	0	0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	0	0	0	0	0	0	0	0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	0	0	0	0	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10914	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	0	0	0	0	0	0	0	0
10915	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	0	0	0	0	0	0	0	0
10916	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	0	0	0	0	0	0	0	0
10917	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	0	0	0	0	0	0	0	0
10918	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	0	0	0	0	0	0	0	0

Fig.6: *Example of vector fusion.*

3.4 Learning models

After extracting features, feature vectors will be fed to learning algorithms: SVM, Naive Bayes, Decision Tree, KNN, DNN (Deep Neural Network), and Bi-LSTM. A linear kernel was set for SVM to learn and create a classifier. Decision Tree utilizes the ID3 algorithm, selecting Gini as the criterion parameter. KNN employs Euclidean distance and K=5 for learning.

DNN is configured with the following learning parameters: dense layer units = 512, activation function = ReLU, dropout rate = 0.2, output activation = Sigmoid, loss function = binary cross-entropy, and optimizer = Adam. The proposed structure of DNN is shown in Fig. 7. For Bi-LSTM is configured with the following learning parameters: bidirectional LSTM with 256 nodes, activation function = ReLU, dropout rate = 0.2, output activation = Sigmoid, loss function = binary cross-entropy, optimizer = Adam. The proposed structure of Bi-LSTM is shown in Fig. 8.

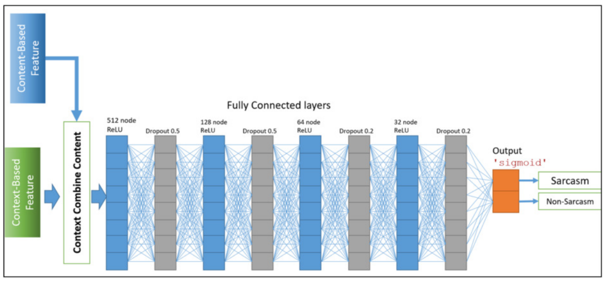


Fig.7: *The proposed DNN model.*

3.5 Performance Evaluation

Several metrics were employed to evaluate the performance of the sarcasm detection models. The classification results were assessed using k-fold cross-validation, where the dataset was partitioned into k subsets. The model was trained on k-1 subsets and evaluated on the remaining subset, with the process repeated k = 10 times, and the results averaged over all times. The following metrics were calculated:

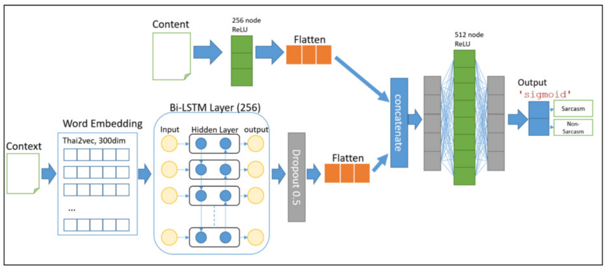


Fig.8: *The proposed Bi-LSTM model.*

Accuracy measures the proportion of correct predictions the model makes over the total number of testing instances. It is defined as $Acc = (TP + TN)/(TP + TN + FP + FN)$, where TP represents the number of sarcastic messages correctly classified, and TN is the number of non-sarcastic messages correctly classified. FN refers to the number of sarcastic messages misclassified as non-sarcastic, and FP refers to the number of non-sarcastic messages misclassified as sarcastic.

Precision is the number of correct predictions from all predictions for each class. Precision for sarcasm is defined as $Psm = TP/(TP + FP)$. Precision for the non-sarcasm is defined as $Pnsm = TN/(TN + FN)$.

The F-measure is the harmonic mean of precision and recall, providing a balanced measure that considers both metrics. F-measure for sarcasm and non-sarcasm are defined $Fsm = 2\times(Psm\times Rsm)/(Psm+Rsm)$ and $Fnsm = 2 \times (Pnsm \times Rnsm)/(Pnsm + Rnsm)$

These metrics comprehensively evaluate the model's performance, assessing its ability to correctly classify sarcastic and non-sarcastic texts.

