BioData Mining

RESEARCH Open Access

# Influenza, dengue and common cold detection using LSTM with fully connected neural network and keywords selection

Wanchaloem Nadda[1], Waraporn Boonchieng[2] and Ekkarat Boonchieng[3]*

* Correspondence: ekkarat.boonchieng@cmu.ac.th
[3]Center of Excellence in Community Health Informatics, Department of Computer Science, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand
Full list of author information is available at the end of the article

## Abstract

Symptom-based machine learning models for disease detection are a way to reduce the workload of doctors when they have too many patients. Currently, there are many research studies on machine learning or deep learning for disease detection or clinical departments classification, using text of patient's symptoms and vital signs. In this study, we used the Long Short-term Memory (LSTM) with a fully connected neural network model for classification, where the LSTM model was used to receive the patient's symptoms text as input data. The fully connected neural network was used to receive other input data from the patients, including body temperature, age, gender, and the month the patients received care in. In this research, a data preprocessing algorithm was improved by using keyword selection to reduce the complexity of input data for overfitting problem prevention. The results showed that the LSTM with fully connected neural network model performed better than the LSTM model. The keyword selection method also increases model performance.

**Keywords:** Long short-term memory, Dengue detection, Influenza detection, Text mining

## Introduction

Symptom-based machine learning models help patients self-detect diseases via electronic devices such as smart phones or robots in hospitals with automated question and answer systems [7]. Recently, several studies improved the text classification model for clinical department classification [27] and disease detection [12]. These studies used text from symptoms and other features of patients for disease detection [17].

Dengue fever (a mosquito-borne viral disease) [18] and influenza are dangerous infectious diseases that many people contract. Dengue and influenza have symptoms like the common cold, but they can be fatal. It is estimated that 3 to 5 million people each year become seriously ill due to influenza [21].

The research about machine learning or deep learning for dengue and influenza is divided into two parts, improvement prediction models for forecasting the number of patients [25] or forecasting an outbreak [8] in some areas or countries such as China

[26], India [16], and Thailand [22]. Another type of research is focused on improving machine learning or deep learning models for detection of dengue fever and influenza from vital signs [6] and symptoms [1] of patients.

The Long Short-term Memory (LSTM) model is a recurrent neural network model. It is commonly used in text classification [13], time series classification [11], and time series forecasting [25].

In this research, we will use the LSTM model to classify the symptoms of patients as text. The LSTM model was concatenated with a fully connected neural network to use patient vital signs and other features as input data, including gender, body temperature, and age of patients to increase the performance of the classification model. Moreover, we improve our method for data preprocessing by removing words that are not important to classification, this simplifies the input data.

## Theorical foundations

In this section, we describe all of the methods we used for modeling in this research.

### Mutual information metric

Mutual information metric (MI) is a value used to show the ability to classify each keyword. We use MI to measure the correlation between each keyword and each class. Mutual information metric is denoted by $\mathrm{MI}(w, c)$, where $w$ is a word and $c$ is a class. It is calculated by Eq. (1).

$$\mathrm{MI}(w,c) = \log \frac{f_A \cdot N}{(f_A + f_C)(f_A + f_B)} \tag{1}$$

When $f_A$ is the number of documents in class $c$ that contain word $w$, $f_B$ is the number of the documents not in class $c$ that contain word $w$, $f_C$ is the number of the documents not in class $c$ that do not contain word $w$. and $N$ is the number of all documents. The $\mathrm{MI}(w, c)$ has a value in range $[-\log(N), \log(N)]$ this is shown in (2) and (3).

$$\mathrm{MI}(w,c) = \log \frac{f_A \cdot N}{(f_A + f_C)(f_A + f_B)} \leq \log \frac{N}{(f_A + f_B)} \leq \log(N) \tag{2}$$

$$\begin{aligned}\mathrm{MI}(w,c) = {} & \log \frac{f_A \cdot N}{(f_A + f_C)(f_A + f_B)} \geq \log \frac{f_A}{(f_A + f_C)} \geq \log \frac{f_A}{(f_A + f_C)} \geq \log \frac{1}{N} \\ = {} & -\log(N)\end{aligned} \tag{3}$$

The MI of each word can be measured by finding the MI between the word and the class with the highest MI value. It is shown in Eq. (4) where $d$ is the number of classes.

$$\mathrm{MI}(w) = \max_{i=1:d} \mathrm{MI}(w, c_i) \tag{4}$$

The MI is the largest in the case of $f_A = 1$, $f_B = 0$, and $f_C = 0$ .The words that have a frequency of 1 are important for classification.

### Word embedding

Word embedding is the method for representing each word with a vector of a real number. Word2vec [15] is a method of word embedding, where neighbors' vectors of each word represents words with similar meaning. We can set the dimension of the vectors for each word when we train the word2vec model. If we use a pre-train word2-vec model, we can use the principal component analysis (PCA) to reduce the dimension of the vector of words to the dimension that we want.

### Interpolation

Interpolation is a method for estimating the missing data using polynomial or other functions [2], to obtain some points of data. An example for calculating the missing point of equation $y = sin(x)$ is shown in Fig. 1.

### LSTM

Long Short-term memory Neural Network (LSTM) [9] is a model architecture for recurrent neural network (RNN). The input data for each record of LSTM model is a sequence of vectors. A structure of LSTM is shown in Fig. 2 where $X_t$ is a vector of input data with time stamp $t$.

The LSTM model is used for classification or prediction of sequential input data. In the present, the LSTM has had several improvements and has been used in several ways for time series prediction and text classification, such as LSTM fully convolutional networks for time series classification [11], bidirectional LSTM for sentiment analysis [13] and medical text classification [7].

### Imbalanced data problem

The imbalanced data problem is a problem of data classification, when the number of records in each class is vastly different [19]. In the case of binary class classification, we call the class with more records than the other class the majority class and call the other class the minority class.

