Data Science and Engineering (DSE) Record, Volume 5, issue 1.

# Development of Deep Learning Techniques to Classify User Concern for Food Delivery Application

Nathakit Keawtoomla [1,4] and Arinya Pongwat [2] and Jakramate Bootkrajang [3] and Haruhiko Takase [4]

[1] Master's Degree Program in Data Science, Chiang Mai University, Chiang Mai, Thailand
[2] College of Arts, Media and Technology, Chiang Mai University, Chiang Mai, Thailand
[3] Department of Computer Science, Faculty of Science, Chiang Mai University, Chiang Mai, Thailand
[4] Computer Engineering Laboratory, Department of Electrical and Electronic Engineering, Faculty of Engineering, Mie University, Tsu, Mie, Japan

Nathakit_keawtoom@cmu.ac.th

**Abstract.** With the rapid growth of the food delivery industry, there is an urgent need to manage software effectively for sharing economy applications. One way to evaluate the effectiveness of these applications is by examining user concerns and feedback. We propose to use a Bi-LSTM-CNN model in a pipeline for automatic classification of the user concerns. The performances of other machine learning and deep learning models were studied and compared. The results showed that the proposed Bi-LSTM-CNN model achieved the highest accuracy score of 84.6%, outperforming the single deep learning models and the traditional machine learning models. Moreover, due to the imbalance nature of the collected data, the impact of data oversampling technique for data imbalance problem was also evaluated. Interestingly, the interplays between the complex representation induced by the proposed Bi-LSTM-CNN model render the selected oversampling scheme e.g., SMOTE, unnecessary for our setting.

**Keywords:** food delivery; food delivery applications; text classification; deep learning; CNN; Bi-LSTM, Bi-LSTM-CNN

## 1 Introduction

During COVID-19 pandemic, millions of people globally always had to practice social distancing and avoid face-to-face interactions. This crisis has affected consumer and corporate activities that influenced changes in consumer behaviour (Guthrie et al., 2021). Most consumers adapt to learn new behaviours in online purchases. For example, instead of using cash, people are getting cashless payments to purchase goods and services (Muangmee et al., 2021). Therefore, online transactions increased rapidly, especially in e-commerce platforms.

In the same way, online-to-offline mobile food delivery services provided a new alternative for restaurants to manage during the pandemic crisis of COVID-19 (Mehrolia et al., 2021). Food services use online platforms to provide convenience for customers to order food through mobile applications (Pigatto et al., 2017). The

emergence of Food Delivery Applications (FDAs) has an impact on society's lifestyle and creates economic growth (Sjahroeddin, 2018). In the United Kingdom, the number of users reaches nearly 6 million downloads and contributed to $6.7 billion of revenue in 2021 (Curry, 2022); Statista, 2021).

Nowadays, the distribution of mobile application development expands rapidly in the software industry. With intense competition in this business, stakeholders should be concerned about issues that occurred with their software (Islam et al., 2010). Previous research reported that usability is the main factor influencing consumers to use applications continuously (Hoehle, and Venkatesh, 2015). App platforms such as Google Play Store and App Store permitted users to give feedback in the reviews. User feedback is a valuable source for developers to comprehend what features users desire or what issues are found in their applications (Ciurumelea et al., 2017); Pagano, and Maalej, 2013). User concerns classification is very critical to identify any problems in applications and maintain stability performance to retain users' intention to use mobile applications (Ciurumelea, Schaufelbühl, Panichella, and Gall, 2017).

In addition, the framework of modelling user concerns in a case study of food delivery applications has been proposed by Williams et al. (2020). This previous study used machine learning for the automatic detection of user concerns. Machine learning has many disadvantages such as dimension explosion and data sparsity. Deep learning, on the other hand, has the additional advantage of the semantic relationship between words (Wu et al., 2020). Although several previous works have explored the food delivery evaluation before, fewer studies were developed by approaching new food delivery application techniques to handle the limitations of traditional text classification.

The contribution of our study can be summarised as follows:

- This study proposed an automatic text classification using deep learning models including, convolutional neural networks (CNN), bidirectional long short-term memory (Bi-LSTM) and Bi-LSTM-CNN hybrid models.
- We compared the performances between deep learning methods and traditional machine learning models to find the state-of-the-art model for food delivery apps.
- The proposed architecture can be adapted for sharing economy applications to understand and evaluate user feedback.

The rest of the paper is organised as follows. Section 2 gives the background of research papers which are related to the present study. Section 3 provides the research methods. Section 4 discusses the results of an experiment. Section 5 concludes our study and describes future directions of work.

# 2 Literature Review

## 2.1 Concept of Food Delivery Applications

The rapid rise of Peer-to-Peer markets in the sharing economy provides a new way to consume goods and services (Wirtz et al., 2019). These services used software platforms as an intermediary to facilitate sharing of their resources between private individuals (Allen, 2015). Sharing economy system succeeded in collaborative consumption by using networks that bring platforms to merge consumer demand and services (Sutherland, and Jarrahi, 2018). Digital technology promotes new services which are easily handled on online platforms via the push of a button and payment on mobile devices (Allen, 2017). Sharing economy technology is transforming the new way of making relationships between people by connecting them to become social groups via digital platforms (Sutherland, and Jarrahi, 2018).