4. RESULTS

This research experiment involves utilizing context-based and content-based feature extraction from text. The experiment was conducted in three formats: 1) using only context-based features, 2) using only content-based features, and 3) combining context-based and content-based features. The experimental results are as follows:

4.1 Experimental results of context-based features

First, an experiment compared the removal and non-removal of stop words. Typically, removing stop words reduces the dataset's features, enabling faster processing without significantly altering experimental outcomes. The experiment assigned the weight of features by using Boolean weighting to showcase the impact of removing and retaining stop words. Table 1 indicates that retaining stop words yields nearly equivalent accuracy across most algorithms, except for the KNN method.

Table 1: *Stop words removal effective (SM = sarcasm, NSM = non-sarcasm).*

ACC	Precision		Recall		F-measure	
	SM	NSM	SM	NSM	SM	NSM
With Stop words						
KNN						
71.72	97.86	45.59	64.26	95.54	77.56	61.68
SVM						
89.31	92.95	85.68	86.66	92.39	89.69	88.90
Decision-Tree						
84.94	89.92	79.96	81.78	88.82	85.65	84.15
Naïve Bayes						
86.38	85.58	87.19	87.02	85.77	86.28	86.46
DNN						
89.38	89.27	89.49	89.27	89.49	89.27	89.49
Bi-LSTM						
92.96	90.98	87.83	88.18	90.75	89.55	89.26
Without Stop words						
KNN						
75.78	86.70	64.83	71.16	83.03	78.14	72.77
SVM						
86.64	90.60	82.71	83.95	89.76	87.14	86.08
Decision-Tree						
82.44	87.19	77.71	79.61	85.86	83.21	81.56
Naïve Bayes						
85.91	84.43	87.47	87.04	84.86	85.69	86.11
DNN						
87.81	88.90	86.74	86.78	88.86	87.83	87.79
Bi-LSTM						
91.39	90.45	89.58	89.70	90.38	90.03	89.94

Table 2 shows the performance of all models on the dataset extracted by using the Boolean, TF, and TF-IDF weighting methods. It can be observed that Bi-LSTM achieves the highest accuracy, yielding 91.39%, 91.31% and 91.31% when using Boolean, TF, and TF-IDF weighting methods, respectively. Additionally, Bi-LSTM provides the highest recall for predicting sarcastic texts on the three weighting methods, with recall values of 89.70%, 91.70%, and 95.01% when using TF and TF-IDF weighting methods, respectively. On the other hand, KNN offers the highest precision for predicting sarcastic texts, reaching 97.67%, 96.51%, and 95.96% when using TF and TF-IDF weighting methods, respectively. However, when considering the F-measure, Bi-LSTM yields the highest value at 90.03%, 90.43%, and 92.48% when using TF and TF-IDF weighting methods, respectively.

The Bi-LSTM model demonstrates the best overall predictive performance when extracting features using the three weighting methods. The model's functionality, which is suitable for continuous value computation and adept at handling textual data patterns, contributes to its efficient overall performance. Its bidirectional computation also enhances learning textual patterns, achieving optimal overall performance.

Table 2: *The performance of context-based features with Boolean, TF, and TF-IDF weighting.*