There are two popular methods for solving the imbalanced data problem:

1) Using under sampling or oversampling for sampling training data in each class to have the same number of records.
2) Using some loss functions for machine learning or deep learning model to increase the weight of the minority class.

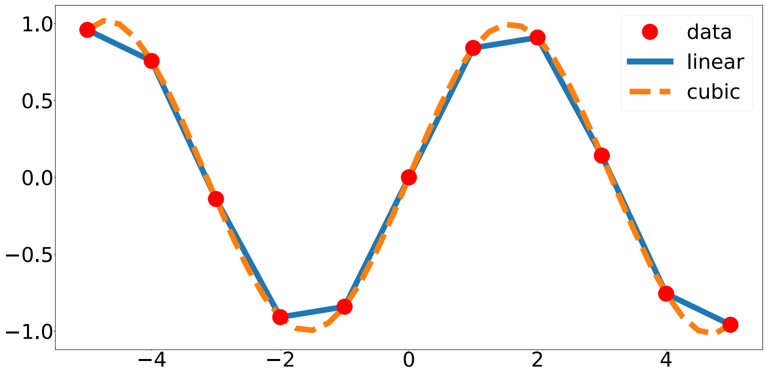

**Fig. 1** Data interpolation with linear and cubic functions

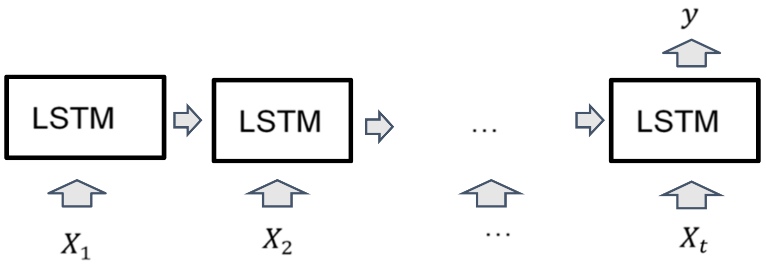

**Fig. 2** LSTM model structure

In this research we use the cost-entropy loss function [24] in Eq. (6) for the loss function of LSTM model for solving the imbalanced data problem. It has been improved upon from the cost-entropy loss in Eq. (5) where $t_k = [t_k(1), t_k(2), \ldots, t_k(d)\ ]$ is the vector of target output of $k^{th}$ record of dataset, $t_k(i) \in \{0, 1\}$ for $i = 1, 2, \ldots, d$, and $y_k = [y_k(1), y_k(2), \ldots, y_k(d)\ ]$ is the vector of output of model for $k^{th}$ record of dataset, and $y_k(i) \in (0, 1)$ for $i = 1, 2, \ldots, d$. Moreover, we set $n_k$ to be the number of records of training data in the class of $k^{th}$ record and set a constant value $\gamma \in [0, 1]$.

$$E = -\sum_{i=1}^{n}\sum_{k=1}^{d} t_k(i)\ \log y_k(i) \tag{5}$$

$$E = -\sum_{i=1}^{n}\sum_{k=1}^{d} t_k(i)\ \log y_k(i)\left(\frac{1}{n_k}\right)^{\gamma} \tag{6}$$

## Material and methods

### Data description

The data used in this research is from medical records from Saraphi Hospital, Chiang Mai Province, Thailand Between 2015 and 2020 [3–5]. We use only records of patients diagnosed with three diseases. This includes the common cold, flu, and dengue. We listed all the attributes we used in this research in Table 1.

The distribution (average and standard deviation) of some features and the number of records for each class are shown in Table 2.

From the statistical hypothesis test (t-test), it was found that:

1) Average of age: It was found that the mean of age of common cold patients was greater

**Table 1** The attributes are used in this research

| Attribute | Description |
|---|---|
| CHIEFCOMP | Text of symptoms of each patient |
| GENDER | Gender of each patient (0 = male, 1 = female) |
| MONTH_SERV | The month, that each patient comes to the hospital in each time. |
| BTEMP | Body temperature of each patient |
| AGE | Age of each patient (year of service minus by year of birth) |

**Table 2** The average, standard deviation, and number of patients for some features

| Attributes\Classes | | cold | Dengue | flu | all |
|---|---|---|---|---|---|
| AGE (years) | mean | 36.188 | 27.269 | 32.6 | 36.002 |
| | std | 26.933 | 15.643 | 20.813 | 26.714 |
| BTEMP (°*C*) | mean | 36.793 | 37.248 | 37.784 | 36.824 |
| | std | 0.827 | 1.191 | 1.148 | 0.858 |
| GENDER (records) | male | 2188 | 25 | 64 | 2277 |
| | female | 2802 | 27 | 76 | 2905 |
| number of records (records) | | 4990 | 52 | 140 | 5182 |
| Length of sentence (words) | mean | 7.930 | 5.096 | 5.771 | 7.843 |
| | std | 6.208 | 2.320 | 5.164 | 6.171 |
| Number of words (words) | | 1279 | 102 | 158 | 1306 |
| Word frequency (records) | mean | 29.18 | 2.57 | 4.86 | 29.36 |
| | std | 159.19 | 3.32 | 10.58 | 161.66 |

than the mean of age of dengue and flu patients ($p$-value < 0.05), but the mean of age of dengue and flu patients was no different. ($p$-value > 0.05).

2) Average body temperature: It was found that the mean body temperature of common cold

patients were less than the mean of body temperature of dengue patients ($p$-value < 0.01), and the mean of body temperature of dengue patients was less than the mean of body temperature of flu patients ($p$-value < 0.01).