Food delivery applications (FDAs) are becoming popular in sharing economy system. In principle, FDAs are an online platform that allows customers to select the type of foods or restaurants with facile interaction platforms and convenient options (Pigatto, Machado, dos Santos Negreti, and Machado, 2017). The arising of FDAs entails intense competition among many start-ups. For example, Instacart has profited more than $2 billion since 2016 (Kung, and Zhong, 2017); Solomon, 2015). Similarly, Brazilians are increasingly using smartphones. People are beginning to use delivery platforms to consume food services. In 2014, HelloFood and iFood had high funding from international investors (Pigatto, Machado, dos Santos Negreti, and Machado, 2017).

Traditionally, Lee et al. (2017) used online questionnaires and structural equation modelling to understand the factors that affect customers' use of FDAs. They found that consumer reviews, restaurant information, system and design quality had influenced their intention to use applications continuously. In the same way, Ray et al. (2019) examined users' behavioural intentions in India. Indicating, that four factors, including, customer experience, the search of restaurants, listing, and ease of use significantly support their hypothesis. Moreover, Bao, and Zhu (2021) found that these factors, namely information quality, system quality and service quality are able to improve customer satisfaction and perceived value, which influenced the intention to reuse the FDAs.

## 2.2 User-Generated Content

The creation of the general public as User-Generated Content (UGC) in social media is turning into a beneficial source for the marketing communication (Daugherty et al., 2008). UGC provides the personality of consumers' attitudes and behaviours that influences customer decision-making for using services and products (Tsiakali, 2018). It is essential to understand the media content of consumer attitudes, which have a huge impact on marketing and media suppliers in both short and long-term products (Daugherty, Eastin, and Bright, 2008). The disseminative information from mobile community

applications can provide insight strategies that have advantages to increase user intention via users' reposts (Chen et al., 2019).

### 2.3 Text Classification with Deep Learning Model

#### 2.3.1 Word Embedding

Natural language processing (NLP) has been studied by several scholars. Recently, word embedding has succeeded in comprehending the relationship among words. Word embeddings are defined as a vector space representation of words that are able to preserve semantic properties or relative meaning between words by building real-valued vectors (Ghannay et al., 2016); Hindocha et al., 2019). Word embedding, which can reduce vector size to become low-dimensional vectors, is mainly useful in deep learning architectures (Liu et al., 2015); Roberts, 2016).

One of the most widely used word embeddings is Global Vectors for Word Representation (GloVe). This model is based on Log-bilinear Regression, combining Global Matrix Factorization and Local Context Window methods for learning word representations. GloVe is unsupervised learning that is trained by the aggregated global word-word co-occurrence statistics and represents an output as words in the form of vectors (Pennington et al., 2014). GloVe cost function: weighting function $f(X_{ij})$ shown as equation 1.

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T w_j + b_i + b_j - logX_{ij})^2 \tag{1}$$

Here, $V$ is denoted as vocabulary size and $w_i$ is the vector for the main word, $w_j$ is the vector for the context word, $b$ are the bias terms.

For the weighting function, Pennington, Socher, and Manning (2014) suggested that this function performed well.

$$f(X_{ij}) = \left\{ \begin{matrix} \left(\frac{x_{ij}}{x_{max}}\right)^{\alpha} & \text{if } X_{ij} < X_{max} \\ 1 & \text{otherwise} \end{matrix} \right\} \tag{2}$$

#### 2.3.2 Convolution Neural Network

The advance of deep learning methods has significantly impacted natural language processing areas. Traditionally, a Convolutional Neural Network (CNN) is highly successful in computer vision such as image processing and pattern recognition (Albawi et al., 2017); O'Shea, and Nash, 2015). CNN model is made up of three types of layers, including convolution layers, pooling layers and fully-connected layers.

In classification tasks, the CNN model is most useful for sentiment analysis and sentence classification (Wang, and Gang, 2018). This model can extract relevant information that can pass values into the next layers without losing semantics similarity between words in sentences. CNN is commonly well combined with word embedding (Chen, 2015). Kim (2014) found that one-dimensional CNN, which used pre-trained vectors, had excellent performance for sentence classification.

For the CNN model, $X_i \in \mathbb{R}^k$ indicates the k-dimensional word vector for the sentence's i-th word. Thus, the sentence of length n shows as equation 3, where $\oplus$ is the concatenation operator.

$$X = X_1 \oplus X_2 \oplus X_3 \oplus ... \oplus X_n \tag{3}$$

To extract a feature $c_i$, a filter $W_j \in \mathbb{R}^{m \times k}$ was applied to the word vector $X_{i:i+h}$ from a sliding window as equation 4, where $b$ is a bias term and $f$ is a non-linear function.

$$C_i = f\left(\sum_{j=1}^{h} W_j X_{i+j-1} + b\right) \tag{4}$$

The feature map is created by a filter which is applied to each potential window of the words vector $c \in \mathbb{R}^{n-h+1}$.

$$c = (c_1 + c_2 + c_3, \ldots, c_{n-h+1}) \tag{5}$$

Finally, max pooling is applied to calculate the maximum value for patches of a feature map.