Weighting methods	ACC	Precision		Recall		F-measure	
		SM	NSM	SM	NSM	SM	NSM
	KNN						
Boolean	69.07	**97.67**	44.26	63.67	94.97	77.08	60.36
TF	71.85	**96.51**	49.36	65.59	93.43	78.09	64.55
TF-IDF	71.57	**95.96**	47.28	64.56	92.24	77.18	62.46
	SVM						
Boolean	84.44	85.90	82.78	83.33	85.42	84.59	84.07
TF	83.24	85.53	83.17	83.60	85.15	84.54	84.14
TF-IDF	84.81	89.16	82.61	83.71	88.37	86.34	85.38
	Decision-Tree						
Boolean	77.50	79.99	77.23	77.83	79.43	78.88	78.30
TF	77.04	80.76	77.41	78.13	80.10	79.41	78.72
TF-IDF	74.63	78.88	76.94	77.39	78.43	78.12	77.67
	Naïve Bayes						
Boolean	84.07	82.83	86.61	86.14	83.44	84.44	84.98
TF	83.52	83.26	85.39	85.10	83.59	84.16	84.47
TF-IDF	83.61	78.10	89.13	87.79	80.28	82.65	84.46
	DNN						
Boolean	85.65	84.20	87.09	86.82	84.57	85.47	85.79
TF	86.11	85.75	86.89	86.77	85.92	86.23	86.38
TF-IDF	84.44	86.31	85.30	85.51	86.20	85.87	85.70
	Bi-LSTM						
Boolean	**91.39**	90.45	89.58	**89.70**	90.38	**90.03**	89.94
TF	**93.61**	89.60	91.69	**91.31**	90.11	**90.43**	90.87
TF-IDF	**93.61**	95.22	89.31	**89.91**	95.01	**92.48**	92.06

Table 3 shows the performance of Bi-LSTM employing word embedding for feature extraction. It shows that Bi-LSTM yields an accuracy of 91.39% and a precision of 90.45% for predicting sarcastic texts. Comparing the use of Boolean, TF, TF-IDF weighting, and word embedding methods in the Bi-LSTM model, the overall performance, based on the F-measure, shows that the Bi-LSTM gives the highest performance when using the TF-IDF weighting method for context feature extraction.

Table 3: *The performance of Bi-LSTM using word embedding.*

ACC	Precision		Recall		F-measure	
	SM	NSM	SM	NSM	SM	NSM
Bi-LSTM						
91.39	90.45	89.58	89.70	90.38	90.03	89.94

4.2 Experimental results of Content-based features

Table 4 displays the experiment using 15 content features. The DNN model achieves the highest accuracy of 80.86%.

Table 4: *The performance of content-based features.*

ACC	Precision		Recall		F-measure	
	SM	NSM	SM	NSM	SM	NSM
KNN						
49.17	99.89	0.23	50.03	43.81	66.65	0.45
SVM						
80.00	72.09	88.98	86.72	76.13	78.71	82.03
Decision-Tree						
80.00	72.04	89.01	86.75	76.10	78.69	82.03
Naïve Bayes						
78.15	57.66	97.57	96.01	69.73	71.93	81.27
DNN						
80.86	71.80	89.74	87.26	76.47	78.78	82.57
Bi-LSTM						
80.37	80.67	81.56	81.46	80.91	**80.99**	81.16

4.3 Experimental results of context-based combined with Content-based features

The experiment used features from context-based combined with content-based features, weighted using Boolean, TF, and TF-IDF weighting, as depicted in Table 5. It illustrates that the Bi-LSTM model yields the highest accuracy, achieving 95.46%, 95.37%, and 96.67% when using Boolean, TF, and TF-IDF weighting methods, respectively. Additionally, it provides the best recall in predicting sarcastic texts, scoring at 92.44%, 94.08%, and 92.05% when using Boolean, TF, and TF-IDF weighting methods, respectively. On the other hand, the KNN model still demonstrates the highest precision in predicting sarcastic text, achieving 95.28% when using TF weighting method. However, when considering the F-measure, Bi-LSTM still yields the highest score at 91.42%, 93.15%, and 92.93% when using Boolean, TF, and TF-IDF weighting methods, respectively.

Table 6 shows the performance of the Bi-LSTM model when using context-based combined with content-based features, weighted using word embedding. It shows that Bi-LSTM achieved highest accuracy of 96.79% among all previously mentioned methods. It also provided the highest recall, precision, and F-measure for predicting sarcastic and non-sarcastic texts. Overall, using context-based combined with content-based features, weighted using word embedding, demonstrates the highest performance in our comprehensive experimentation.