## Data preprocessing

In this research, the features used for classification include CHIEFCOMP, GENDER, MONTH_SERV, BTEMP, and AGE. For numerical features (BTEMP and AGE), we use min-max normalization to adjust the values in range [0,1]. Examples of data are shown in Table 3. For MONTH_SERV, we use one hot encoder to convert each value to a vector of integers. For the CHIEFCOMP column, the data in this column is a sentence in the Thai language. We use a python library "pythainlp" [20] for word tokenization. Here is an example of word tokenization, from the sentence "เป็นหวัดมีน้ำมูกไอ" (English: "Having a cold with a runny nose

**Table 3** Examples of data in our dataset

| Attribute | 1st patient | 2nd patient |
|---|---|---|
| CHIEFCOMP | (English: having a cold with a runny nose and cough) | 6<br>(English: 6-day fever, weakness, cough, and sore throat) |
| GENDER | 1 | 0 |
| MONTH_SERV | 08 (August) | 03 (March) |
| BTEMP | 36.5 | 38 |
| AGE | 50 | 23 |
| DISEASE | cold | dengue |

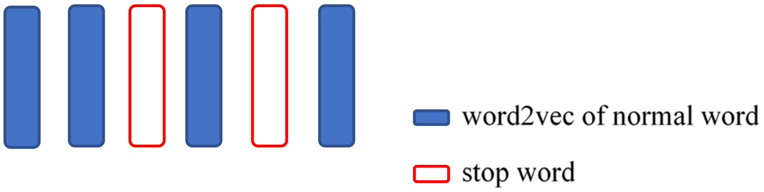

**Fig. 3** Vectors of words in a sentence after the removal of 2 stop words

and cough") to a list of words ["เป็น", "หวัด", "มี", "น้ำมูก", "ไอ"]. Then the python library "Gensim" [14] is used to create a word2vec model that converts the text of each record into a matrix of a real number.

### Keywords selection

In the process of text preprocessing for LSTM training. We removed words that were not important for classification to simplify the incoming data including:

1. Low MI: words with low mutual information metric (bottoms 5%).

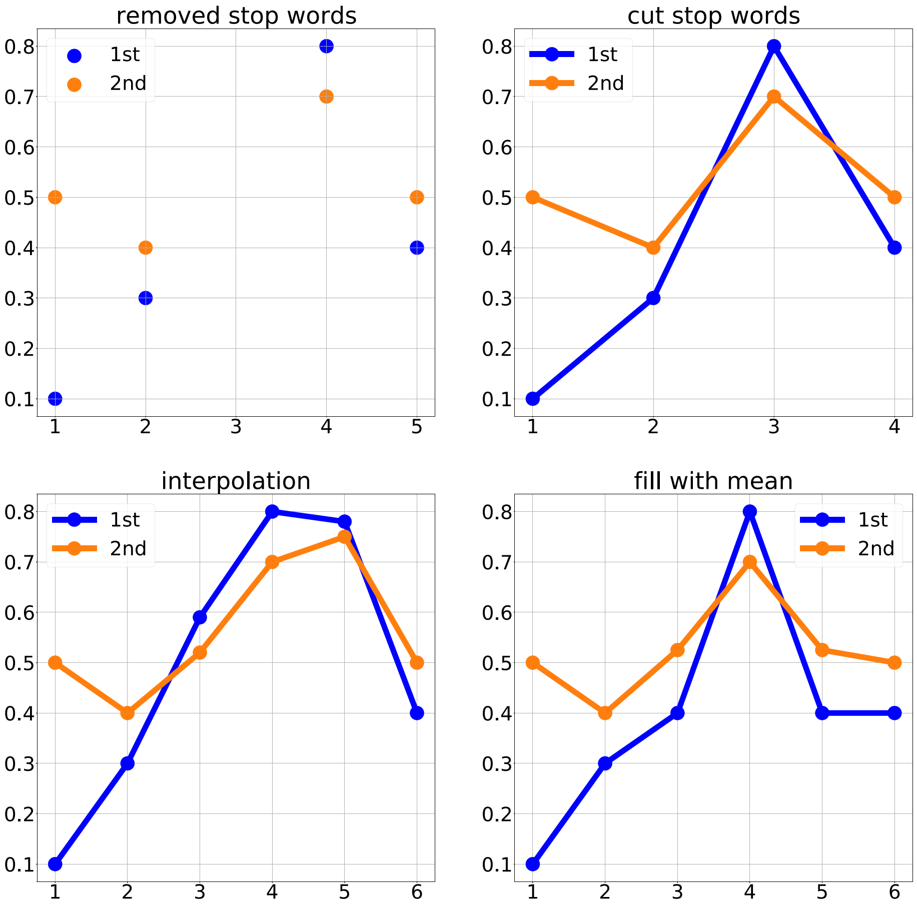

**Fig. 4** Solving missing values problem

2. Low frequency: words with low frequency (frequency < 2) because it had high MI. That is, it has a high ability for classification. However, it may be a typographical error.

These words are defined as stop words, and all stop words are removed from the data. Next, we set the positions of the removed words to missing values. It is shown in Fig. 3.

We use three methods to solve the missing values problem:

1. Cut the stop words: cut the vectors of all stop words in the sentence.
2. Fill with mean: fill the vectors of the missing values by the mean of word2vec of all words in the sentence with the corresponding position.