$$z = max\{c\} \tag{6}$$

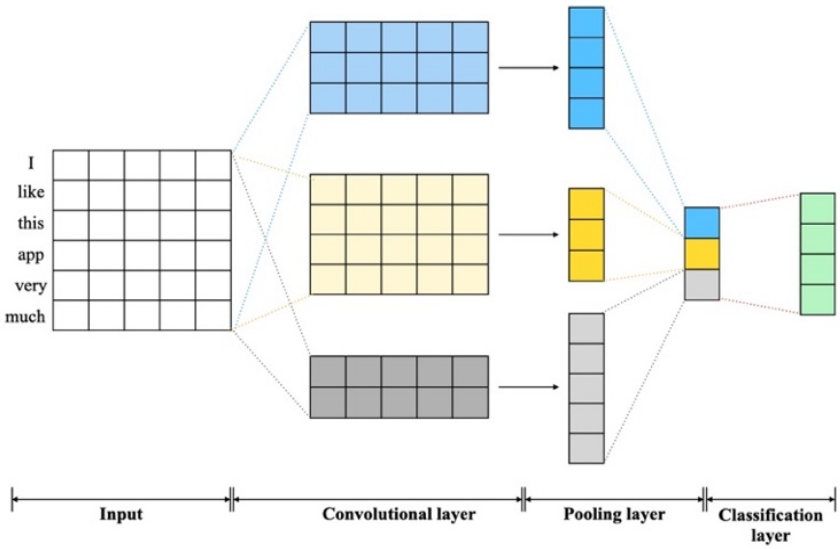



**Figure 1** Convolutional Neural Network Architecture

### 2.3.3 Bidirectional Long Short-Term Memory

Long Short-Term Memory (LSTM) has been proposed to solve the limitation of Recurrent Neutral Network (RNN) known as the vanishing gradient problem (Hochreiter, and Schmidhuber, 1997). LSTM has many short-term memory cells that link to building long-term memory. This model consists of three types of gates, including, the forgetting gate layer, the input gate layer and the output gate (Zhao et al., 2020). Thus, LSTM can learn long-term dependencies by using a forgetting mechanism to remove external context. The architecture of the LSTM model is provided in Figure 2.

The LSTM has a horizontal line at the top of the diagram (from $C_{t-1}$ to $C_t$). This line is called a cell state ($C_t$). It is like a conveyor belt that information can be removed via multiplication operation or added to memory through addition operation. The information was considered in the forgetting gate layer in the first step. The current input at the time *t* ($X_t$) and the previous output at the time *t-1* ($h_{t-1}$) are passed through the sigmoid layer (σ). The sigmoid function generates the output ($f_t$), which is a number between 0 to 1. Indicating, $f_t = 0$ represents that $C_{t-1}$ should be eliminated while $f_t = 1$ retains fully the state $C_{t-1}$

$$f_t = \sigma\left(W_f X_t + U_f h_{t-1} + b_f\right) \tag{7}$$

$$C_t = C_{t-1} \otimes f_t \tag{8}$$

where $W_f$ and $U_f$ are the weights, $b_f$ is the bias weight vector.
The next step has two processes. First, $X_t$ and $h_{t-1}$ run through the gate activation function (σ) in the input gate layer, and the new information was decided whether to be updated or not. Next, the input activation function (tanh) assigns weight to the values that pass through to the state. After that, the new cell state, which is added to the cell memory line, equals the sum of old memories with new memories.

$$i_t = \sigma\left(W_i X_t + U_i h_{t-1} + b_i\right) \tag{9}$$

$$a_t = tanh\left(W_a X_t + U_a h_{t-1} + b_a\right) \tag{10}$$

$$C_t = C_{t-1} \otimes f_t + i_t \otimes a_t \tag{11}$$

In the output gate, the sigmoid layer decides what parts of the cell state should be the output. Then, the new cell state is passed through the tanh function. These two results are multiplied one by one. The output layer produces prediction and sends info back into the node in the next time steps. Due to the fact that the limitation of the LSTM model is time inefficient, this model takes longer to train than other deep learning models (Li et al., 2017).

