https://doi.org/10.1007/s42979-021-00775-6

ORIGINAL RESEARCH

Thai Fake News Detection Based on Information Retrieval, Natural Language Processing and Machine Learning

Phayung Meesad[1]

Received: 6 May 2021 / Accepted: 12 July 2021 / Published online: 23 August 2021

Abstract

Fake news is a big problem in every society. Fake news must be detected and its sharing should be stopped before it causes further damage to the country. Spotting fake news is challenging because of its dynamics. In this research, we propose a framework for robust Thai fake news detection. The framework comprises three main modules, including information retrieval, natural language processing, and machine learning. This research has two phases: the data collection phase and the machine learning model building phase. In the data collection phase, we obtained data from Thai online news websites using web-crawler information retrieval, and we analyzed the data using natural language processing techniques to extract good features from web data. For comparison, we selected some well-known classification Machine Learning models, including Naïve Bayesian, Logistic Regression, K-Nearest Neighbor, Multilayer Perceptron, Support Vector Machine, Decision Tree, Random Forest, Rule-Based Classifier, and Long Short-Term Memory. The comparison study on the test set showed that Long Short-Term Memory was the best model, and we deployed an automatic online fake news detection web application.

Keywords Fake news detection · Information retrieval · Natural language processing · Machine learning

Introduction

The evolution of information and communication technology has dramatically increased the number of Internet users. It transforms the way people consume information and news from traditional to digital, resulting in comfort and speed for both news presenters and newsreaders. In its convenience, the Internet system also generates a lot of fake news content. Fake news has become one of the major concerns as it can destabilize governments that endanger modern society [1]. For example, the electoral campaign in the USA in 2016 [2] had the term "fake news" found to gain much prominence due to the influence of fraudsters. The Internet is a big data source of online news. Not like in the past, news was published on paper. Now newspaper bureaus have moved to online platforms. The readers can easily access from any place at any time via the Internet. People are now comfortable accessing online news and can quickly share the news contents across the social network media such as WWW, Google, YouTube, Google+, Facebook, Twitter, Instagram, and Line [3, 4]. Fake news is a threat to democracy around the world, which has weakened the confidence of governments, newspapers, and civil society. The public's popularity on social media and social networks has led to the proliferation of fake news with conspiracy theories, distortions, and violent views. Detecting and mitigating the impact of fake news is one of the fundamental problems of modern times and is gaining widespread attention. While fact-checking websites such as Snopes, PolitiFact, and big companies like Google, Facebook, and Twitter, have taken some preliminary steps in dealing with fake news. Many communities include machine learning, databases, journalism, political science, and many others, pay attention to aspects of fake news as an interdisciplinary topic. There is still a lot to do to cope with the fake news issues [5].

Many researchers have proposed various machine learning approaches for fake news detection. Shu et al. [6] proposed a fake news detection framework exploiting social context called

This article is part of the topical collection "Advanced Machine Learning Approaches in Cognitive Computing" guest edited by Kuntpong Woraratpanya and Phayung Meesad.

✉ Phayung Meesad
pym@kmutnb.ac.th

[1] Department of Information Technology Management, Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand

a tri-relationship embedding framework TriFN. The model was based on publisher–news relations and user–news interactions simultaneously for fake news classification. They demonstrated that the proposed method significantly outperforms other existing fake news detection approaches. Yanagi et al. [7] proposed a neural network-based model for fake news detection with generated comments for news articles to help classification. Umer et al. [8] proposed a fake news stance detection using deep learning architecture based on convolutional neural networks and long short-term memory (CNN-LSTM). The method in [8] passed the non-reduced feature set with and without preprocessing to the neural network. The research used the principal component analysis (PCA) for dimensionality reduction, which increases the classifier performance because it removes the irrelevant, noisy, and redundant features from the feature vector. Akhter et al. [9] proposed an annotated corpus of Urdu news articles for the fake news detection tasks. The researchers used ensemble learning methods based on Naïve Baye, Decision Tree, and Support Vector Machine to improve the fake news detection system performance.

In Thailand, there are some Thai fake news detection research works based on machine learning. Aphiwongsophon and Chongstitvatana [10] studied Thai fake news from Twitter by employing three popular methods in the experiments: Naive Bayes (NB), Neural Network (NN), and Support Vector Machine (SVM). They found that the normalization process was mandatory for cleaning data before feeding the data to machine learning to classify data. The best model found in their collected data was SVM achieve an accuracy of 99.90%. Aphiwongsophon and Chongstitvatana [10] focused only on Twitter data; there was no implementation for online detection. Mookdarsanit and Mookdarsanit [11] proposed a deep learning framework for Thai COVID-19 fake news detection from the social text. Mookdarsanit and Mookdarsanit [11] built transferred learning models including Bidirectional Encoder Representations from Transformers (BERT), Universal Language Model FIne-Tuning (ULMFIT), and Generative Pre-trained Transformer (GPT). The researchers used COVID-19 news open datasets translated to Thai and pretraining Thai COVID-19 deep learning models. To fine-tune for a local dataset, the researchers used additional data by crawling Thai texts from social media and labeled them as fake and real samples. The best results from their experiments achieved the best accuracy performance of 72.93%. There was no report of real use cases for Thai fake news in the research.

Building fake news detection is a challenging task. It will be even more difficult for Thai fake news detection in the real situation, as the Thai language is one of the most complex languages with no space between words. In this research, we propose a framework to create an automatic online Thai fake news detection system. The proposed framework comprises three modules: information retrieval, natural language processing, and machine learning. Construction of the online fake news has three phases: data collection, data preparation, and machine learning modeling. The contributions of this research are as follows: (1) We propose a framework of online fake news detection as the main contribution. (2) In this research, a feature selection algorithm is also a result of natural language analysis. (3) To build Thai fake news detection, we collected a dataset and labeled them as fake–real-suspicious news. Lastly, (4) we develop an online Thai fake news web-based application, and it runs online at https://thaidimachine.org.

Literature Reviews

Fake News

In the digital age, more and more people use their daily lives to connect to the Internet and social networks. People are using the Internet on the rise with the convenience of delivering, accessing, and sharing news via the Internet and social networks, which makes it easy to spread information without any restrictions while posting it on these platforms. However, the information that is published may contain both real news and fake news. Some malicious users take advantage of these platforms by generating fake news, spreading them on the Internet and social media networks to damage the reputation of individuals, businesses, and politics [12, 13].

Misinformation can appear in different formats and domains, such as fake news, click baits, and false rumors, and much of the previous research has focused on modeling specific to a single domain [14, 15]. These domains may have different formats, such as long articles versus short headlines and tweets, and their exact purpose like “This is a fake” vs. “Click Bait”; however, they have the same goal of deceiving the readers. As a result, content that exhibits similar linguistic features, such as the use of exciting themes to arouse curiosity or intense emotional responses from readers [14]. Therefore, many researchers proposed a way to detect fake news to stop the distribution of fake news. Online news is dynamics during propagation on social media. Malicious users can diverge from the original and create fake news. It makes detecting fake news automatically from the Internet a challenging task in detecting fraud [16, 17].

Creating automatic fake news or misinformation detection involves many theories and practices. The main disciplines may include information retrieval, natural language processing, and machine learning.

Information Retrieval

Databases store the information in a structured manner in many documents. When searching documents, it is a problem to find information needed, such as search terms or sample documents. An information retrieval system (IR) is a software system that provides an access to documents to manage and store them. An IR system is a branch developed in conjunction with a database system. IR can be considered as the science of searching for information in documents, manual document searching, and descriptive metadata search, and for databases of text, images, or sound [18, 19]. It is the activity of obtaining information from information system resources relevant to the information needed. A query can be full-text or other content indexing. An automated IR system can reduce data overload. For basic concepts in IR, documents can be explained by a set of terms representing a document is called index terms. Different index terms are relevant when used to describe the content of a document. This effect is assigned a numerical weight to each document index, such as term frequency and inverse document frequency (TF × IDF).

IR models have three types: Boolean Model, Vector Model, Probability Model. The Boolean model is an exact match between the index terminology and the search terms. Boolean information retrieval predicts each document whether it is relevant or not relevant to the document query [20]. For a vector information retrieval model, vocabulary, or word (term) is used instead of attributes. The searched document comprises words converted to numbers called term frequency or weight values. The weight values are a substitute for document queries. With the weight values, distance formula or similarity measure calculates the relationship between the query against the document in the database. The vector information retrieval emphasizes the frequency of the words contained in the document and the effect on the weighting of the term against the word count of the document word weight [21, 22]. The third model, the probability information retrieval model is based on a user query. The probability information retrieval model sequences the documents according to the probability based on their relationship or relevance to the query text, where high probability means high relevance. The accepted probability calculation method is calculated from the word frequency data [23].

Natural Language Processing

Natural Language Processing (NLP) [24] is a sub-branch of linguistics, computer science, data engineering, and artificial intelligence. NLP relates to the interaction between humans and computers. NLP is a method for processing and analyzing large amounts of natural language data. NLP has many applications such as machine translation, speech recognition, sentiment analysis, automatic question and answer generation, automatic message digest, chatbot, intelligence, text classification.

In the NLP, one crucial step is text extraction, a preprocessing step for using the analysis of text, documents, news, and information before implementing the clustering, classification, or other machine learning tasks [25]. The fundamental preprocessing step for NLP includes word segmentation, tokenization, word stopping, word stemming, term frequency weighting, term frequency, and inverse document frequency weighting [26]. Some advanced NLP techniques may include more complex tasks in the pipeline, such as parts of speech tagging, dependency parsing, named entity recognition, and conference resolution [27]. Advanced NLP techniques may employ lexical analysis, syntactic analysis, semantic analysis, disclosure integration, and pragmatic analysis [28].