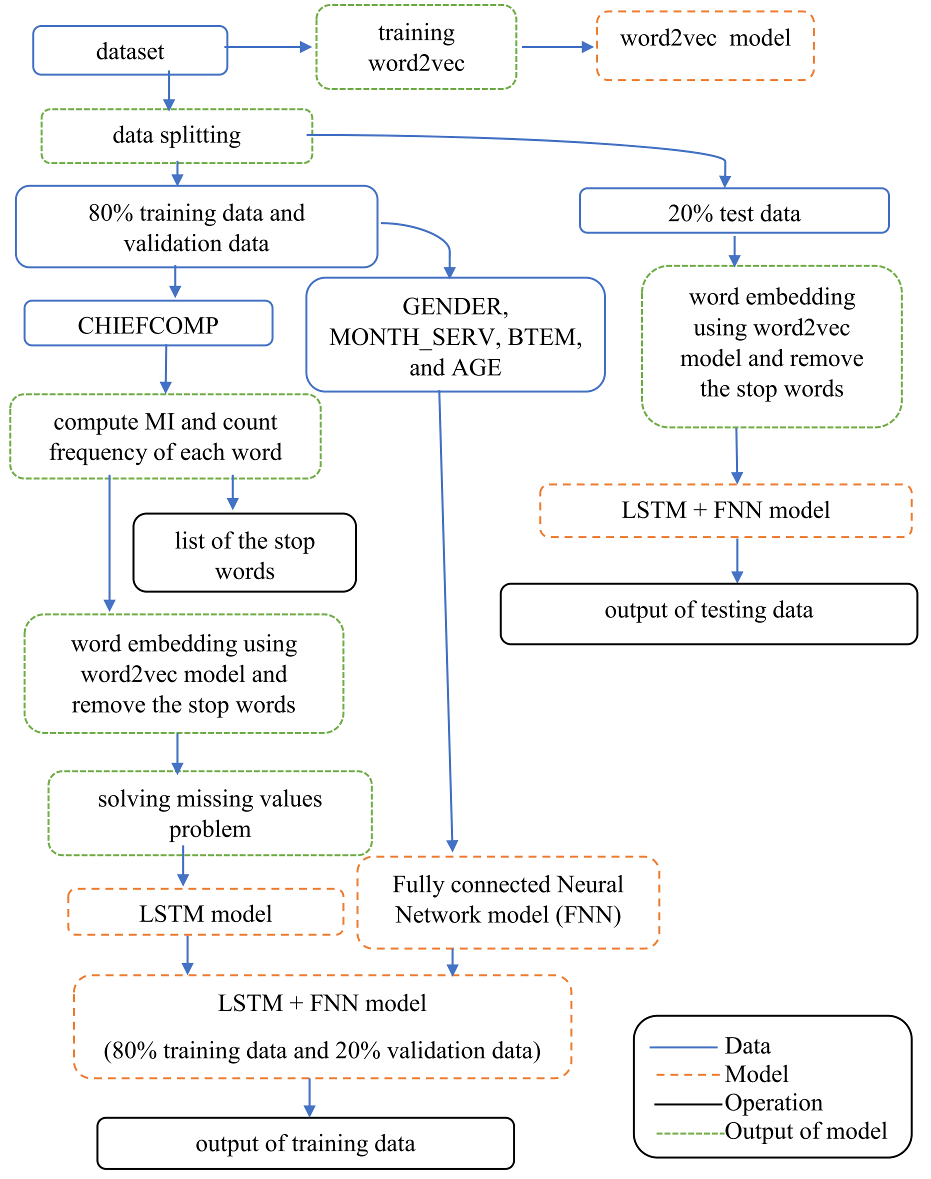
dataset
training word2vec
word2vec model
data splitting
80% training data and validation data
20% test data
CHIEFCOMP
GENDER, MONTH_SERV, BTEM, and AGE
word embedding using word2vec model and remove the stop words
compute MI and count frequency of each word
LSTM + FNN model
list of the stop words
output of testing data
word embedding using word2vec model and remove the stop words
solving missing values problem
LSTM model
Fully connected Neural Network model (FNN)
LSTM + FNN model
(80% training data and 20% validation data)
output of training data
Data
Model
Operation
Output of model

**Fig. 5** Research conceptual framework

**Table 4** Performance of models -- Area under the ROC Curve (AUC)

| | Dataset | dengue + cold | | | flu + dengue + cold | | | flu + cold | | | SMS Spam Collection Dataset |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Model Architecture / Filing missing** | **LSTM** | **LSTM with numerical** | **LSTM + FNN** | **LSTM** | **LSTM with numerical** | **LSTM + FNN** | **LSTM** | **LSTM with numerical** | **LSTM + FNN** | **LSTM** |
| original | | 0.798 | 0.823 | 0.829 | 0.753 | 0.767 | 0.776 | 0.670 | 0.674 | 0.662 | 0.912 |
| MG | | 0.797 | 0.808 | 0.826 | 0.750 | 0.757 | 0.779 | 0.678 | 0.670 | 0.676 | 0.913 |
| keywords selection (cut words: frequency < 2) | cubic interpolation | 0.798 | 0.791 | 0.817 | 0.728 | 0.755 | 0.738 | 0.734 | 0.770 | 0.795 | 0.920 |
| | cut | 0.803 | 0.831 | 0.837 | 0.723 | 0.778 | 0.782 | 0.671 | 0.675 | 0.658 | 0.922 |
| | mean | 0.802 | 0.800 | 0.825 | 0.719 | 0.781 | 0.760 | 0.729 | 0.743 | 0.692 | 0.959 |
| keywords selection (cut words: MI bottom 5%) | cubic interpolation | 0.744 | 0.628 | 0.684 | 0.622 | 0.675 | 0.594 | 0.691 | 0.808 | 0.831 | 0.930 |
| | cut | 0.712 | 0.641 | 0.754 | 0.689 | 0.753 | 0.716 | 0.653 | 0.616 | 0.551 | 0.929 |
| | mean | 0.721 | 0.637 | 0.718 | 0.681 | 0.769 | 0.717 | 0.686 | 0.776 | 0.673 | 0.968 |
| keywords selection (cut words: MI bottom 5% or frequency < 2) | cubic interpolation | 0.776 | 0.645 | 0.689 | 0.623 | 0.683 | 0.742 | 0.688 | 0.798 | 0.841 | 0.928 |
| | cut | 0.750 | 0.650 | 0.758 | 0.691 | 0.763 | 0.727 | 0.658 | 0.621 | 0.550 | 0.944 |
| | mean | 0.754 | 0.644 | 0.725 | 0.677 | 0.775 | 0.696 | 0.688 | 0.792 | 0.628 | 0.965 |
| keywords selection (cut words: frequency < 2) + MG | cubic interpolation | 0.794 | 0.784 | 0.809 | 0.743 | 0.752 | 0.742 | 0.665 | 0.755 | 0.818 | 0.916 |
| | cut | 0.794 | 0.807 | 0.845 | 0.764 | 0.759 | 0.786 | 0.682 | 0.670 | 0.675 | 0.904 |
| | mean | 0.801 | 0.784 | 0.826 | 0.769 | 0.778 | 0.771 | 0.669 | 0.713 | 0.717 | 0.952 |
| keywords selection (cut words: MI bottom 5%) + MG | cubic interpolation | 0.681 | 0.626 | 0.662 | 0.643 | 0.659 | 0.591 | 0.726 | 0.803 | 0.836 | 0.933 |
| | cut | 0.666 | 0.635 | 0.706 | 0.668 | 0.750 | 0.705 | 0.660 | 0.622 | 0.591 | 0.939 |
| | mean | 0.693 | 0.640 | 0.690 | 0.660 | 0.771 | 0.690 | 0.671 | 0.702 | 0.797 | 0.959 |
| keywords selection (cut words: MI bottom 5% or frequency < 2) + MG | cubic interpolation | 0.709 | 0.638 | 0.661 | 0.642 | 0.663 | 0.720 | 0.705 | 0.793 | 0.845 | 0.919 |
| | cut | 0.699 | 0.637 | 0.700 | 0.671 | 0.746 | 0.735 | 0.663 | 0.628 | 0.593 | 0.927 |
| | mean | 0.678 | 0.657 | 0.683 | 0.677 | 0.765 | 0.698 | 0.687 | 0.698 | 0.766 | 0.951 |