$$O_t = \sigma\left(W_o X_t + U_o h_{t-1} + b_o\right) \tag{12}$$

$$h_t = tanh\,(C_t) \otimes O_t \tag{13}$$

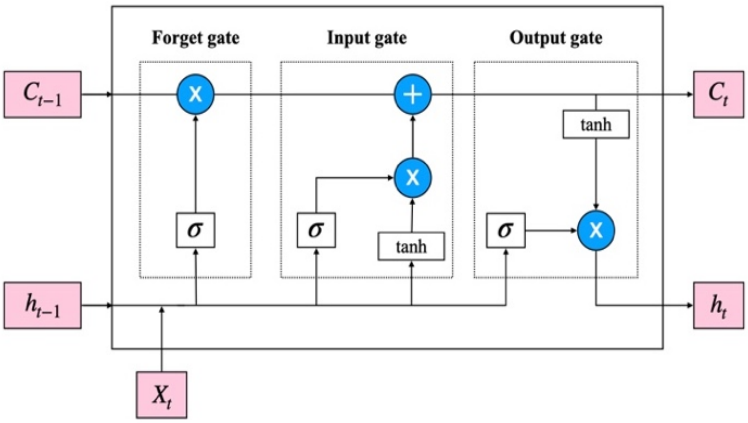



**Figure 2** Long Short-Term Memory Architecture

Bidirectional Long Short-Term Memory (Bi-LSTM), which is one type of LSTM model, is a neural network that has two-way contextual information. Both sides of the Bi-LSTM model are effectively able to learn long-term dependencies which keep the semantics of sentences and characteristics of phrases (Jang et al., 2020); Liang, and Zhang, 2016); Zhao, Zhang, Yuan, Liu, Shan, and Zhang, 2020). The architecture of Bi-LSTM provides in Figure 3. In Bi-LSTM, the input sequence is calculated in the forward direction $\vec{h}$ and the backward direction $\overleftarrow{h}$. Lastly, the output is created by both $\vec{h}$ and $\overleftarrow{h}$.

$$\overrightarrow{h}_t = \phi(W_f X_f + \overrightarrow{h}_{t-1} W_f + b_f) \tag{14}$$

$$\overleftarrow{h}_t = \phi(W_b X_b + \overleftarrow{h}_{t+1} W_b + b_b) \tag{15}$$

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{16}$$

$$O_t = h_t W_{hq} + b_q \tag{17}$$

where $W_{hq}$ is the weight matrix, $b_q$ is the bias and $\phi$ denotes the hidden layer activation function.

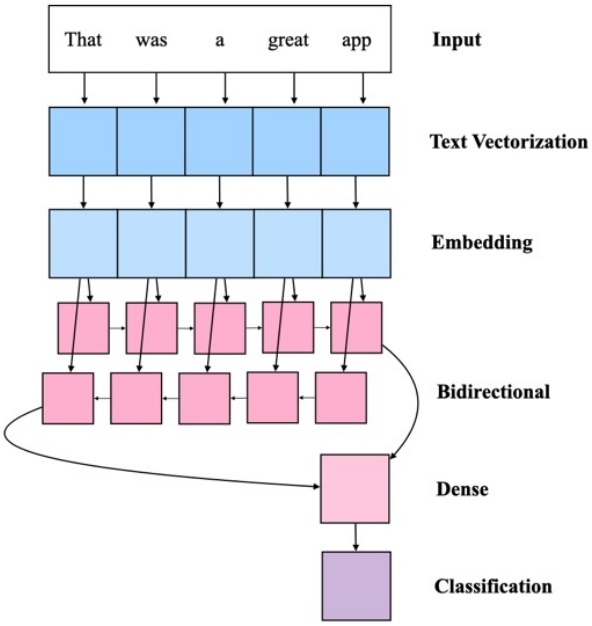



**Figure 3** Bidirectional Long Short-Term Memory Architecture

### 2.3.4 Bi-LSTM-CNN hybrid model

The emergence of deep learning which consists of convolution neural networks and bidirectional long short-term memory models has been reported in several research. Senthil Kumar, and Malarvizhi (2020) used a Bi-LSTM-CNN combined model for classifying customer opinions in social media. As the result, the Bi-LSTM-CNN algorithm gives the best accuracy than individual deep learning models. Similarly, Bhuvaneshwari et al. (2022) presented the Bi-LSTM Self Attention-based Convolutional Neural Network (BAC) model to classify emotion polarity in online reviews. They found that the hybrid models can better capture the semantic sentence in text sequences which results in better performance than other baseline models.

In principle, the CNN model combines three context vectors as presented in equation 18: left context vector ($\overrightarrow{h_t}(X_t)$), right context vector ($\overleftarrow{h_t}(X_t)$), and current word's word vector ($h_{t-1}$). Indicating, the semantics of the sentence will be more accurate and can drop the ambiguity of the $X_t$ word.

$$X_t = (\overrightarrow{h}_t(X_t), h_{t-1}, \overleftarrow{h}_t(X_t)) \tag{18}$$

Next, the tanh activation is applied as in equation 19 and sent the output to the max-pooling layer, to convert the length of text into same-length vectors.

$$y_t = tanh(W_t X_t + b_t) \tag{19}$$

Finally, the normalized exponential function or the softmax function converts a vector into a probability distribution as in equation 20.