The Thai language is a language that has words in continuous place without space in consecutive sentences in the documents. For analysis, the Thai documents need to break down into a single word like English. Word segmentation is the separation of each word from sentences, which still has correct meaning by using a dictionary database of words. There are many techniques for word segmentation including Longest Word Pattern Matching, Shortest Word Pattern Matching, Word Wrapping, Probabilistic Word Segmentation, Back Tracking, Feature-based, and Machine Learning-based techniques [26, 29, 30].

Machine Learning

Machine learning is a study field in computer science, which involves creating adaptive programs that can learn via training data. There are many forms of machine learning, including supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Normally, building a machine learning model starts with data preparation for two sets: training data and test data. Machine learning learns from training data. The users evaluate the trained machine learning model using the test data. The evaluation by the test data is to make sure that we can use the trained model to predict the future unseen data with confidence. Training the machine learning model is a search for optimal parameters. The users seek the most suitable machine learning model parameters. The users choose a machine learning model for a proper task as different techniques will suit different tasks.

In this paper, we focus on supervised learning that include Logistic Regression (LR), K-Nearest Neighbor (KNN), Rule-Based Classifier (RBC), Decision Tree (DT), Random Forest (RF), Naïve Bayesian (NB), Multilayer Perceptron (MLP), Support Vector Machine (SVM), and Long Short-Term Memory (LSTM).

Logistic Regression (LR) is a mathematical modeling-based machine learning used to describe the relationship of several independent variables to a dependent dichotomous (binary) variable. We can train LR as classifier as it is a suitable regression analysis for the dependent dichotomous variable. Logistic regression can describe the relationship between one or more dependent binary variables and independent variables that specify at least one sequence, range, or ratio level [31–33].

K-Nearest Neighbors (KNN) is a simple non-parameter classification method. The KNN is a case-based learning method that maintains all training data for the classification task. To use KNN, we need to choose a suitable *K* value, and the classification result depends on this value. There are many ways to select a *K* value, but the easy way is to run the algorithm multiple times with different *K* values and choose the most effective one. To classify data using a KNN classifier, we need three things: stored training data, *K* value, distance, or similarity metric. The KNN performs as follows. (1) read in a data record to classify. (2) compute the distance between the classifying data record to all stored training data. (3) select the *K* smallest distance. And (4) classify the classifying data based on the majority vote from the *K* nearest data records' labels [34].

Rule-Based Classifiers (RBC) comprise a rule set in the form IF *X*, then *Y*. Using classification training dataset, we can train a rule-based system to become an RBC. RBC needs a rule-based algorithm to generate a rule set as a classification scheme defined as a set of IF-THEN rules. We then can use the ruleset to classify each instance in the dataset. CN2 is one of the most widely used as a rule induction algorithm. CN2 Rule Induction is a rule-based algorithm of the rule-based classifiers. CN2 uses the heuristic function, such as Entropy, Laplace, and Accuracy, to terminate the search during rule formation based on the noise approximation present in the data. The specified rules may not correctly classify all training samples. However, it works fine with the new unseen data. The CN2 accepts only rules of exceptional precision so it can deal with noise. Besides, CN2 can create a sorted or unordered rule list [35, 36].

Decision Tree Classifier (DTC) is a tree-like model represented as a recursive partition of the data space. A decision tree consists of a most discriminant node that forms a rooted tree. At the top, a tree starts with a root node that does not have an incoming branch or link, or edge. All other nodes have only one incoming edge. Nodes with outgoing edges are called intermediate or test nodes. At the lowest levels, nodes are called leaves, which are decision nodes. There are many decision tree induction algorithms; some famous algorithms are ID3, C4.5, and C5 [37–40].

Random Forest (RF) is one of the best algorithms for classification tasks. The basic idea behind RF is that a group of weak learners can form a strong learner. RF can classify large datasets with high accuracy and precision. RF acts as a classifier with every tree dependent on a random vector value. RF generates many decision trees at the time of training, and the outcome of the modalities predicted by each tree created using bootstrap samples of training data and random selection of attributes in tree induction. Prediction is formed by combining using majority vote or averaging all decision trees [41–44].

Naïve Bayesian (NB) is an easy learning probabilistic-based algorithm that uses Bayes' rule in conjunction with the explicit assumption that attributes are conditionally independent of each other. Based on training data, NB estimates the posterior probability $P(y|x)$ of each class, *y*, of a given object, *x*. We can use the estimation for classification applications. Because of its computational efficiency and many other desirable properties, NB appears as an acceptable solution in many practical implementations [45–48].

Multilayer Perceptron (MLP) is one type of artificial neural network (ANN) based on simulating the function of the human brain using a computer program. The goal of ANN is to make computers as intelligent as humans are. ANN can learn from training data and recall back knowledge to apply to the specific trained problems such as Classification, Regression, and Clustering. MLP is often referred to as a "black box" because of its functionality. MLP, sometimes called a feedforward network, has several calculation steps. It starts with the input data entered at the input layer having synaptic weight linked to neurons in the hidden layers. MLP may have several hidden layers depending on the complexity of data. Each hidden layer has synaptic weights connected to the next layer. The outputs from the previous layer act as the input to the next layer. The signal reaches the output layer, where the prediction output goes out from the neural network [41, 49–51].

Support Vector Machine (SVM) is based on the learning of statistical theory. Several researchers applied SVM to many applications in data classification or pattern recognition. SVM theoretical concepts are as follows. (1) Structural Risk Minimization is a concept that expresses the extent of the risk or the likelihood of learning errors. The SVM learning process determines the function of decision-making to minimize the error rate. The kernel function is an important concept that supports a vector machine technique. A kernel function maps data from input space to feature space to create non-linear decision-making functions to data in the leading space. (2) Optimal margin hyperplane is a crucial concept of vector machine support techniques. The learning process of SVM is to find the plane with the maximum margin, in which it can separate the data into two groups apart and solve the problem of overfitting [52–54].

Long Short-Term Memory (LSTM) is a deep learning model in a recurrent Neural Network (RNN) group. RNN provides hidden state feedback as input that makes it possible to capture the dependency of sequence data such as time series and natural languages. RNN is not only to process a single data point but also to process sequential data. LSTM, developed to solve the problem of exploding and vanishing the slope error faced in traditional RNN, is well suited for classification, processing, and prediction based on time series data as there may be an unknown period between events in time series. One can use LSTM for many tasks such as sentiment analysis from documents, handwriting recognition, speech recognition, and anomaly detection of network data [55–57].

The Proposed Fake News Detection Framework

Fake news or misinformation contents are overly broad and very dynamics making it hard to build machine learning as a fake news detection system. However, it is not impossible to build a robust news classifier. As news producers distribute and publish news online, people can access it quickly via the Internet from anywhere. The www and social media keep both real and fake news on the cloud servers. People pose a comments discussion on the news website and share them on social media.

We propose a fake news detection framework based on three main modules: Information Retrieval (IR), Natural Language Processing (NLP), and Machine Learning (ML). The IR module retrieves news information from the Internet according to the news query fed by the user. The results from the search content are the relevant news contents from many news online sources. Next, the NLP module analyzes the retrieved documents by performing segmentation, cleansing, and feature extraction. Finally, the ML module classifies the news into three known classes are Real, Suspicious, and Fake. Figure 1 illustrates the proposed fake news detection framework.

The implementation of fake news detection comprises two phases: (1) news collection and training and (2) machine learning prediction. Each involves IR, NLP, and ML modules. In the first phase, web crawlers in parallel collect data from www and social media and preprocessed them to train machine learning as a fake news detection model. In the data collection and training phase, the IR module crawls the web to retrieve the news data from news websites and use them as domain corpus used in the NLP module. The user query is the entry point to get the training data. For each news query, the system sends web crawlers to fetch and retrieve a related news list. The relevant news list is processed to get featured data for training the machine learning model. Each news query will act as a user query. It means that the web crawler will fetch the web to retrieve a relevant news list corresponding to the news query. NLP will process the retrieved news list and return featured data. The NLP module receives the news content and performs text segmentation, cleansing, and feature extraction. Finally, the resulted feature data flow to build the machine learning. We used the featured data for analysis, labeling them as fake, real, or suspicious.

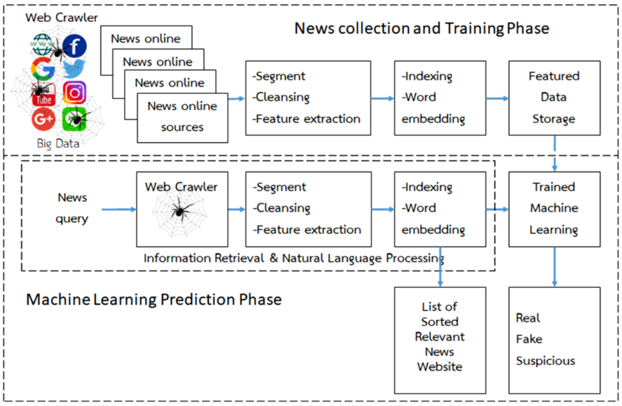


Fig. 1 Fake news detection framework

The data collection must be large enough to be used to train machine learning models. We separate the feature data into two sets: a training set and a test set. We use the training set to train and cross-validate the machine learning model during the training period, while the test set evaluates the resulting trained model. The training phase stops when the best model is ready for further deployment in the machine prediction phase. The machine learning prediction phase is the online fake news prediction deployment. It is a web application development of machine learning-based fake news prediction. In this phase, the IR module receives a news query from a user then the web crawlers crawl the web and social media to retrieve the relevant news contents with cosine similarity to the news query. NLP receives the news contents and analyzes them by doing word segmentation, cleansing, feature extraction. The trained machine learning model gets the featured data and classifies them into three classes: Real, Fake, or Suspicious. Also, the retrieved news contents list according to the similarity to the query.

Information Retrieval

Our challenging task is to create a fake news detection model. Then we can use it to detect fake news online to warn people not to share or distribute fake news or misinformation to others. Information retrieval plays a crucial role in the framework. It is a key to access news content on the www and social media on the Internet. Not only the Internet store real news, but it also stores fake news. Web crawler-based Information retrieval is a better way to collect the news content on online news websites.