**Table 5** Performance of models (G-mean)

| | Dataset | dengue + cold | | | flu+ dengue + cold | | | flu+ cold | | | SMS Spam Collection Dataset |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Model Architecture / Filing missing** | **LSTM** | **LSTM with numerical** | **LSTM + FNN** | **LSTM** | **LSTM with numerical** | **-LSTM + FNN** | **LSTM** | **LSTM with numerical** | **LSTM + FNN** | **LSTM** |
| original | | 0.692 | 0.800 | 0.771 | 0.458 | 0.586 | 0.543 | 0.528 | 0.532 | 0.565 | 0.854 |
| MG | | 0.729 | 0.766 | 0.773 | 0.588 | 0.592 | 0.531 | 0.526 | 0.524 | 0.481 | 0.852 |
| keywords selection (cut words: frequency < 2) | cubic interpolation | 0.768 | 0.752 | 0.779 | 0.488 | 0.533 | 0.513 | 0.649 | 0.682 | 0.695 | 0.820 |
| | cut | 0.758 | 0.798 | 0.808 | 0.402 | 0.594 | 0.601 | 0.530 | 0.532 | 0.565 | 0.849 |
| | mean | 0.732 | 0.699 | 0.742 | 0.498 | 0.569 | 0.568 | 0.624 | 0.682 | 0.608 | 0.894 |
| keywords selection (cut words: MI bottom 5%) | cubic interpolation | 0.734 | 0.545 | 0.649 | 0.300 | 0.383 | 0.286 | 0.670 | 0.719 | 0.744 | 0.871 |
| | cut | 0.690 | 0.569 | 0.697 | 0.522 | 0.546 | 0.460 | 0.317 | 0.397 | 0.574 | 0.860 |
| | mean | 0.641 | 0.613 | 0.599 | 0.542 | 0.505 | 0.528 | 0.678 | 0.701 | 0.607 | 0.896 |
| keywords selection (cut words: MI bottom 5% or frequency < 2) | cubic interpolation | 0.715 | 0.628 | 0.599 | 0.372 | 0.392 | 0.439 | 0.667 | 0.687 | 0.757 | 0.851 |
| | cut | 0.668 | 0.568 | 0.714 | 0.497 | 0.556 | 0.478 | 0.319 | 0.357 | 0.569 | 0.886 |
| | mean | 0.714 | 0.599 | 0.605 | 0.488 | 0.555 | 0.560 | 0.676 | 0.675 | 0.609 | 0.900 |
| keywords selection (cut words: frequency < 2) + MG | cubic interpolation | 0.763 | 0.759 | 0.762 | 0.466 | 0.532 | 0.529 | 0.565 | 0.671 | 0.711 | 0.849 |
| | cut | 0.725 | 0.723 | 0.818 | 0.482 | 0.576 | 0.528 | 0.526 | 0.526 | 0.483 | 0.845 |
| | mean | 0.738 | 0.728 | 0.782 | 0.584 | 0.596 | 0.553 | 0.621 | 0.654 | 0.673 | 0.876 |
| keywords selection (cut words: MI bottom 5%) + MG | cubic interpolation | 0.698 | 0.606 | 0.620 | 0.222 | 0.000 | 0.282 | 0.661 | 0.728 | 0.754 | 0.864 |
| | cut | 0.680 | 0.655 | 0.641 | 0.484 | 0.555 | 0.455 | 0.402 | 0.358 | 0.495 | 0.876 |
| | mean | 0.697 | 0.621 | 0.590 | 0.490 | 0.501 | 0.477 | 0.656 | 0.642 | 0.766 | 0.894 |
| keywords selection (cut words: MI bottom 5% or frequency < 2) + MG | cubic interpolation | 0.703 | 0.604 | 0.581 | 0.293 | 0.000 | 0.405 | 0.677 | 0.722 | 0.716 | 0.845 |
| | cut | 0.671 | 0.609 | 0.642 | 0.483 | 0.537 | 0.504 | 0.364 | 0.359 | 0.492 | 0.836 |
| | mean | 0.697 | 0.621 | 0.590 | 0.478 | 0.563 | 0.529 | 0.678 | 0.647 | 0.666 | 0.884 |

3. Interpolation: fill the vectors with the missing values by interpolation using the corresponding position in vectors.

We show the example of filling missing values for 2 dimensional word2vec vectors in Fig. 4.