$$p_t = \frac{exp(y_t)}{\sum_{k=1}^{n} exp(y_k)} \tag{20}$$

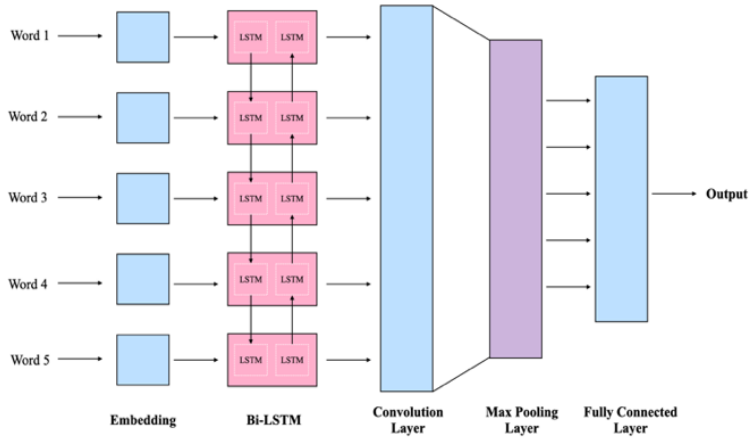



**Figure 4** Bi-LSTM-CNN Architecture

# 3 Data and Methodology

In this study, we presented a conceptual framework for the classification of user concerns from food delivery applications, as depicted in Figure 5. The provided diagram involves collecting and cleaning text data, tokenising words, converting them into numerical vectors, and using TF-IDF for weighting. The sentiment analysis model is trained on labelled data to identify sentiments (positive, negative, or neutral) in reviews. The model's accuracy is evaluated, and adjustments are made to improve performance. Ultimately, the trained model is deployed for real-world applications, such as analysing and understanding user concerns to enhance service based on feedback.

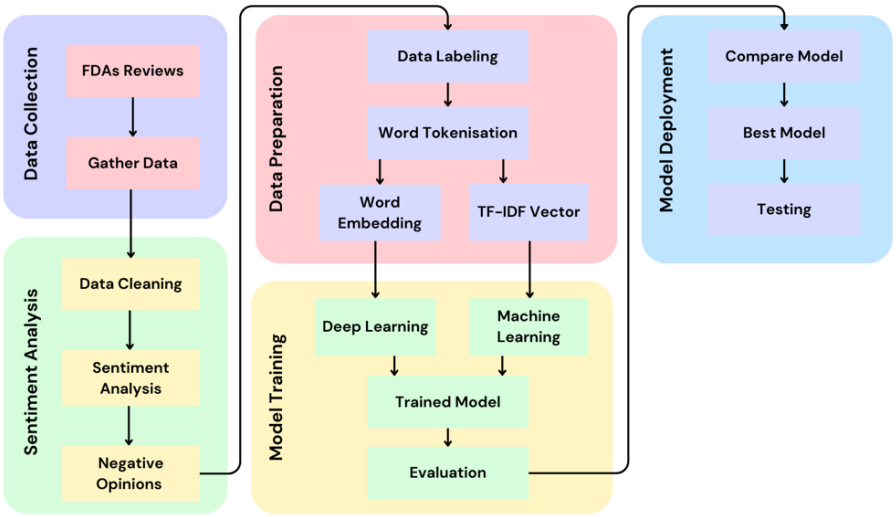



**Figure 6** Conceptual framework in research

## 3.1 Data Collection

The data set of food delivery apps for this research was obtained from AppFollow (https://appfollow.io/). AppFollow is an app monitoring platform that can help to gather enormous comment reviews from App Store and Google Play. We selected four popular food delivery apps in the United Kingdom, including Deliveroo, Foodhub, Grabhub and Justeat. The data set was collected between 1 May 2021 to 1 August 2021. The number of user feedback on the App Store is 4,504 reviews while Google Play is 16,131 reviews. The English language reviews from all platforms were combined into a single data set. Table 1 summarises the details of food delivery apps in each platform.

**Table 1** The details of the data set

| App name | Google Play | App Store | Total reviews |
|---|---|---|---|
| Deliveroo | 4,511 | 1,210 | 5,721 |
| Foodhub | 86 | 495 | 581 |
| Grabhub | 8,272 | 1,987 | 10,239 |
| Justeat | 3,262 | 832 | 4,094 |

### 3.2 Data Preprocessing

For Natural Language Processing tasks, cleaning or preprocessing text data is essential before building the model. We used the Natural Language Toolkit (NLTK) to transform raw data into a format that can be comprehended and analyzed by deep learning. NLTK is an open-source license that suitable for text processing. NLTK consists of many modules and corpora which are easy to use for conducting NLP research (Bird, and Loper, 2004); Madnani, 2007).
The steps can be summarised as follows:

- Splitting sentences: break up or split a complex sentence into two or more sentences. For instance, “Fast and with a lot of options. I wish it had more coupons but it’s good either way.” 1) “Fast and with a lot of options.” and 2) “I wish it had more coupons but it's good either way.”
- Removing punctuations: There are some punctuations in sentences, for example, “,”, “;”, “?” etc. This process will help to treat each text equally.
- Removing stop words: removal of stop words such as “the”, “a”, “an” etc. can reduce dataset size and the training time during the training model.
- Removing non-ASCII characters: Replace non-ASCII characters with ASCII by removing accents and remaining non-ASCII characters. For instance, convert “á” to “a”.
- Text lower case: change the capitalisation or upper case of text to lower case e.g., “Food” to “food”.
- Text lemmatization: convert a word to its base form such as “paid” to “pay”.
- Text tokenisation: breaking the text into words when we split the text into sentences. For example, “awful customer service” to “awful”, “customer”, “service”.