The proposed web crawler-based information retrieval module comprises the news collection process and the feature extraction process. In the news collection process, web crawlers are web robots or agents sent out to collect information from news sources on the web considering big data. The web crawlers retrieve web data and send them to the

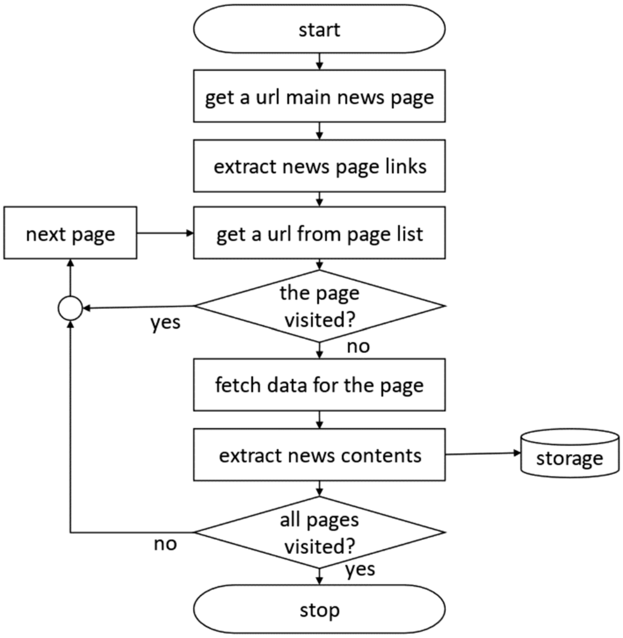


Fig. 2 Web crawler-based Information Retrieval

data preparation process. The news list is the result of web crawler-based information retrieval. Each news list related to a query consider high similarity in the contents. Figure 2 shows flowchart of web crawler-based information retrieval.

From Fig. 2, the web crawler-based information retrieval sends a web crawler that is a search robot out to www and social media. It starts with an entry to the main page, where the robot can extract links to news pages. Each link is a URL linked to a news page a target the robot needs to retrieve the contents. The robot check if the page is not visited, it fetches the page and extracts the news contents. The robot stores the resulted from the news page into database storage for further analysis. If all the pages in the list are visited, the robot stops. The web crawler robots continuously collect the data daily to get up-to-date news content. The natural language module in the data preparation process analyzes the retrieved news contents.

Natural Language Processing

Natural language processing (NLP) operates crucial tasks in the data preparation process. NLP performs word segmentation, data cleansing, word stopping, feature extraction, word indexing, and word embedding. The data preparation process produces clean texts of news contents before being sent to a feature extraction process for converting to featured data vectors and stored in a featured database. The feature extraction subsequently receives data from the information retrieval process, which gets a query from a user one at a time sent to the web crawlers to seek the relevant news contents from the Internet. The query process sends the retrieved news contents to the data preparation process and transforms them into featured data vectors. The feature extraction module also performs a similarity measure of retrieved news contents with the news storage and sorts the news contents according to their distance to the query. Later the sorted news list can be displayed as related news to the user on a website. The featured data will be used for traditional machine learning models, while LSTM deep learning uses a sequence of text input. Figure 3 shows the natural language processing framework proposed in this research.

Machine Learning Modeling

Machine learning (ML) is an engine used to a type of news into one of three groups: fake, real, or suspicious. ML receives feature data from the NLP module that processes text data into document vectors. Machine learning models in this work comprise both traditional classifier and modern deep learning models. The conventional models include Logistic Regression (LR), K-Nearest Neighbor (KNN), Rule-Based Classifier (RBC), Decision Tree (DT), Random Forest (RF), Naïve Bayesian (NB), Multilayer Perceptron (MLP), and Support Vector Machine (SVM). The modern deep learning model is Long Short-Term Memory (LSTM). Deep learning model can directly connect with text data as it can automatically extract features during the training period. Figure 4 shows fake news detection with machine learning framework.

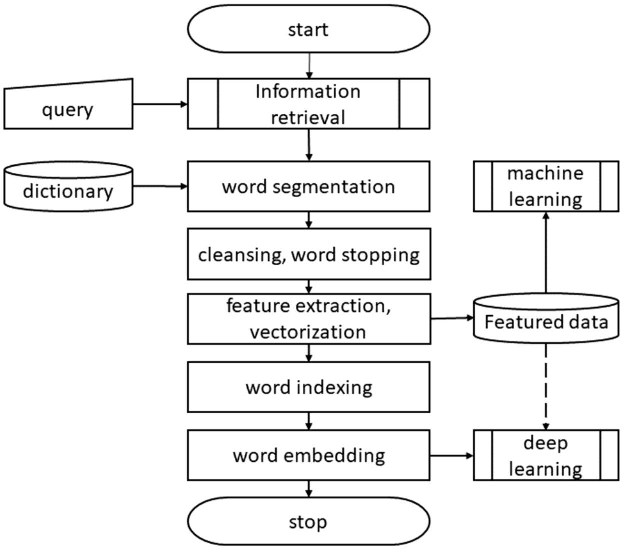


Fig. 3 Natural language processing framework

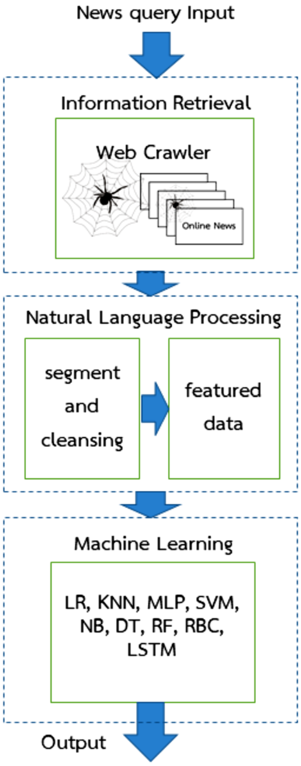


Fig. 4 Fake news detection with machine learning framework

Mathematical Models and Algorithms

For the information retrieval process in this research, we send out a web crawler as a search engine to retrieve relevant news corresponding to the news query. Mathematical models are defined. Let us define a news vector document as in (1):

$$\mathbf{n}_i = [n_{j,i}] = [n_{1,i}, n_{2,i}, \dots, n_{N,i}], \tag{1}$$

where $n_{j,i}$ is a word or term jth in the ith news document $\mathbf{n}_i$, and N is the number of words in the ith news document. N is varied for each news document.

Let us define a news query document as in (2):

$$\mathbf{q} = [q_j] = [q_1, q_2, \dots, q_Q], \tag{2}$$

where q_j is a word or term jth in the query, and Q is the number of words in the query. Q is varied for each query.

In measuring the similarity of two vectors, the two vectors must have the same size. Since Q and N are varied, we need to make them the same size.

Let us define a truncated cosine similarity modified from cosine similarity [58] as in (3) and (4):

$$\mathrm{tcs}(\mathbf{q}, \mathbf{n}_i) = \frac{\mathbf{q} \cdot \mathbf{n}_i}{||\mathbf{q}|| \times ||\mathbf{n}_i||} = \frac{\sum_{j=1}^{L} q_j n_{j,i}}{\sqrt{\sum_{j=1}^{L} q_j^2} \cdot \sqrt{\sum_{j=1}^{L} n_{j,i}^2}} \tag{3}$$

$$L = \min(Q, N), \tag{4}$$

where $\mathrm{tcs}(\mathbf{q}, \mathbf{n}_i)$ is the truncated cosine similarity between the query $\mathbf{q}$ and the ith news document $\mathbf{n}_i$.

Let us define $\mathbf{S}$ represents a list of segmented news texts, as in (5):

$$\mathbf{S} = [S_1, S_2, \dots, S_P], \tag{5}$$

where $S_p, p = 1, 2, \dots, P$, defined in (6), is a concatenated text sequence retrieved from a preprocess of relevant news texts corresponding to a news query, $\mathbf{q}_p$. P is the number of queries.

$$S_p = [w_1, w_2, \dots, w_{L_p}], \tag{6}$$

where $w_i, i = 1, 2, \dots, L_p$, is a word in the text sequence. L_p is the dimension or length of the pth text sequence.

Let us define a data matrix $\mathbf{D}$ as in (7) and a target matrix $\mathbf{T}$ as in (8):

$$\mathbf{D} = [\mathbf{d}_p] = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_P], \tag{7}$$

$$\mathbf{T} = [t_1, t_2, \dots, t_P], \tag{8}$$

where $\mathbf{d}_p \in \Re^5, p = 1, 2, \dots, P$, is a feature data vector and t_p is a target label of $\mathbf{d}_p$. Each $t_p \in [C_1, C_2, \dots, C_G]$, G is the number of classes.

Let us define fs, as in (9), representing fake news score, rs, as in (10), representing real news score, sm, as in (11), representing similarity matching score, lf, as in (12) and (13), representing the length of domains of fake news, and lr, as in (14) and (15), representing the length of domains of real news.

$$fs = \mathrm{count}(\mathrm{NT}), \tag{9}$$

where NT the predefined negative terms in $\mathbf{n}_i$ that are the words that usually appear in fake news.

$$rs = \mathrm{count}(\mathrm{PT}), \tag{10}$$

where PT is the predefined positive terms in $\mathbf{n}_i$ that are the words that usually appear in real news.

$$sm = \mathrm{sum}(tcs(\mathbf{q}_j, \mathbf{n}_i) | tcs \geq \alpha), \tag{11}$$

where α represents a user defined threshold similarity between a query to the retrieved news document; it is a scalar value in range [0.0, ..., 1.0], e.g., set $\alpha = 0.3$.

$$lf = \mathrm{count}(\mathrm{DF}), \tag{12}$$

where DF is the domain fake news that are the web domains in which the fake news appears. Alternatively, lf can be calculated from

$$lf = \mathrm{length}(\mathrm{DF}), \tag{13}$$

where length is the length of domains containing fake news.

$$lr = \text{count}(\text{DR}), \tag{14}$$

where DR is the domain real news that are the web domains in which the real news appears. Alternatively, lr can be obtained from

$$lr = \text{length}(\text{DR}), \tag{15}$$

where length is the length of domains containing real news.