### LSTM with fully connected neural network model

For training the models, we divide the data into 3 datasets including: training data, validation data, and testing data. At first, we use all of the words in the CHIEFCOMP column of the dataset to train the word2vec model, then we divide the dataset into two datasets: 80% training and validation data and 20% testing data.

In the next step, we find MI of all words in the training and validation dataset and then cut out the words that have low MI (bottom 5%) and cut out words with frequency less than 2. Next, we solve the missing values problem, and then use the training and validation dataset to train LSTM with the fully connected neural network model, by dividing the training and validation dataset into 80% training and 20% validation data. We show the conceptual framework for our research in Fig. 5. The softmax function in Eq. (7) is used as an activation function for the last layer of the classification model to compute probability of each record in each class where $y = [y_1, y_2, \ldots, y_d\ ]$ is a vector of real number.

$$\text{softmax}(y_i) = \frac{y_i}{\sum_{j=1}^{d} \exp\left(y_j\right)} \tag{7}$$

## Results and discussion

### Performance measurement

Since the dataset in this research is an imbalanced dataset, we cannot use accuracy to measure the performance of the model. For this reason we use G-mean (geometric mean of recall) [28] for measurement of the performance models. G-mean is defined in Eq. (9) where d is the number of classes, recall(class $c_i$) is a recall of class $c_i$ defined in Eq. (8).

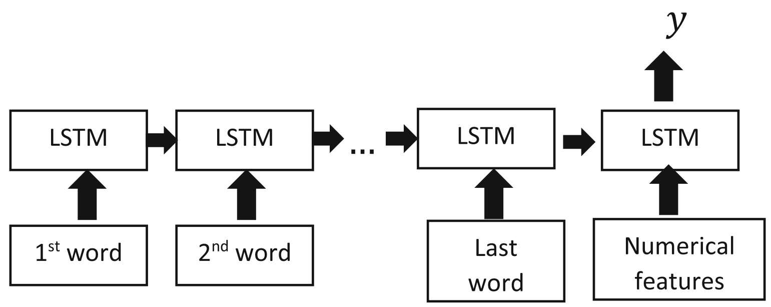

**Fig. 6** LSTM model structure

**Table 6** The time of data preprocessing + models training and testing of each model (second). Run on data science server at Chiang Mai University, Thailand (LINUX VPS, RAM 16 GB, CPU INTEL CORE i9, GPU 2080TI 11GB)

| | **Dataset** | **dengue + cold** | | | **Flu + dengue + cold** | | | **Flu + cold** | | | **SMS Spam Collection Dataset** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Model Architecture / Filing missing** | **LSTM** | **LSTM with numerical** | **LSTM + FNN** | **LSTM** | **LSTM with numerical** | **LSTM + FNN** | **LSTM** | **LSTM with numerical** | **LSTM + FNN** | **LSTM** |
| original | | 13.3 | 17.7 | 8.6 | 11.9 | 15.7 | 10.1 | 26.3 | 27.6 | 12.7 | 98.5 |
| MG | | 11.6 | 17.1 | 8.6 | 10.7 | 14.0 | 10.1 | 26.2 | 27.8 | 18.4 | 85.5 |
| keywords selection (cut words: frequency < 2) | cubic interpolation | 12.4 | 12.3 | 12.9 | 55.2 | 47.5 | 10.9 | 24.0 | 24.7 | 20.6 | 114.2 |
| | cut | 13.4 | 17.6 | 8.6 | 10.9 | 17.1 | 11.7 | 26.3 | 27.6 | 12.5 | 107.7 |
| | mean | 14.3 | 16.7 | 8.5 | 11.6 | 17.8 | 12.0 | 23.6 | 16.5 | 8.0 | 101.5 |
| keywords selection (cut words: MI bottom 5%) | cubic interpolation | 22.5 | 14.5 | 12.3 | 48.9 | 29.3 | 10.9 | 24.1 | 25.2 | 21.5 | 171.0 |
| | cut | 22.9 | 15.6 | 11.5 | 12.5 | 19.3 | 10.2 | 24.9 | 25.7 | 12.8 | 111.0 |
| | mean | 25.2 | 12.2 | 10.9 | 16.8 | 22.8 | 10.8 | 19.9 | 20.0 | 8.5 | 116.6 |
| keywords selection (cut words: MI bottom 5% or frequency < 2) | cubic interpolation | 13.4 | 14.7 | 13.0 | 49.0 | 29.3 | 11.0 | 24.8 | 25.3 | 20.0 | 111.6 |
| | cut | 26.4 | 20.3 | 11.6 | 12.7 | 22.4 | 10.3 | 25.1 | 26.1 | 12.7 | 145.2 |
| | mean | 25.1 | 12.6 | 10.7 | 22.1 | 22.2 | 10.8 | 21.7 | 19.9 | 8.1 | 105.3 |
| keywords selection (cut words: frequency < 2) + MG | cubic interpolation | 11.2 | 12.2 | 12.2 | 55.3 | 47.2 | 11.0 | 22.3 | 16.5 | 22.1 | 134.7 |
| | cut | 11.5 | 16.9 | 8.7 | 10.6 | 13.8 | 10.2 | 26.6 | 27.6 | 18.1 | 87.5 |
| | mean | 14.7 | 14.3 | 8.5 | 14.2 | 19.2 | 12.5 | 24.1 | 14.3 | 8.0 | 116.7 |
| keywords selection (cut words: MI bottom 5%) + MG | cubic interpolation | 23.9 | 15.1 | 11.1 | 27.6 | 28.9 | 10.9 | 24.0 | 25.3 | 20.0 | 206.7 |
| | cut | 18.3 | 15.7 | 8.5 | 12.2 | 16.4 | 10.2 | 24.8 | 29.0 | 15.6 | 219.0 |
| | mean | 25.0 | 11.8 | 10.2 | 16.6 | 20.6 | 11.1 | 15.4 | 16.7 | 15.4 | 116.8 |
| keywords selection (cut words: MI bottom 5% or frequency < 2) + MG | cubic interpolation | 13.4 | 13.6 | 11.0 | 36.0 | 29.5 | 10.8 | 23.9 | 25.4 | 18.4 | 147.3 |
| | cut | 25.2 | 15.4 | 8.6 | 12.7 | 16.4 | 10.2 | 25.0 | 29.1 | 15.4 | 141.9 |
| | mean | 19.7 | 11.8 | 8.5 | 15.8 | 20.6 | 10.9 | 21.6 | 17.0 | 8.0 | 80.2 |