### 3.3 Data Filtering

In this step, we identified the polarity of sentiment in each sentence that users expressed emotions during using the food delivery application. To automatically classify sentences, Valence Aware Dictionary and sEntiment Reasoner (VADER) was used to determine whether the users’ opinions are positive, negative, or neutral. VADER is a lexicon-based approach which can classify sentiment without requiring training dataset (Elbagir, and Yang, 2019). In the VADER method, sentiment scores consist of four polarities, including, positive, negative, neutral and compound polarity. Both positive and negative polarity have normalized between 0 and 1 while the compound polarity,

which is a summation of positive, negative, and neutral scores, has normalized between -1 (negative) and 1 (positive).

After that, we chose 3,600 negative sentences randomly and considered only relevant sentences regarding user concerns. The sentences were manually labelled for text classification, following by Williams, Tushev, Ebrahimi, and Mahmoud (2020) guideline. The reference provided a definition of user concerns in Table 2.

We employed three domain experts, including two software engineering professionals and one graduate student from the Data Science Consortium, to manually label sentences. Finally, the user concerns data set contains 2,715 sentences, of which 1,194 are human class, 708 are bug reports class, 544 are market class and 269 are feature request class. The bar chart in Figure 6 provides information about the number of user concerns classes in the data set.

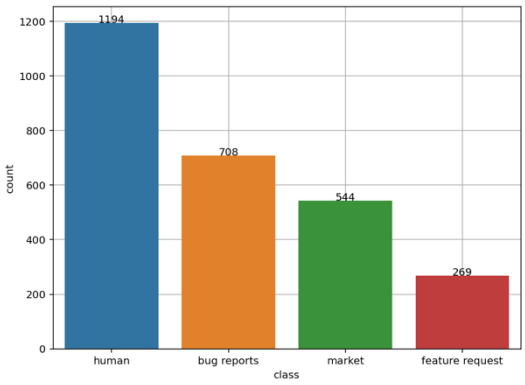



**Figure 5** The number of each target attributes

**Table 2** The user concern definition in food delivery apps

| Class | Definition | Review |
|---|---|---|
| **Human** | The interactions between drivers, customer service or restaurants, and users during using this service. Users are dissatisfied with late orders, orders missing, cancelling orders and drivers lost. | "It's sad that I always order food and it's always missing or they deliver to the wrong address." "Drivers get very annoyed and that is with me like its my fault." |
| **Market** | Users mentioned the service charge or extra fees were high for delivery | "Uber eats will not refund your money or we place the order if you are |

| | | |
|---|---|---|
| | service. Others are customer service policies such as refunds, promotions, and delivery zones. | missing items out of your order."<br>"There are hidden charges which make the check out amount more than what we expect from the promo code." |
| **Bug reports** | The software is an error, flaw and incorrect result or causes the software to behave erratically. | "It's stuck at the location page and just keeps spinning."<br>"Same issue as everyone here, the app gets stuck at the location selection." |
| **Feature requests** | The user suggestions to modify and improve apps or ask to add new functionality and feature. | "Really need to be able to cancel an order when no one picking it up"<br>"No option to contact deliveroo or driver or to get missing items." |

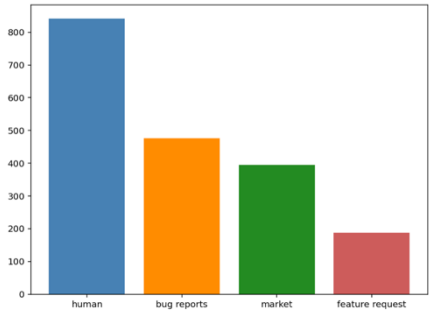
Finally, the user concerns data set contains 2,715 sentences, of which 1,194 are human class, 708 are bug reports class, 544 are market class and 269 are feature request class. The bar chart in Figure 6 provides information about the number of user concerns classes in the data set.

### 3.4 Data Resampling Technique

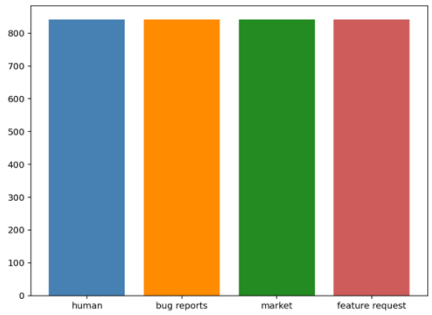
Before creating models, data is divided into two subsets. One is a training data set used to train and develop models. The other is the test data set used after the training is done. In this experiment, data is split at a 70-30 ratio of training versus testing data. There are 1,900 samples in the training data set and the number of examples in each class is rather imbalanced with 44.26% majority class more than other minority classes as shown in Figure 7(a). Typically, the minority class is very essential for investigation in multi-class classification. We used Synthetic Minority Oversampling Technique (SMOTE) to increase the number of samples of the minority class. SMOTE, the oversampling technique, is capable of handling the overfitting problem (Koto, 2014); Rupapara et al., 2021). Several previous research has approached this technique to

increase classification accuracy. Rupapara, Rustam, Shahzad, Mehmood, Ashraf, and Choi (2021) suggest that balancing the data can reduce the overfitting problem which also happens when training a model. Xu et al. (2015) proposed word embedding composition for text classification. The result showed that SMOTE algorithm was effective in handling data imbalance in classification tasks. Although SMOTE technique can handle binary-class problems effectively, it was reported that the technique yielded poor results in the multi-class problem (Danuri et al., 2022). In addition, Padurariu, and Breaban (2019) pointed out that a more complex embedding like Glove does not work well in small sets with large imbalanced data.