All the experimental results show that utilizing context-based and content-based features yields better performance than using only context-based or content-based features alone. Utilizing context-based features is insufficient in distinguishing sarcastic texts, as some texts convey opposite meanings to the actual intention of the communicator. Using context-based features alone leads to poor classification of sarcastic texts. On the other hand, employing only content-based features does not give

Table 5: *The performance of context-based+content -based features with Boolean, TF, TF-IDF weighting.*

Weighting methods	ACC	Precision		Recall		F-measure	
		SM	NSM	SM	NSM	SM	NSM
	KNN						
Boolean	77.59	92.77	64.63	77.45	89.94	81.34	75.16
TF	76.11	95.28	58.08	69.49	92.58	80.34	71.29
TF-IDF	83.52	77.51	91.06	89.71	80.19	83.14	85.26
	SVM						
Boolean	87.87	87.69	87.43	87.48	87.63	87.58	87.52
TF	87.22	87.59	86.29	86.49	87.43	87.02	86.84
TF-IDF	88.15	87.61	89.80	89.60	87.85	88.85	88.80
	Decision-Tree						
Boolean	81.67	84.06	83.32	83.46	83.93	83.75	83.62
TF	82.68	83.76	83.48	83.53	83.71	83.64	83.59
TF-IDF	81.57	82.75	83.46	83.34	82.88	83.04	83.16
	Naïve Bayes						
Boolean	88.06	88.11	89.37	89.27	88.38	88.68	88.81
TF	87.31	88.78	87.46	87.66	88.62	88.21	88.03
TF-IDF	88.55	77.17	94.75	93.64	80.58	84.60	87.09
	DNN						
Boolean	89.81	88.66	89.20	89.14	88.69	88.89	88.94
TF	89.81	88.32	90.26	90.08	88.55	89.18	89.38
TF-IDF	90.00	87.48	89.66	89.48	87.76	88.44	88.67
	Bi-LSTM						
Boolean	**95.46**	90.47	92.63	**92.44**	90.74	**91.42**	91.66
TF	**95.37**	92.27	94.29	**94.08**	92.54	**93.15**	93.39
TF-IDF	**96.67**	93.91	91.97	**92.05**	93.90	**92.96**	92.91

Table 6: *The performance of Bi-LSTM using word embedding.*

ACC	Precision		Recall		F-measure	
	SM	NSM	SM	NSM	SM	NSM
Bi-LSTM						
96.79	98.80	94.72	95.08	98.79	96.88	96.68

high performance in the classification of texts, although content-based features are derived from sarcastic texts. Therefore, combining both sets of features is suitable for the classification of sarcastic texts, as they represent a better proxy for sarcastic text patterns, resulting in improved classification performance.

Word embedding gives the highest performance for sarcasm detection among the feature extraction methods. It has advantages in learning text patterns and can analyze similarities in text patterns. Among the models, the Bi-LSTM gives the highest accuracy because it captures bidirectional text patterns and significantly enhances the ability to comprehend a broader range of text patterns. Thus, Bi-LSTM is the most suitable model for classifying sarcastic texts on Thai-language online social networks.