Relations among (1) thru (15) can be explained as follows. Equations (1) and (2) represent news document vectors and a news query document vector, respectively. Equation (3) is a proximity formula based on cosine similarity to measure the similarity among vectors, while (4) is used in (3). Equation (3) determines the relevant news document vectors returned by the web crawlers. Equations (5) and (6) represent the text sequences obtained from retrieved relevant news processed by (1)–(4). Equations (7) and (8) represent numerical features and corresponding labels, respectively. Equation (7) stores featured data extracted from (9) to (15). We can use (12) and (13) alternatively. Equations (14) and (15) also can be used interchangeably.

For news data preprocessing, we introduce three algorithms: Web Crawler-based Information Retrieval, NLP-based Feature Extraction, and Clustering based News Labeling. The three algorithms are shown in Algorithm 1, Algorithm 2, and Algorithm 3.

Algorithm 1, Crawler-based Information Retrieval, receives a news query from a user. The system sends crawlers to visit websites and retrieves the relevant news with metadata. This algorithm calculates the similarity between the user news query with the retrieved and returns the news metadata with similarity scores. Algorithm 1 is used by Algorithm 2.

Algorithm 2, NLP-based Feature Extraction, processes the news documents and transforms them into featured data used by traditional machine models. The algorithm loops to a list of queries and calls Algorithm 1 to obtain relevant news from news sources. Each new query then has a list of the relevant news set. Next, Algorithm 2 processes each relevant news content by performing word segmentation, word stopping, word cleansing, fake news score calculation, real new score calculation, similarity matching calculation, length of domain fake news calculation, length of domain real news calculation, and store them in featured data vector. This algorithm also performs text concatenation where the high similarity among the news query and the retrieved news are concatenate after word segmentation and cleansing processes. The algorithm returns the feature data vector as well as the cleaned text sequences. Algorithm 3, Clustering-based News Labeling, is for a labeling process. The algorithm receives featured data and clusters them into three segments. Given initial centroids as three randomly chosen from the training data, the algorithm takes one record at a time and calculates the Euclidean distance with the centroids. The algorithm finds the winner, which is the closest centroid to the checking datum. The algorithm assigns a label to the checking datum as a member of the winner centroid. The label is the same as the winner centroid. Next, the algorithm continues receiving the next data point and repeats the process until all data are visited. The algorithm updates the centroids according to the member in each cluster. If the centroids do not converge, it loops again for the next iteration until the algorithm converges to stable centroids.

Algorithm 1 Crawler-based Information Retrieval

```
Input: news query $\mathbf{q}$ and $M$ is the number of website
Output: retrieved news $\mathbf{n}_i$ with similarity values to the new query $\mathbf{q}$
    function INFORMATION RETRIEVAL($\mathbf{q}, M$)
        for $i$ from 1 to $M$ do:
            send crawlers to visit web$_i$
            retrieve $\mathbf{n}_i$ from the web$_i$
            calculate $tcs(\mathbf{q}, \mathbf{n}_i)$
    return $\mathbf{n}_i$
```

Algorithm 2 NLP based Feature Extraction

```
Input: the news document $\mathbf{q}$ and $P$ is the number of queries
Output: the feature data $\mathbf{D} = [\mathbf{d}_p] = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_P]$ and text sequence $\mathbf{S} = [S_p] = [S_1, S_2, ..., S_P]$
    function NATURAL LANGUAGE PROCESSING($\mathbf{q}, M$)
        for $p$ from 1 to $P$ do:
            call the information retrieval
            retrieve news document set $\mathbf{n}$
            according to the query $q_p$
            for each $\mathbf{n}_i, i$ from 1 to $M$ do:
                word segmentation
                word stopping
                word cleansing
                calculate fake news score,
                $fs = count(predefined\ negative\ terms\ in\ \mathbf{n}_i)$
                calculate real news score
                $rs = count(predefined\ positive\ terms\ in\ \mathbf{n}_i)$
                calculate similarity matching
                $sm = sum(tcs(\mathbf{q}_p, \mathbf{n}_i)|tcs \geq \alpha)$
                calculate length of domain fake news
                $lf = count(domain\ fake\ news)$
                calculate length of domain real news
                $lr = count(domain\ real\ news)$
                $S_p = concatenate(\mathbf{n}_i|tcs \geq \alpha)$
            append featured data $\mathbf{D} \leftarrow \mathbf{d}_p = [fs, rs, sm, lf, lr]$
            append text sequence $\mathbf{S} \leftarrow S_p = [w_1, w_2, ..., w_{L_p}]$
    return $\mathbf{D}$ and $\mathbf{S}$
```

Algorithm 3 Clustering based News Labeling

```
Input: feature data $\mathbf{D}$
Output: feature data with label $\mathbf{D}, \mathbf{T}$
    function NEWS LABELING($\mathbf{D}$)
        set the number of clusters = 3
        set initial centroids $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3]$
        by selecting 3 records randomly from $\mathbf{D}$
        while not converge do:
            for $\mathbf{d}_i$ in $\mathbf{D}$ do:
                for $\mathbf{c}_j$ in $\mathbf{C}$ do:
                    compute Euclidean distance
                    $dist(\mathbf{d}_i, \mathbf{c}_j) = \sqrt{\sum_{k=1}^{L}(d_{i,k} - c_{j,k})^2}$
                find the $\mathbf{c}_{winner}$
                $winner = argmin_j dist(\mathbf{d}_i, \mathbf{c}_j))|j = 1, 2, 3$
                set winner as a label $\mathbf{T}_i$
                assign $\mathbf{d}_i$ as a member of $\mathbf{c}_{winner}$
            update centroids
            average value of the members of each centroid
            if the centroids do not change then:
                converge=True.
    return $\mathbf{D}, \mathbf{T}$
```

Machine Learning Modeling and Evaluation

We used machine learning as an engine to classify fake news. There are many machine learning models for classification. But what the most suitable model for Thai fake new detection based on feature data extracted in this research was our concern. To select the best model, we need metrics to measure the efficiency of the models. A confusion matrix is defined as in Table 1, where $T_{i,i}, i = 1, 2, ..., G$, is the number of correct predictions, and $F_{i,j}, i = 1, 2, ..., G, j = 1, 2, ..., G, i \neq j$, is the number of false predictions, class i is incorrectly classified as class j. Based on confusion matrix, ones can compute performance metrics of classifiers. The popular metrics for classification models include accuracy, precision, recall, and F-measure (or f_1 − score). For multi-class classification problems, the formula of the metrics are as in (16)–(22).

$$\text{accuracy} = \frac{\sum_{i=1}^{G} T_{i,i}}{|\mathbf{D}|}, \tag{16}$$

where accuracy is the overall accuracy of the classifier, $|\mathbf{D}|$ is the number of test data.

$$pr_i = \frac{T_{i,i}}{\sum \text{column}_i} \tag{17}$$

$$\text{precision} = \frac{\sum_{i=1}^{G} \text{pr}_i}{G}, \tag{18}$$

where $T_{i,i}, i = 1, 2, ..., G$, is the number of correct predictions, $\sum \text{column}_j$, $j = 1, 2, ..., G$, represents the number of machine learning predicted class; pr_i is the precision of class i; precision is the overall precision.

$$\text{re}_i = \frac{T_{i,i}}{\sum \text{row}_i} \tag{19}$$

$$\text{recall} = \frac{\sum_{i=1}^{G} \text{re}_i}{G}, \tag{20}$$

where $T_{i,i}, i = 1, 2, ..., G$, is the number of correct predictions, $\sum \text{row}_i$, $i = 1, 2, ..., G$, represents the number of true class, re_i is the recall of class i; recall is the overall recall.

$$f_i = 2\frac{\text{pr}_i \times re_i}{\text{pr}_i + \text{re}_i} \tag{21}$$

$$F - \text{Measure} = \frac{\sum_{i=1}^{G} f_i}{G}, \tag{22}$$

where f_i is the F-Measure of class i; F-Measure is the overall recall taking weighted average from all classes.

The Experimental Results

Data Collection and Data Preparation

To build a fake news detection system, we need data to train machine learning to classify the news. We need data as much as possible to cover the news domain. In doing so, we collected data using web robot crawlers. The detail of the web crawler-based information retrieval process as in Algorithm 1.

It is a challenging task for Thai language processing because the Thai language has no space between words. We applied PyThaiNLP [30] as a tool for Thai word segmentation. We used the maximum matching method for Thai word segmentation and a custom dictionary with the size of vocabularies of 75,936 words used in this study. Figure 3 shows the flowchart of the natural language processing framework.

Table 1 Confusion matrix

	Predicted classes				
Actual classes	C_1	C_2	C_3	…	C_G
C_1	$T_{1,1}$	$F_{1,2}$	$F_{1,3}$	…	$F_{1,G}$
C_2	$F_{2,1}$	$T_{2,2}$	$F_{2,3}$	…	$F_{2,G}$
C_3	$F_{3,1}$	$F_{3,2}$	$T_{3,3}$	…	$F_{3,G}$
…	…	…	…	…	.
C_G	$F_{G,1}$	$F_{G,2}$	$F_{G,3}$	…	$T_{G,G}$

For feature extraction, Algorithm 2, we use words that usually appear on fake or real news. The feature extraction takes acts when the web crawlers retrieve the news contents relevant to the query based on the truncated cosine similarity defined in (3). We used (7)–(13) for feature extraction from each news document. The data extracted are expected to have discriminatory characteristics of fake and real news. Please be informed that the context of fake and real news may be different in each country. The predefined negative words and positive words may vary in other countries. Here the positive and negative words are for Thai news.