In this study, we try to apply SMOTE to generate the minority class (feature request, market, bug reports) to obtain a class balanced in the training data set. The number of each class in the training data set is shown in Figure 7(b).




**A) Bar graph of examples in each class in training data**




**B) Bar graph of examples in training data after SMOTE oversampling**

**Figure 6** Bar graph of examples in each class in training data

### 3.5 Machine Learning Models

In this section, we created several machine learning models, including Random Forest (RF), Decision Tree (DT) and K-Neighbor Nearest (KNN). All these traditional models are used to classify user concerns as human, bug reports, markets, or feature requests, based on the sentence of reviews. For creating a model, hyperparameter tuning is essential so that the model can improve the performance for predicting the data (Yang, and Shami, 2020). A grid search was employed to optimize the model to find

the best hyperparameter. However, the training data process will cause overfitting that the model gives a good performance in training data but gives a low performance in test data. Thus, cross-validation is applied to tune the hyper-parameter in the grid-search process (Sumathi, 2020). During the model fitting, we separated the training set into 5-fold cross-validation and the parameters in each model were set as follows:

1) Decision Tree: the function to measure the quality of a split (criterion) is "gini", the splitter used to choose the split at each node (splitter) is "random" and the maximum depth of the tree (max_depth) is 20.
2) K-Nearest Neighbors: The number of neighbours to use by default for kneighbors queries (n_neighbors) is 4, the weight function used in prediction (weights) is "distance" and the algorithm used to compute the nearest (algorithm) is "auto"
3) Random Forest: the number of trees in the forest (n_estimators) is 91, the function to measure the quality of a split (criterion) is "entropy" and the maximum depth of the tree (max_depth) is 30.

### 3.6 Deep Learning Models

In the next process, we built deep learning models to classify the aspect categories. The models are CNN, Bi-LSTM and Bi-LSTM-CNN models. The architecture of the CNN model is provided in Figure 8. The input layer takes in reviews in the form of feature vectors from NLTK. A Glove embedding layer then learns word embeddings into 300-dimensional space. The dense vectors are then fed into a 1-D convolutional layer that contains 64 filters, and 9 kernel sizes with Rectified linear unit activation function (ReLU). After that, the input is taken by a global max pooling 1D to retain only the maximum value. Then, the preceding layer connected the two dense layers with ReLU activation function. Lastly, the softmax function in the output layer calculated the prediction of the target variable.

The Bi-LSTM model has an architecture that is slightly different from the CNN model. The Bi-LSTM model utilizes the dropout layers to reduce overfitting. In the same way, the Bi-LSTM-CNN hybrid model uses Bi-LSTM architecture to capture the semantics of sentences in both left and right directions, and then pass the output to the CNN model for predicting the targets.

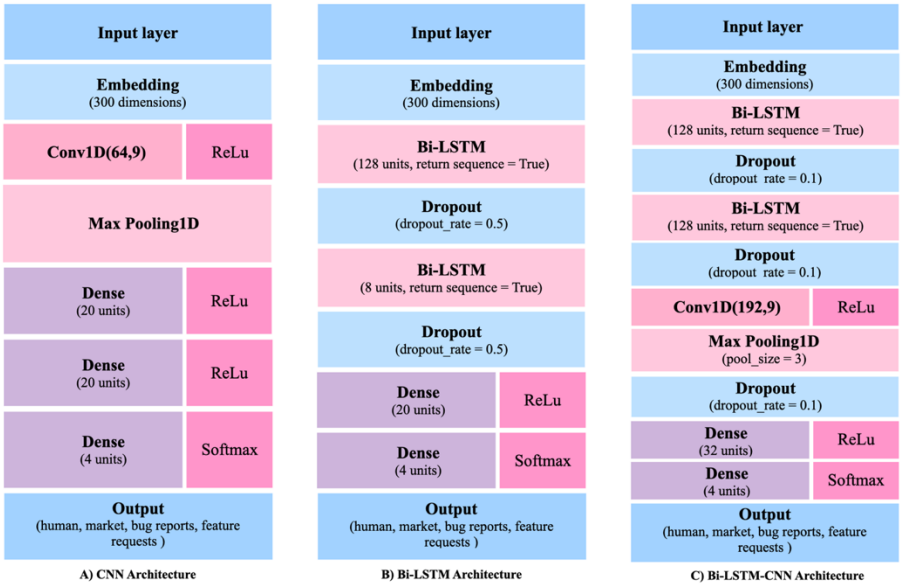



**Figure 8** Deep learning model Architecture

### 3.7 Model Evaluation

Model evaluation is the method of using different evaluations to comprehend the performance of a model and to realise its strengths or weaknesses. In this research, we used all these metrics namely, confusion matrix, accuracy, precision, recall and F1-score for measuring the performance of the models. Additionally, we ran all models 5 times and reported the average performance of scores.

#### 3.7.1 Confusion Matrix

A confusion matrix is a table that is used to describe a performance measurement for the classification method. The table of confusion matrix provides numbers from the actual and predicted values (Kulkarni et al., 2020).