$$\text{recall}(\text{class } c_i) = \frac{\text{number of records in class } c_i \text{ that true classification}}{\text{number of all records in class } c_i} \tag{8}$$

$$\text{G–mean} = \sqrt[d]{\prod_{i=1}^{d} \text{recall}(\text{class } c_i)} \tag{9}$$

### Performance of model

We have shown the performance of all models in Tables 4 and 5. Label-indicator morpheme growth (MG) [10] is the method that adds weight to the keywords with the highest MI (top 5%). SMS spam dataset is the basic dataset for text classification [23]. The model used in this research, was single layer LSTM and single hidden layer neural network (5 hidden nodes) with Adam optimizer in python library "keras".

For the LSTM model, we use LSTM with no hidden layer and LSTM with single hidden layer (the size of the vector in the hidden layer is 5) for performance comparison. In addition, for the single hidden layer fully connected neural network model, we ran the number of hidden nodes as 5, 10, 15, and 20. Moreover, for the word2vec model, we ran the size of the vector as 20, 25, and 30.

We considered our dataset in three ways, two of which are binary classes. It consists of 1) the common cold and dengue class, 2) the cold and flu class, and the other dataset is the multiple class (common cold, dengue and influenza class). For the SMS spam collection dataset, which is a standard dataset used to test the performance of our method. It consists of two classes, include ham and spam message.

In addition to use the LSTM and LSTM with a fully connected neural network. We also used the LSTM model with numerical features as shown in Fig. 6. to compare the model's performance.

The results showed that LSTM with a fully connected neural network had better performance than normal LSTM. Moreover, removing stop words increased the G-mean value of the testing data for all datasets. For the medical records dataset, LSTM with Fully connected Neural network gives the best G-mean value when words with low MI (bottom 5%) and low frequency (frequency < 2) together are considered stop words. If we set the stop words to be the words with low frequency (frequency < 2), then it reduces the training time (shown in Table 6) and increases the performance of the LSTM model. Moreover, LSTM with the feed forward fully connected neural network model uses less time for training than the LSTM model, because it has a faster convergence.

## Conclusion

This research used the LSTM model with fully connected neural network for dengue fever and influenza detection. Text of symptoms and other features including age, body temperature, gender, and month of service were used for input data. The results showed that the LSTM with the fully connected neural network model had higher performance than the normal LSTM model. In addition, removing unimportant keywords from the dataset and also increased their performance.

**Acknowledgments**

This work was supported in part by the Center of Excellence in Community Health Informatics and Fundamental Fund 2022, Chiang Mai University; in part by NSRF via the Program Management Unit for Human Resources & Institutional Development, Research and Innovation [Grant Number B05F640183].

### Authors' contributions

WN designed and performed the literature review and lead the writing of the paper. WB contributed to the design of the literature review, wrote parts of the paper and gained ethical approval. EB contributed to the design of the literature review, wrote parts of the paper and reviewed the manuscript. All authors reviewed and edited the manuscript and approved the final version of the manuscript.

### Funding

Funding for this work was provided by Center of Excellence in Community Health Informatics and Fundamental Fund 2022, Chiang Mai University; in part by NSRF via the Program Management Unit for Human Resources & Institutional Development, Research and Innovation [Grant Number B05F640183].

### Availability of data and materials

Both programming code (python) and data are available upon request (ekkarat.boonchieng@cmu.ac.th).

## Declarations

### Ethics approval and consent to participate

Ethical approval was obtained from Faculty of Public Health, Chiang Mai University.
(Approval number ET036/2564)

### Consent for publication

Not applicable.

### Competing interests

The authors declare that they have no competing interests.

### Author details

[1]Department of Computer Science, Faculty of Science, Chang Mai University, Chiang Mai 50200, Thailand. [2]Faculty of Public Health, Chiang Mai University, Chiang Mai 50200, Thailand. [3]Center of Excellence in Community Health Informatics, Department of Computer Science, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand.

Received: 16 May 2021 Accepted: 23 January 2022
Published online: 14 February 2022