5. CONCLUSION

This paper investigates the detection of Thai sarcastic texts on social networks. The dataset used for experimentation was gathered from Facebook users by employing a method that collects data from Facebook through hashtag searches for ♯sarcasm. Two types of features were analyzed for Thai sarcasm detection. Context-based features are extracted from the text's surrounding context, and content-based features are derived from the text's content. To extract context-based features, texts were cleaned by removing English text, numbers, and special characters. Then, they were segmented to get the set of features (words). Features were weighted by using Boolean, TF, TF-IDF, and word embedding methods. For content-based features were derived from the text's content, such as indicators of positive or negative emotions, unusual word usage, exclamation marks, laughter expressions, general sarcasm indicators, apostrophes, profanity, ellipses, emoticons conveying positive or negative emotions, as well as plus and minus signs. The learning models, SVM, Naive Bayes, Decision tree, KNN, DNN, and Bi-LSTM, were also investigated to detect sarcastic texts on context- and content-based features. From the investigation, Bi-LSTM reached the highest F-measure, 92.48%, using only context-based features weighted by the TF-IDF weighting method. Moreover, using only content-based features, Bi-LSTM achieved the highest F-measure, 80.99%. However, utilizing all extracted features achieved the highest classification performance in sarcasm detection. Bi-LSTM could reach an accuracy of 96.79% and an F-measure of 96.88% for detecting sarcastic Thai texts. Therefore, the combination of context-based and content-based features outperforms the features alone. The integration of context-based and content-based features provides a stronger set of features, which enhances the overall effectiveness of sarcasm detection.

ACKNOWLEDGEMENT

This research project was financially supported by Mahasarakham University.

AUTHOR CONTRIBUTIONS

Conceptualization, P. Namwong and P. Songram; methodology, P. Namwong; software, P. Namwong; validation, P. Namwong, P. Songram and K. Rukpukdee; formal analysis, P. Namwong; investigation, P. Namwong, P. Songram and K. Rukpukdee; data curation, P. Namwong; writing-original draft preparation, P. Namwong; writing-review and editing, P. Namwong, P. Songram and K. Rukpukdee; visualization, P. Namwong and P. Songram; supervision, P. Namwong and P. Songram. All authors have read and agreed to the published version of the manuscript.

References

[1] S. W. K. Chan and M. W. C. Chong, "Sentiment analysis in financial texts," *Decision Support Systems*, vol. 94, pp. 53-64, Feb. 2017.

[2] S. Tartir and I. Abdul-Nabi, "Semantic Sentiment Analysis in Arabic Social Media," *Journal of King Saud University - Computer and Information Sciences*, vol. 29, pp. 229-233, Apr. 2017.

[3] S. Gitto and P. Mancuso, "Improving airport services using sentiment analysis of the websites," *Tourism Management Perspectives*, vol. 22, pp. 132-136, Apr. 2017.

[4] G. Jasso and I. Meza, "Character and Word Baselines Systems for Irony Detection in Spanish Short Texts," *Procesamiento Del Lenguaje Natural*, no. 56, pp. 41-48, Mar. 2016.

[5] M. Bouazizi and T. Otsuki Ohtsuki, "A Pattern-Based Approach for Sarcasm Detection on Twitter," in *IEEE Access*, vol. 4, pp. 5477-5488, 2016.

[6] M. S. Razali, A. A. Halin, L. Ye, S. Doraisamy and N. M. Norowi, "Sarcasm Detection Using Deep Learning With Contextual Features," in *IEEE Access*, vol. 9, pp. 68609-68618, 2021.

[7] S. G. Wicana, T. Y. İbisoglu and U. Yavanoglu, "A Review on Sarcasm Detection from Machine-Learning Perspective," *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*, San Diego, CA, USA, pp. 469-476, 2017.

[8] O. o. t. R. Society, "Thai Royal Institute Dictionary," 18-feb-2021, 2021.

[9] Z. Karkiner and M. Sert, "Sarcasm Detection in News Headlines with Deep Learning," *2024 32nd Signal Processing and Communications Applications Conference (SIU)*, Mersin, Turkiye, pp. 1-4, 2024.

[10] N. A. Helal, A. Hassan, N. L. Badr and Y. M. Afify, "A contextual-based approach for sarcasm detection," *Scientific Reports*, vol. 14, no. 1, pp. 15415, 2024.

[11] O. Vitman, Y. Kostiuk, G. Sidorov and A. Gelbukh, "Sarcasm detection framework using context, emotion and sentiment features," *Expert Systems with Applications*, vol. 234, pp. 121068, 2023.

[12] K. Bari, "Sarcasm Detection of Newspaper Headlines Using LSTM-RNN," *2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India, pp. 604-608, 2023.