The sample negative words, translated from Thai, are as follows: [ambiguous facts, ancient stories, artificial news, bad information, bad news, baseless, brag, but did you know, cannot cure disease, cannot do it, casual, catch pontoon, claims, cut paste, deceitful, deception, defamation, distorted messages, do not believe, do not share, does not exist, don't become victims, editing, fake, fake events, fake information, fake messages, fake news, fake news messages, fake news stories, fake stories, false, false beliefs, false facts, false information, false news, false reports, false statement, false stories, falsely, fraud, fraudulent web, garbage, incorrect facts, insecure facts, insecure information, insecure news, insufficient data, invalid information, is not true, lie, madden news, make a story, misinformation, misleading information, misrepresentation, mistakes of information, misunderstanding, negative news, no indication, no information, not qualified, not real, not real information, not real news, not true, not trustworthy, prank, propaganda, rumor, scam, slang, slogans, suspicious information, uncertain facts, uncertain information, uncertain news, unclear information, uncoordinated data, unreliable facts, unreliable information, unreliable news, untrue facts, valuable information, worthless facts, worthless news, wrong news, wrong ways].

The sample positive words, translated from Thai, are shown as the following list: [authenticity, confirmed to be true news, no distortion, no fake, no false, no false news, non fake news, non-fake news, non-false news, not fake news, not false, real data, real information, real message, real news, shareable, true, true message, true news, true story, verified news].

Using Algorithm 1, Algorithm 2, and Algorithm 3, we collected data for 41,448 samples with three groups: real, fake, and suspicious. Each group has 13,816 records equally distributed. We separated data into three sets: training set, validation set, and test set. The number of training data was 20,310. The number of validation data was 8,704, and the number of test data was 12,435. Table 2 shows sample data collected to build machine learning for fake news detection. Table 3 shows training, validation, and test sets.

Table 2 Sample data collected

Labels	No. samples
Fake	13,816
Real	13,816
Suspicious	13,816
Total	41,448

Table 3 Training, validation, and test sets

Data sets	No. samples	Ratio
Training	20,723	0.50
Validation	8290	0.20
Test	12,435	0.30
Total	41,448	1.00

Extracted Feature Data

NLP analyzes the retrieved data from web crawling. Word segmentation separates text into word tokens. The cleansing process further cleans segmented tokens by removing unnecessary words and characters. The feature extraction process extracts import characteristics from the news content.The extracted features in this research comprise five characteristics: score fake, score real, sim matched, domain fake, and domain real. score fake represents the count of negative words and fake group words that appear on the retrieved news contents. Also, score real is the count of positive or authentic group words found on the retrieved news contents. The sim matched feature is the accumulative cosine similarity between news query and the retrieved news contents. Besides, domain fake and domain real represent the number of websites or the length of domain websites that have fake news and real news, respectively. Table 4 shows feature data correlation.

From Table 4, the featured data include score fake, score real, sim matched, domain fake, and domain real. The targeted classes comprise fake, real, and suspicious. It is worth noting that the extracted features correlate with the targets. The sim matched shows a positive correlation to fake class as well as real class. score fake and domain fake features have 0.7 and 0.76 having predictive influence with class fake. Besides, score real and domain real features have a positive correlation of 0.16 and 0.11 with the class real. It implies that the data can represent fake and real classes quite well. However, class suspicious has a negative correlation with features. It would be difficult to differentiate the suspicious group.

Figure 5 shows the scatter joint plot of clustered feature data. The data clustering shows that it is possible to build a classifier to differentiate the three classes. Class fake

Table 4 Feature data correlation

	Score fake	Score real	Sim matched	Domain fake	Domain real	Fake	Real	Suspicious
Score fake	1.00	0.07	0.35	0.91	0.15	0.70	– 0.43	– 0.35
Score real	0.07	1.00	0.16	0.10	0.86	0.04	0.16	– 0.22
Sim matched	0.35	0.16	1.00	0.37	0.22	0.27	0.30	– 0.65
Domain fake	0.91	0.10	0.37	1.00	0.20	0.76	– 0.47	– 0.38
Domain real	0.15	0.86	0.22	0.20	1.00	0.11	0.11	– 0.25
Fake	0.70	0.04	0.27	0.76	0.11	1.00	– 0.62	– 0.49
Real	– 0.43	0.16	0.30	– 0.47	0.11	– 0.62	1.00	– 0.38
Suspicious	– 0.35	– 0.22	– 0.65	– 0.38	– 0.25	– 0.49	– 0.38	1.00

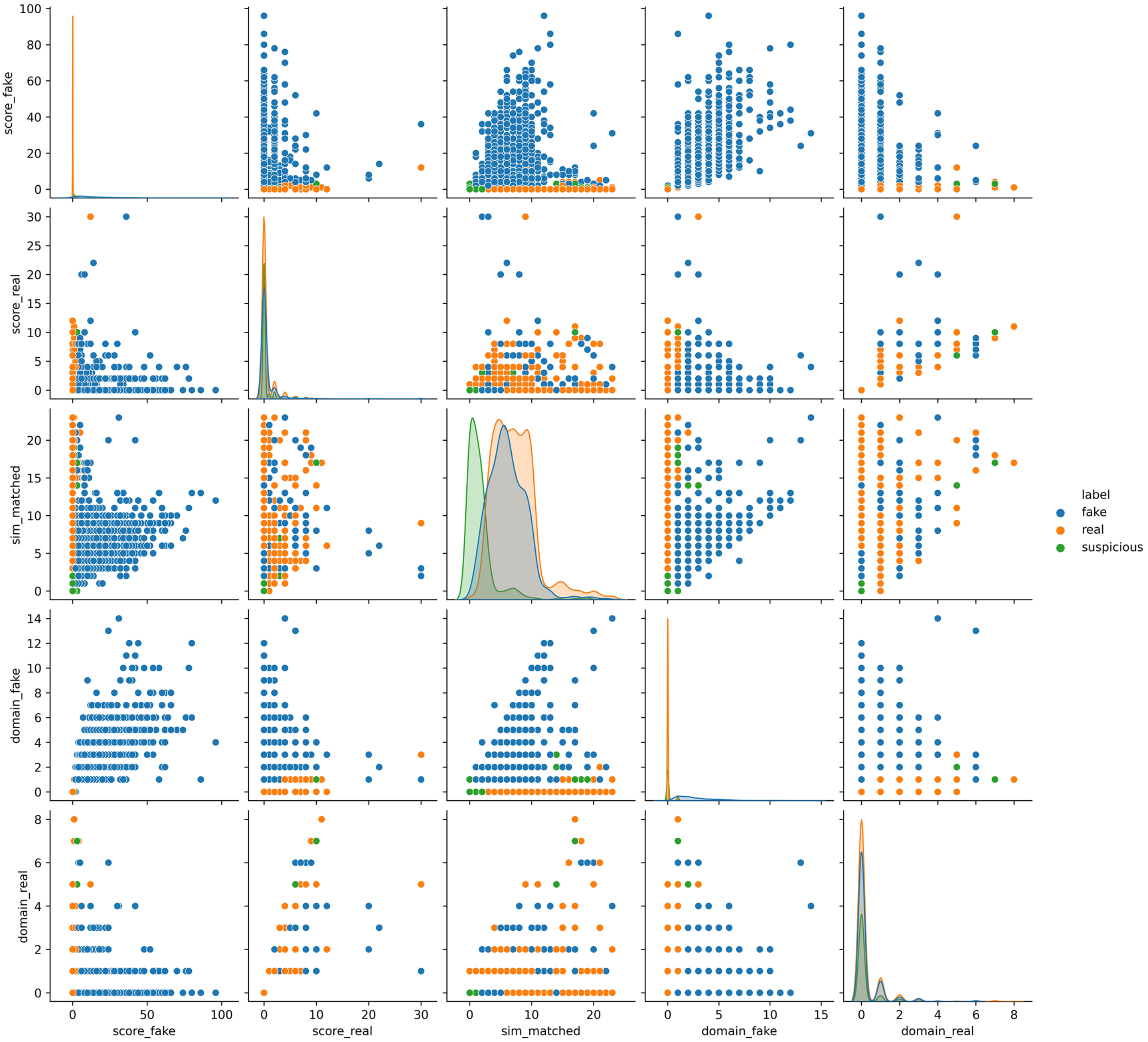


Fig. 5 The scatter joint plot of clustered feature data

Table 5 The details of LSTM model settings

Layer type	Activation	Output shape	No. Params
Text vectorization		(None, None)	0
Embedding		(None, None, 128)	589,568
Bidirectional LSTM		(None, 256)	263,168
Dense	Relu	(None, 64)	16,448
Dense	Relu	(None, 256)	16,640
Dropout		(None, 256)	0
Dense	Relu	(None, 256)	65,792
Dropout		(None, 256)	0
Dense	Softmax	(None, 3)	771

Table 6 Test performance of RBC

	Precision	Recall	*F*-measure
Fake	0.98	0.95	0.96
Real	0.77	0.97	0.86
Suspicious	0.97	0.82	0.89
Weighted avg	0.92	0.90	0.91
Accuracy		0.90	

seems to be well separate from the others, while real and suspicious seem to have an overlapped characteristic.

Preprocessing Setting for Machine Learning Models

In this research, we performed experiments based on two groups of machine learning. The first group was traditional machine learning comprising LR, KNN, NB, MLP, RF, and RBC. The second group was the deep learning LSTM model recurrent-based model. For the first group, the input to the models was the featured data extracted from the NLP module. Unlike the traditional models, the input of the LSTM model was a sequence of text content concatenated from retrieved news descriptions. LSTM model used the relevant news content and classified it into fake, real, or suspicious.

There were 952,387 total trainable parameters for the LSTM model. The details of LSTM model settings were as shown in Table 5.