### References

1. Amin S, Uddin MI, Hassan S, Khan A, Nasser N, Alharbi A, et al. Recurrent neural networks with TF-IDF embedding technique for detection and classification in tweets of dengue disease. IEEE Access. 2020;8:131522–33. https://doi.org/10.1109/ACCESS.2020.3009058e.
2. Atkinson K. INTERPOLATION. 2003. http://homepage.math.uiowa.edu/~atkinson/ftp/ENA_Materials/Overheads/sec_4-1.pdf.
3. Boonchieng E, Boonchieng W, Senaratana W, Singkaew J. Development of mHealth for public health information collection, with GIS, using private cloud: A case study of Saraphi district, Chiang Mai, Thailand. In: 2014 International Computer Science and Engineering Conference (ICSEC); 2014. p. 350–3. https://doi.org/10.1109/ICSEC.2014.6978221.
4. Boonchieng W, Boonchieng E, Tuanrat WC, Khuntichot C, Duangchaemkarn K. Integrative system of virtual electronic health record with online community-based health determinant data for home care service: MHealth development and usability test. IEEE Healthc Innov Point Care Technol (HI-POCT). 2017;2017:5–8. https://doi.org/10.1109/HIC.2017.8227571.
5. Boonchieng W, Chaiwan J, Shrestha B, Shrestha M, Dede AJO, Boonchieng E. mHealth technology translation in a limited resources community—process, challenges, and lessons learned from a limited resources Community of Chiang Mai Province, Thailand. IEEE J Transl Eng Health Med. 2021;9:1–8. https://doi.org/10.1109/JTEHM.2021.3055069.
6. Briyatis SHU, Premaratne SC, De Silva DGH. A novel method for dengue management based on vital signs and blood profile. Int J Eng Adv Technol. 2019;8(6 special issue 3):154–9. https://doi.org/10.35940/ijeat.F1025.0986S319.
7. Chen CW, Tseng SP, Kuan TW, Wang JF. Outpatient text classification using attention-based bidirectional LSTM for robot-assisted servicing in hospital. Information (Switzerland). 2020;11(2):106. https://doi.org/10.3390/info11020106.
8. Fu B, Yang Y, Ma Y, Hao J, Chen S, Liu S, et al. Attention-based recurrent Multi-Channel neural network for influenza epidemic prediction. In: Proceedings - 2018 IEEE international conference on bioinformatics and biomedicine, BIBM 2018; 2018. p. 1245–8. https://doi.org/10.1109/BIBM.2018.8621467.
9. Gers FA, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. Neural Comput. 2000;12(10): 2451–71. https://doi.org/10.1162/089976600300015015.
10. Hu Y, Wen G, Ma J, Li D, Wang C, Li H, et al. Label-indicator morpheme growth on LSTM for Chinese healthcare question department classification. J Biomed Inform. 2018;82:154–68. https://doi.org/10.1016/j.jbi.2018.04.011.
11. Karim F, Majumdar S, Darabi H, Chen S. LSTM fully convolutional networks for time series classification. IEEE Access. 2017;6:1662–9. https://doi.org/10.1109/ACCESS.2017.2779939.
12. Lee SH, Levin D, Finley PD, Heilig CM. Chief complaint classification with recurrent neural networks. J Biomed Inform. 2019;93:103158. https://doi.org/10.1016/j.jbi.2019.103158.
13. Long F, Zhou K, Ou W. Sentiment analysis of text based on bidirectional LSTM with multi-head attention. IEEE Access. 2019;7:141960–9. https://doi.org/10.1109/ACCESS.2019.2942614.
14. Gensim: Topic modeling for humans. 2019. https://radimrehurek.com/gensim/.
15. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings; 2013. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85083951332&partnerID=40&md5=20428820e8b09cdfb5078ea812a71f2d.

16. Murhekar M, Joshua V, Kanagasabai K, Shete V, Ravi M, Ramachandran R, et al. Epidemiology of dengue fever in India, based on laboratory surveillance data, 2014–2017. Int J Infect Dis. 2019;84:S10–4. https://doi.org/10.1016/j.ijid.2019.01.004.
17. Nadda W, Boonchieng W, Boonchieng E. Dengue fever detection using Long short-term memory neural network. In: 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2020; 2020. p. 755–8. https://doi.org/10.1109/ECTI-CON49241.2020.9158315.
18. Nadda W, Boonchieng W, Boonchieng E. Weighted extreme learning machine for dengue detection with class-imbalance classification. In: 2019 IEEE Healthcare Innovations and Point of Care Technologies, (HI-POCT); 2019. p. 151–4. https://doi.org/10.1109/HI-POCT45284.2019.8962825.
19. Petmezas G, Haris K, Stefanopoulos L, Kilintzis V, Tzavelis A, Rogers JA, et al. Automated Atrial Fibrillation Detection using a Hybrid CNN-LSTM Network on Imbalanced ECG Datasets. In: Biomedical Signal Processing and Control; 2021. p. 63. https://doi.org/10.1016/j.bspc.2020.102194.
20. PyThaiNLP. 2020. https://github.com/PyThaiNLP/pythainlp
21. Rangarajan P, Mody SK, Marathe M. Forecasting dengue and influenza incidences using a sparse representation of Google trends, electronic health records, and time series data. PLoS Comput Biol. 2019;15(11):e1007518. https://doi.org/10.1371/journal.pcbi.1007518.
22. Rotejanaprasert C, Ekapirat N, Areechokchai D, Maude RJ. Bayesian spatiotemporal modeling with sliding windows to correct reporting delays for real-time dengue surveillance in Thailand. Int J Health Geogr. 2020;19(1):4. https://doi.org/10.1186/s12942-020-00199-0.
23. SMS Spam Collection Dataset. 2016. https://www.kaggle.com/uciml/sms-spam-collection-dataset.
24. Tran D, Mac H, Tong V, Tran HA, Nguyen LG. A LSTM based framework for handling multiclass imbalance in DGA botnet detection. Neurocomputing. 2018;275:2401–13. https://doi.org/10.1016/j.neucom.2017.11.018.
25. Venna SR, Tavanaei A, Gottumukkala RN, Raghavan VV, Maida AS, Nichols S. A novel data-driven model for real-time influenza forecasting. IEEE Access. 2019;7:7691–701. https://doi.org/10.1109/ACCESS.2018.2888585.
26. Xiao JP, He JF, Deng AP, Lin HL, Song T, Peng ZQ, et al. Characterizing a large outbreak of dengue fever in Guangdong Province, China. Infect Dis Poverty. 2016;5(1):44. https://doi.org/10.1186/s40249-016-0131-z.
27. Zhao S, Cai Z, Chen H, Wang Y, Liu F, Liu A. Adversarial training based lattice LSTM for Chinese clinical named entity recognition. J Biomed Inf. 2019;99:103290. https://doi.org/10.1016/j.jbi.2019.103290.
28. Zong W, Huang GB, Chen Y. Weighted extreme learning machine for imbalance learning. Neurocomputing. 2013;101:229–42. https://doi.org/10.1016/j.neucom.2012.08.010.

## Publisher's Note

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