[13] P. Kumar and G. Sarin, "WELMSD – word embedding and language model based sarcasm detection," *Online Information Review*, vol. 46, no. 7, pp. 1242-1256, 2022.

[14] F. Kunneman, C. Liebrecht, M. van Mulken and A. van den Bosch, "Signaling sarcasm: From hyperbole to hashtag," *Information Processing &*

Management, vol. 51, no. 4, pp. 500-509, Jul. 2015.

[15] P. Vateekul and T. Koomsubha, "A study of sentiment analysis using deep learning techniques on Thai Twitter data," *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Khon Kaen, Thailand, pp. 1-6, 2016.

[16] M. Bouazizi, and T. O. Ohtsuki, "A Pattern-Based Approach for Sarcasm Detection on Twitter," in *IEEE Access*, vol. 4, pp. 5477-5488, 2016.

[17] A. Rajadesingan, R. Zafarani and H. Liu, "Sarcasm Detection on Twitter: A Behavioral Modeling Approach," *WSDM '15: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pp. 97–106, 2015.

[18] S. Hiai and K. Shimada, "A Sarcasm Extraction Method Based on Patterns of Evaluation Expressions," *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, Kumamoto, Japan, pp. 31-36, 2016.

[19] J. Karoui, F. B. Zitoune and V. Moriceau, "SOUKHRIA: Towards an Irony Detection System for Arabic in Social Media," *Procedia Computer Science*, vol. 117, no. Supplement C, pp. 161-168, Jan. 2017.

[20] N. Boudad, R. Faizi, R. Oulad Haj Thami and R. Chiheb, "Sentiment analysis in Arabic: A review of the literature," *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 2479-2490, 2018.

[21] A. N. Reganti, T. Maheshwari, U. Kumar, A. Das and R. Bajpai, "Modeling Satire in English Text for Automatic Detection," *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Barcelona, Spain, pp. 970-977, 2016.

[22] M. Mladenović, C. Krstev, J. Mitrović and R. Stanković, "Using Lexical Resources for Irony and Sarcasm Classification," *BCI '17: Proceedings of the 8th Balkan Conference in Informatics*, no. 13, pp. 1-8, 2017.

[23] M. P. L, "OSKut (Out-of-domain StacKed cut for Word Segmentation)," 2022.

[24] P. Limkonchotiwat, W. Phatthiyaphaibun, R. Sarwar, E. Chuangsuwanich and S. Nutanong, "Handling Cross- and Out-of-Domain Samples in Thai Word Segmentation," *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 1003–1016, 2021.

[25] R. Ahuja and S. C. Sharma, "Transformer-Based Word Embedding With CNN Model to Detect Sarcasm and Irony," *Arabian Journal for Science and Engineering*, vol. 47, pp. 9379-9392, 2022.

[26] K. Pasupa and T. Seneewong Na Ayutthaya, "Thai sentiment analysis with deep learning techniques: A comparative study based on word embedding, POS-tag, and sentic features," *Sustainable Cities and Society*, vol. 50, p. 101615, 2019.

Pramote Namwong is a Ph.D. student, major in computer science from the Faculty of Informatics, Mahasarakham University (MSU), Thailand. His areas of study interest include Machine Learning, Artificial Intelligence, Data Analytics, and Data Visualization. He can be contacted at email: 60011260505@msu.ac.th

Panida Songram is an Associate Professor at the Faculty of Informatics, Mahasarakham University (MSU), Thailand. Her research focuses on pattern mining, text classification, and data mining. She can be contacted at panida.s@msu.ac.th.

Kriangsak Rukpukdee is an Assistant Professor at the Faculty of Business Administration and Management, Ubon Ratchathani Rajabhat University (UBRU), Thailand. His areas of study interest include Machine Learning, Artificial Intelligence, Data Analytics, and Data Science. He can be contacted at email: kriangsak.r@ubru.ac.th