Machine Learning Modeling Comparisons

After data collection, feature extraction, and data analysis, we built a fake news detection system. We performed model comparisons to choose the best model as a classifier in our news detection system. We selected open-source tools to construct a fake news detection system. The data analysis and machine learning modeling tools include LR, MLP, SVM, DTC, RF, NB, KNN, RB, and LSTM. The performance metrics used include accuracy, precision, recall, and *F*1-measure. Tables 6, 7, 8, 9 illustrate sample of the test performance results based on RBC, SVM, RF, and LSTM, respectively. It is noticed that the sampled models can classify fake and suspicious classes with high precision and F-measure, while they achieve lower scores with the real class. Table 10 shows the summary results of all machine learning models. Figure 6 shows a box plot based on test accuracy for 10-fold-cross-validation. It confirms that LSTM was the best model for achieving a perfect accuracy score. It can be seen that a deep learning LSTM model yields the highest accuracy, precision, recall, and F-measure with a perfect score; all accuracy, precision, recall, and f-measure are 1.00. MLP, RF, and SVM are among the second group with accuracy, precision, recall, and f-measure of 0.96-0.97. NB has the least accuracy, precision, recall, and f-measure, 0.78, 0.85, 0.78, and 0.79, respectively.

Table 7 Test performance of SVM

	Precision	Recall	*F*-measure
Fake	0.99	0.99	0.99
Real	0.89	0.99	0.94
Suspicious	0.99	0.90	0.94
Weighted avg	0.96	0.96	0.96
Accuracy		0.96	

Table 8 Test performance of RF

	Precision	Recall	*F*-measure
Fake	0.99	1.00	1.00
Real	0.90	0.99	0.94
Suspicious	1.00	0.91	0.95
Weighted avg	0.97	0.96	0.96
Accuracy		0.96	

Table 9 Test performance of LSTM

	Precision	Recall	*F*-measure
Fake	1.00	1.00	1.00
Real	1.00	1.00	1.00
Suspicious	1.00	1.00	1.00
Weighted avg	1.00	1.00	1.00
Accuracy		1.00	

Discussion

Fake news data are very dynamics. It is not an easy task to build a fake news detection system that generalizes all unseen data. Our idea is to exploit the news data on the Internet and social media by using it as inputs fed to the

Table 10 Machine learning model comparisons

Metrics|Models	NB	LR	MLP	SVM	DT	RF	KNN	RBC	LSTM
Accuracy	0.78	0.92	0.96	0.96	0.92	0.96	0.93	0.90	1.00
Precision	0.85	0.92	0.96	0.96	0.92	0.97	0.93	0.92	1.00
Recall	0.78	0.92	0.96	0.96	0.92	0.96	0.93	0.90	1.00
F-measure	0.79	0.92	0.96	0.96	0.92	0.96	0.93	0.91	1.00

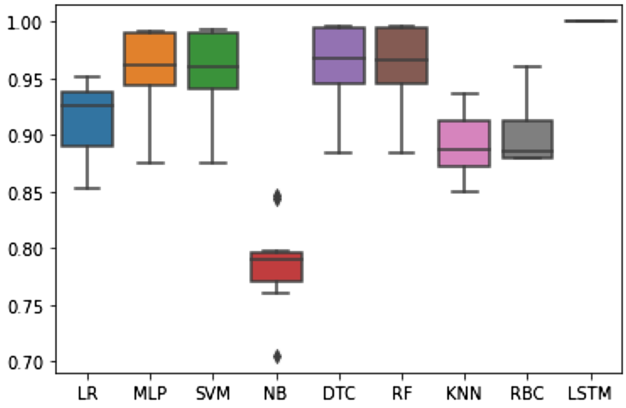


Fig. 6 Test accuracy box plot of machine learning classifier

classifier. As discussed in the data preparation process, we used web crawler-based information retrieval to retrieve the data that contain fake and real news. The feature extraction step extracts news data for five features based on fake news score, real news score, similarity matching, length of fake news domain, and length of real news domain. The extracted feature data have highly distinguished characteristics, as shown in the subsequence machine learning that can perform a classification task with higher accuracy than 90% for most of the classifier models except for the NB. It confirms that the proposed NLP-based feature selection is suitable for the fake news classification task. Besides, LSTM with concatenated text from relevant news having high similarity to the news query achieved best with a perfect test score for all metrics, including accuracy, precision, recall, and *F*-measure.

It is worth noting that the rule-based classifier provides a good feature as an explainable fake news detector. If–Then rules are good for reasoning why the classifier has such an answer to the query. A sample of If–Then rules extracted from the data are listed below.

IF score fake ≥ 9.0 AND score real ≤ 2.0 THEN label=fake

IF score fake ≥ 7.0 AND domain fake ≤ 106.0 THEN label=fake

IF score fake ≥ 3.0 AND score real ≤ 4.0 THEN label=fake

IF score fake ≥ 4.0 AND sim matched ≤ 5.0 THEN label=fake

IF score fake ≥ 4.0 AND score real ≤ 6.0 THEN label=fake

IF domain fake ≥ 29.0 AND domain real ≤ 22.0 THEN label=fake

IF score fake ≥ 4.0 AND sim matched ≥ 8.0 THEN label=fake

IF sim matched ≥ 10.0 AND domain fake ≤ 13.0 THEN label=real

IF score fake ≤ 1.0 AND score real ≥ 2.0 THEN label=real

IF score real ≥ 8.0 AND score fake ≤ 7.0 THEN label=real

IF score fake ≤ 2.0 AND score real ≥ 3.0 THEN label=real

IF score fake ≤ 1.0 AND sim matched ≥ 3.0 THEN label=real

IF domain real ≥ 20.0 AND score real ≤ 2.0 THEN label=real

IF sim matched ≤ 14.0 AND sim matched ≥ 2.0 THEN label=real

IF score fake ≤ 1.0 AND sim matched ≤ 2.0 THEN label=suspicious

IF sim matched ≤ 4.0 AND domain fake ≤ 13.0 THEN label=suspicious

IF score fake ≤ 1.0 AND domain fake ≥ 15.0 THEN label=suspicious

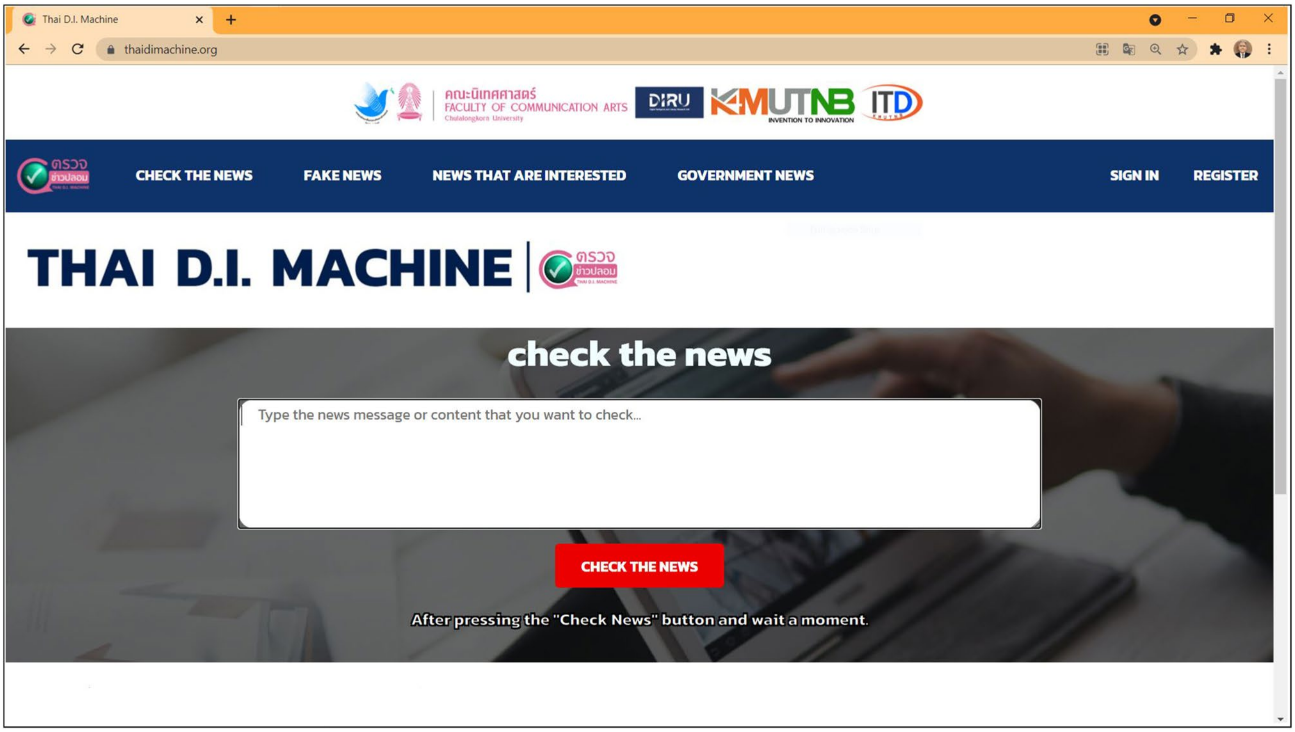
IF score fake ≤ 2.0 AND score fake ≥ 2.0 THEN label=suspicious

IF domain real ≥ 26.0 AND score fake ≤ 4.0 THEN label=suspicious

IF domain fake ≤ 13.0 AND sim matched ≥ 6.0 THEN label=suspicious

IF TRUE THEN label = suspicious

The above rule set is an ordered rules list in which the last rule is the default rule. To decide for each input featured datum, the system checks which rule matches or covers the input datum. If the input datum matches the condition of a rule (TRUE statement), the decision is the label output from the conclusion part of the covered rule. If the datum matches a rule, the decision is made based on the covered rule. There is no other rule needed to check. However, if no rule covered the input datum, the default rule “IF TRUE THEN label = suspicious” will activate. The decision label will be “suspicious.”



Type the news message or content that you want to check...

After pressing the "Check News" button and wait a moment.

Fig. 7 The automatic online Thai fake news detection

Web Application

After data preprocessing and machine learning modeling phases, we designed and developed a fake news detection system using the best-trained machine learning model. For stability, we designed cloud-based online fake news detection. We used the following tools for system development: Ubuntu 20.04 operating system, MongoDB database system for storing news data, Python for information retrieval, natural language processing, and machine learning, Django web framework for frontend web development, and Apache2 as a web server.

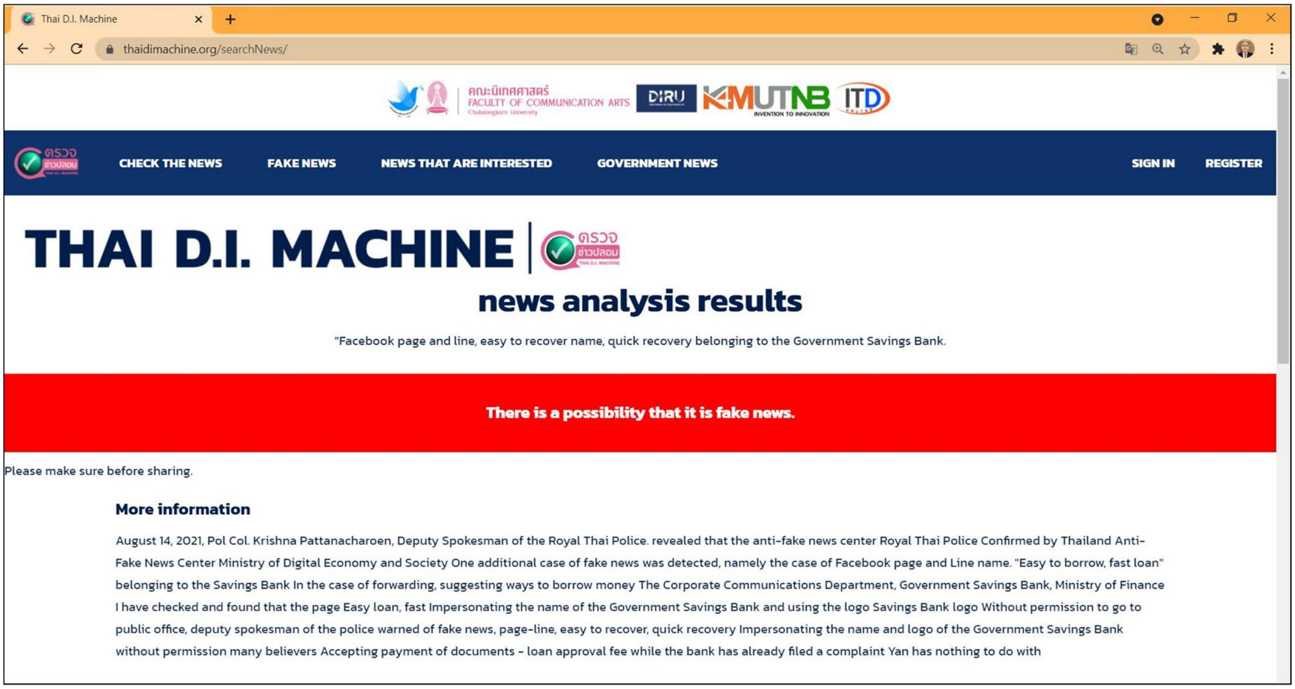
The main functions of the system include the query users entered for checking fake or real news. The web application will take the user query to analyze and return the response result with related news websites sorted based on the similarity. The user enters a news query via the text area input form, then the system in parallel sends out web-crawler information retrieval agents to fetch related news from the web and social media. The returned relevant news list is processed via the NLP module to get featured data and fed to the machine learning prediction module. The whole process time may take about 3–10 s to respond, depending on how popular the news query.

It is noticed that we used LSTM instead of BERT and GPT because we use machine learning for classifying the type of news. When a user enters a news query into the system, the user expects a fast response as the best user experience. Having too many parameters, BERT or GPT may not respond quickly enough for classifying news. Just do a classifying job, then we chose LSTM instead.

The web application provides known fake and real news articles, which are currently in the attention of social media communities. Figures 7, 8 show sample pages of the automatic online Thai fake news detection. The web application can be accessible at https://thaidimachine.org.

Conclusion

Detecting fake news is a difficult task as the news stories are very dynamic. This research proposes a new robust method to tackle fake news or misinformation. We employ three main techniques to build automatic online fake news detection. In our methodology, first, we use Information Retrieval as a mechanism to retrieve data from an online news website and social media. Next, the natural language processing analyzes the retrieved news, which results in feature data that are well distinguished. Lastly, machine learning receives the feature data and classifies the news articles into three classes: real, fake, and suspicious. We used a web robot to crawl data for 41,448 samples and pre-classified them into real, fake, and suspicious classes. The number of data samples in each group is balanced. We



THAI D.I. MACHINE

news analysis results

"Facebook page and line, easy to recover name, quick recovery belonging to the Government Savings Bank.

There is a possibility that it is fake news.

Please make sure before sharing.

More information

August 14, 2021, Pol Col. Krishna Pattanacharoen, Deputy Spokesman of the Royal Thai Police. revealed that the anti-fake news center Royal Thai Police Confirmed by Thailand Anti-Fake News Center Ministry of Digital Economy and Society One additional case of fake news was detected, namely the case of Facebook page and Line name. "Easy to borrow, fast loan" belonging to the Savings Bank In the case of forwarding, suggesting ways to borrow money The Corporate Communications Department, Government Savings Bank, Ministry of Finance I have checked and found that the page Easy loan, fast Impersonating the name of the Government Savings Bank and using the logo Savings Bank logo Without permission to go to public office, deputy spokesman of the police warned of fake news, page-line, easy to recover, quick recovery Impersonating the name and logo of the Government Savings Bank without permission many believers Accepting payment of documents - loan approval fee while the bank has already filed a complaint Yan has nothing to do with

Fig. 8 The automatic online Thai fake news detection

separate the data into three sets: training set, validation set, and test set, each for 50%, 20%, and 30%, respectively. The machine learning models used in the study were Logistic Regression (LR), K-Nearest Neighbor (KNN), Naïve Bayesian (NB), Multilayer Perceptron (MLP), Random Forest (RF), Rule-Based Classifier (RBC), and Long Short-Term Memory (LSTM). We found that LSTM was the best model that achieved 100% on test data measured by accuracy, precision, recall, and f-measure. Finally, we deployed an automatic online fake news detection web application run at https://thaidimachine.org, based on the best machine learning model.

For future research, we will explore more on the deep learning models. We have a research question on how to make deep learning understand the news more as humans do. The machine can predict the type of news and explain why it gives such an answer or response. Besides, if news contents comprise text, sound, and video, the machine must analyze and respond correctly. In addition, we plan to apply study for a deeper understanding of the language. We will need a more complex architecture such as BERT and GPT. That leaves for further investigation in the future.

Acknowledgements This paper was part of research project entitled "Development of Fake News Detection Using Machine Learning and People's Fact Checking" funded by Thai Media Fund, the fiscal year 2019, granted project id: 2562-T3/0064. The author would like to thanks Associate Professor Phnom Kleechaya, who is the head of Digital Intelligence and Literacy Research Unit (DIRU), Faculty of Communication Art, Chulalongkorn University, for supporting this research.

Declarations

Conflict of interest The author declares that there is no conflict of interest.

References

1. Rodríguez ÁI and Iglesias LL. Fake news detection using deep learning. ArXiv. 2019. arXiv:1910.03496.
2. Allcott H, Gentzkow M. Social media and fake news in the 2016 election. J Econ Perspect. 2017;31(2):211–36. https://doi.org/10.1257/jep.31.2.211.
3. Jiang T, Li JP, Haq AU, Saboor A, Ali A. A novel stacking approach for accurate detection of fake news. IEEE Access. 2021;9:22626–39. https://doi.org/10.1109/ACCESS.2021.3056079.
4. Rahman MS, Halder S, Uddin MA, Acharjee UK. An efficient hybrid system for anomaly detection in social networks. Cybersecurity. 2021. https://doi.org/10.1186/s42400-021-00074-w.
5. Lakshmanan LVS, Simpson M, Thirumuruganathan S. Combating fake news: a data management and mining perspective. Proc VLDB Endow. 2019;12(12):1990–3. https://doi.org/10.14778/3352063.3352117.
6. Shu K, Wang S, and Liu H. Beyond news contents: the role of social context for fake news detection. In: Proc ACM Inter Con

on Web Search and Data Mining. 2019. pp. 312–320. https://doi.org/10.1145/3289600.3290994.

7. Yanagi Y, Orihara R, Sei Y, Tahara Y, and Ohsuga A. Fake news detection with generated comments for news articles. In: IEEE 24th Inter Con Intelligent Engineering Systems (INES). 2020. pp. 85–90. https://doi.org/10.1109/INES49302.2020.9147195.
8. Umer M, Imtiaz Z, Ullah S, Mehmood A, Choi GS, On BW. Fake news stance detection using deep learning architecture (CNN-LSTM). IEEE Access. 2020;8:156695–706. https://doi.org/10.1109/ACCESS.2020.3019735.
9. Akhter MP, Zheng J, Afzal F, Lin H, Riaz S, Mehmood A. Supervised ensemble learning methods towards automatically filtering Urdu fake news within social media. PeerJ Comput Sci. 2021;7:1–24. https://doi.org/10.7717/peerj-cs.425.
10. Aphiwongsophon S and Chongstitvatana P. Detecting fake news with machine learning method. In: 2018 15th Inter Con Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). 2018. pp. 528–531. https://doi.org/10.1109/ECTICon.2018.8620051.
11. Mookdarsanit P and Mookdarsanit L. The COVID-19 fake news detection in Thai social texts. 2021;10(2):988–998. https://doi.org/10.11591/eei.v10i2.2745.
12. Ireton C, Posetti J, and UNESCO, Journalism. fake news. et disinformation: handbook for journalism education and training. 2018.
13. Quandt T, Frischlich L, Boberg S, and Schatto–Eckrodt T. Fake news. In: Int Enc J Studies, American Cancer Society. 2019. pp. 1–6.
14. Lee N, et al. On unifying misinformation detection. ArXiv210405243 Cs. 2021. arXiv:2104.05243. Accessed 24 Apr 2021.
15. Shu K, Wang S, Lee D, and Liu H. Mining disinformation and fake news: concepts, methods, and recent advancements. 2020. https://www.arxiv-vanity.com/papers/2001.00623/. Accessed 25 Apr 2021.
16. Ahmed AAA, Aljabouh A, Donepudi PK, and Choi MS. Detecting fake news using machine learning: a systematic literature review. ArXiv210204458 Cs. 2021. arXiv:2102.04458. Accessed 24 Apr 2021.
17. Guo M, Chen X, Li J, Zhao D, and Yan R. How does Truth Evolve into Fake News? An Empirical Study of Fake News Evolution. ArXiv210305944 Cs. 2021. arXiv:2103.05944. Accessed 24 Apr 2021.
18. Parikh SB and Atrey PK. Media-Rich Fake news detection: a survey. In: 2018 IEEE Con Mult Infor Proc and Ret (MIPR). https://doi.org/10.1109/MIPR.2018.00093. 2018. pp. 436–441.
19. Wang X, Gao L, Song J, Shen H. Beyond frame-level CNN: saliency-aware 3-D CNN with LSTM for video action recognition. IEEE Signal Process Lett. 2017;24(4):510–4. https://doi.org/10.1109/LSP.2016.2611485.
20. Lashkari AH, Mahdavi F, and Ghomi V. A boolean model in information retrieval for search engines. In: Int Con Inf Man and Eng. https://doi.org/10.1109/ICIME.2009.101. 2009. pp. 385–389.
21. Billhardt H, Borrajo D, Maojo V. A context vector model for information retrieval. J Am Soc Inf Sci Technol. 2002;53(3):236–49. https://doi.org/10.1002/asi.10032.
22. Salton G, Wong A, Yang CS. A vector space model for automatic indexing. Commun ACM. 1975;18(11):613–20. https://doi.org/10.1145/361219.361220.
23. Robertson S, Zaragoza H. The probabilistic relevance framework: BM25 and beyond. Foundations TrendsR Inf Retrieval. 2009;3(4):333–89. https://doi.org/10.1561/1500000019.
24. Jing K and Xu J. A survey on neural network language models. ArXiv190603591 Cs. 2019. arXiv:1906.03591. Accessed 20 Mar 2020.
25. Zhang F, Fleyeh H, Wang X, Lu M. Construction site accident analysis using text mining and natural language processing techniques. Autom Constr. 2019;99:238–48. https://doi.org/10.1016/j.autcon.2018.12.016.
26. Chirawichitchai N, Sa-nguansat P, and Meesad P. Developing an effective Thai document categorization framework base on term relevance frequency weighting. In: Eighth Int Con ICT and Know Eng. https://doi.org/10.1109/ICTKE.2010.5692907. 2010. pp. 19–23.
27. Lample G, Ballesteros M, Subramanian S, Kawakami K, and Dyer C. Neural architectures for named entity recognition. ArXiv160301360 Cs. 2016. arXiv:1603.01360. Accessed 21 Apr 2021.
28. Sharma Y, Agrawal G, Jain P, and Kumar T. Vector representation of words for sentiment analysis using GloVe. In: Int Con Int Com and Comp Tech (ICCT). https://doi.org/10.1109/INTELCCT.2017.8324059. 2017. pp. 279–284.
29. Chormai P, Prasertsom P, and Rutherford A. AttaCut: a fast and accurate neural Thai word segmenter. ArXiv191107056 Cs. 2019, arXiv:1911.07056. Accessed 21 Apr 2021.
30. Phatthiyaphaibun W, et al. PyThaiNLP v2.3.1 release!. Zenodo. 2021. https://doi.org/10.5281/zenodo.4662045. Accessed 29 Apr 2021.
31. Kleinbaum DG, Klein M. Logistic regression: a self-learning text. 3rd ed. Springer; 2010.
32. LaValley MP. Logistic Regression. Circulation. 2008;117(18):2395–9. https://doi.org/10.1161/CIRCULATIONAHA.106.682658.
33. Wright RE. Logistic regression. In: Reading and understanding multivariate statistics. Washington: American Psychological Association; 1995. pp. 217–244.
34. Guo G, Wang H, Bell D, Bi Y, and Greer K. KNN model-based approach in classification. In: On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, 2888, R. Meersman, Z. Tari, and D. C. Schmidt, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg; 2003. pp. 986–996.
35. Clark P and Boswell R. Rule induction with CN2: some recent improvements. In: machine learning - EWSL-91. Berlin: Heidelberg; 1991, pp. 151–163. https://doi.org/10.1007/BFb0017011.
36. Clark P, Niblett T. The CN2 induction algorithm. Mach Learn. 1989;3(4):261–83. https://doi.org/10.1023/A:1022641700528.
37. Hamsa H, Indiradevi S, Kizhakkethottam JJ. Student academic performance prediction model using decision tree and fuzzy genetic algorithm. Procedia Technol. 2016;25:326–32. https://doi.org/10.1016/j.protcy.2016.08.114.
38. Myles AJ, Feudale RN, Liu Y, Woody NA, Brown SD. An introduction to decision tree modeling. J Chemom. 2004;18(6):275–85. https://doi.org/10.1002/cem.873.
39. Quinlan JR. Improved use of continuous attributes in C4.5. J Artif Intell Res. 1996;4:77–90. https://doi.org/10.1613/jair.279.
40. Yang H, Xu A, Chen H, and Yuan C. A Review: the effects of imperfect data on incremental decision tree. In: Ninth Inter Con P2P, Parallel, Grid, Cloud and Internet Computing. 2014. pp. 34–41. https://doi.org/10.1109/3PGCIC.2014.34.
41. Ahmad I, Yousaf M, Yousaf S, Ahmad MO. Fake news detection using machine learning ensemble methods. Complexity. 2020;2020:e8885861. https://doi.org/10.1155/2020/8885861.
42. Misra S and Li H. Chapter 9—noninvasive fracture characterization based on the classification of sonic wave travel times. In: Misra S, Li H, and He J. (Eds.). Machine Learning for Subsurface Characterization. Gulf Professional Publishing. 2020. pp. 243–287.
43. Shrivastava D, Sanyal S, Maji AK, and Kandar D. Chapter 17 - Bone cancer detection using machine learning techniques. In: Paul

S and Bhatia D. (Eds.). Smart Healthcare for Disease Diagnosis and Prevention. Academic Press. 2020. pp. 175–183.

44. Svetnik V, Liaw A, Tong C, Culberson JC, Sheridan RP, Feuston BP. Random forest: a classification and regression tool for compound classification and QSAR modeling. J Chem Inf Comput Sci. 2003;43(6):1947–58. https://doi.org/10.1021/ci034160g.
45. Becker B, Kohavi R, Sommerfield D. Visualizing the Simple Bayesian Classifier. 1997.
46. Kim HC and Ghahramani Z. Bayesian classifier combination. In: Artif Intell and Stat. pp. 619–627. 2012. http://proceedings.mlr.press/v22/kim12.html. Accessed 24 Apr 2021.
47. Yager RR. An extension of the naive Bayesian classifier. Inf Sci. 2006;176(5):577–88. https://doi.org/10.1016/j.ins.2004.12.006.
48. Zhang H. The optimality of naive bayes. In: Proc FLAIRS. 2004. p.6.
49. Hagan MT, Demuth HB, Beale MH, Jesús OD. Neural network design. 2nd ed. Wrocław: Martin Hagan; 2014.
50. Morshedizadeh M, Kordestani M, Carriveau R, Ting DSK, Saif M. Power production prediction of wind turbines using a fusion of MLP and ANFIS networks. IET Renew Power Gener. 2018;12(9):1025–33. https://doi.org/10.1049/iet-rpg.2017.0736.
51. Zhang L, Tian F. Performance study of multilayer perceptrons in a low-cost electronic Nose. IEEE Trans Instrum Meas. 2014;63(7):1670–9. https://doi.org/10.1109/TIM.2014.2298691.
52. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995;20(3):273–97. https://doi.org/10.1007/BF00994018.
53. Chen TT, Lee SJ. A weighted LS-SVM based learning system for time series forecasting. Inf Sci. 2014;299:99–116. https://doi.org/10.1016/j.ins.2014.12.031.
54. Mareeswari V and Gunasekaran G. Prevention of credit card fraud detection based on HSVM. In: 2016 Inter Con Infor Com and Emb Sys (ICICES). 2016. pp. 1–4. https://doi.org/10.1109/ICICES.2016.7518889.
55. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.
56. Choudhury N, Faisal F, and Khushi M. Towards an LSTM-based predictive framework for literature-based knowledge discovery. ArXiv190709395 Cs. 2019. arXiv:1907.09395. Accessed 6 Sep 2019.
57. Gers FA, Schmidhuber J and Cummins F. Learning to forget: continual prediction with LSTM. In: Proc ICANN 99. (Conf. Publ. No. 470), 1999. pp. 850–855. https://doi.org/10.1049/cp:19991218.
58. Luo C, Zhan J, Xue X, Wang L, Ren R, and Yang Q. Cosine Normalization: Using Cosine Similarity Instead of Dot Product in Neural Networks. In: Proc ICANN, 2018. pp. 382–391, https://doi.org/10.1007/978-3-030-01418-6_38.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.