**Table 3** Confusion Matrix

| | Actual | |
|---|---|---|
| | Yes | No |
| Yes | TP | FP |
| No | FN | TN |

- True Positive (TP): an instance correctly classified as Yes
- True Negative (TN): an instance correctly classified as No
- False Positive (FP): an instance incorrectly classified as Yes

- False Negative (FN): an instance incorrectly classified as No

#### 3.7.2 Accuracy

Accuracy is the ratio of correctly predicted samples to the total samples. Accuracy is normally between 0 to 1. Accuracy is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{21}$$

#### 3.7.3 Precision

Precision or positive predictive value is the ratio of true predicted positive samples with all total samples classified as positive. Precision is defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

#### 3.7.4 Recall

The recall is the proportion of relevant samples that were retrieved. The recall is defined as follows:

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

#### 3.7.5 F1-score

F1-score is the harmonic mean between precision and recall. F1 is the single metric that can measure model performance. F1-score is defined as follows:

$$F_1 = \frac{2}{recall^{-1} + precision^{-1}} \tag{24}$$

## 4 Result and Discussion

In this section, we compared the performance scores between deep learning and machine learning models on the user concerns dataset. We also reported the experimental results both with and without oversampling.

### 4.1 Performance of deep learning versus machine learning models

Table 3 shows that the Bi-LSTM-CNN model outperformed the other models, achieving the highest accuracy with 0.846 and the best values for precision, recall, and F1-score on the user concerns dataset with 0.848, 0.846 and 0.845 respectively. The decision tree model, a traditional machine learning model, had a relatively poor accuracy of 0.748, which was 9.8% lower than the Bi-LSTM-CNN model. Results indicate that combining Bi-LSTM and CNN can improve the performance better than only individual architecture.

A comparison of metrics of class classifiers in each classification model is illustrated in Table 4. The result provided that the Bi-LSTM-CNN model achieved the highest F1-score when classifying the sentences into aspects of feature request, human and market with 0.627, 0.894, 0.810 respectively while the Bi-LSTM got the highest score for classifying bug reports class with 0.878. Further, the results reveal that human concerns class achieved an F1-score higher than other classes. This is expected because most sentences can arguably related to human concerns more than any other class labels. Due to the fact that the classes are imbalanced data, the classifiers are usually biased toward a large proportion of the data set (Padurariu, and Breaban, 2019). As a result, a human concern which is the majority class is implemented for easier classifying with the highest F1-score than other label classes.

**Table 4** The performance of deep learning and machine learning models

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| CNN | 0.846 | 0.842 | 0.844 | 0.836 |
| Bi-LSTM | 0.834 | 0.834 | 0.834 | 0.834 |
| Bi-LSTM-CNN | **0.846** | **0.848** | **0.846** | **0.845** |
| RF | 0.807 | 0.807 | 0.788 | 0.768 |
| DT | 0.748 | 0.748 | 0.735 | 0.723 |
| KNN | 0.750 | 0.750 | 0.724 | 0.719 |

**Table 5** The performance scores of each class

| Model | Class | Precision | Recall | F1 |
|---|---|---|---|---|
| CNN | bug reports | 0.878 | 0.878 | 0.878 |
| | feature request | 0.754 | 0.484 | 0.576 |
| | human | 0.862 | 0.928 | 0.888 |
| | market | 0.804 | 0.814 | 0.796 |
| Bi-LSTM | bug reports | 0.880 | 0.846 | 0.862 |
| | feature request | 0.648 | 0.544 | 0.592 |
| | human | 0.884 | 0.886 | 0.886 |
| | market | 0.752 | 0.854 | 0.800 |
| Bi-LSTM-CNN | bug reports | 0.884 | 0.853 | 0.868 |
| | feature request | 0.710 | 0.563 | 0.627 |
| | human | 0.897 | 0.892 | 0.894 |
| | market | 0.750 | 0.881 | 0.810 |
| RF | bug reports | 0.851 | 0.819 | 0.836 |
| | feature request | 0.894 | 0.268 | 0.411 |

| | | | | |
|---|---|---|---|---|
| | human | 0.742 | 0.964 | 0.835 |
| | market | 0.853 | 0.671 | 0.751 |
| DT | bug reports | 0.704 | 0.837 | 0.759 |
| | feature request | 0.705 | 0.332 | 0.451 |
| | human | 0.763 | 0.849 | 0.801 |
| | market | 0.769 | 0.596 | 0.671 |
| KNN | bug reports | 0.684 | 0.771 | 0.771 |
| | feature request | 0.730 | 0.458 | 0.458 |
| | human | 0.870 | 0.770 | 0.770 |
| | market | 0.585 | 0.670 | 0.670 |

We also present the confusion matrix of each model in Figure 9. The results suggest that Bi-LSTM-CNN achieved the highest correctly classified with 695 correct predictions namely, 195 bug reports, 48 feature request, 320 human and 132 market and only give 120 wrong predictions out of 815 examples. Meanwhile, the decision tree, which is the worst performing model, gives 614 instances correctly including, 159 bug reports, 34 feature request, 324 human and 97 market and 201 examples gives incorrectly classified.

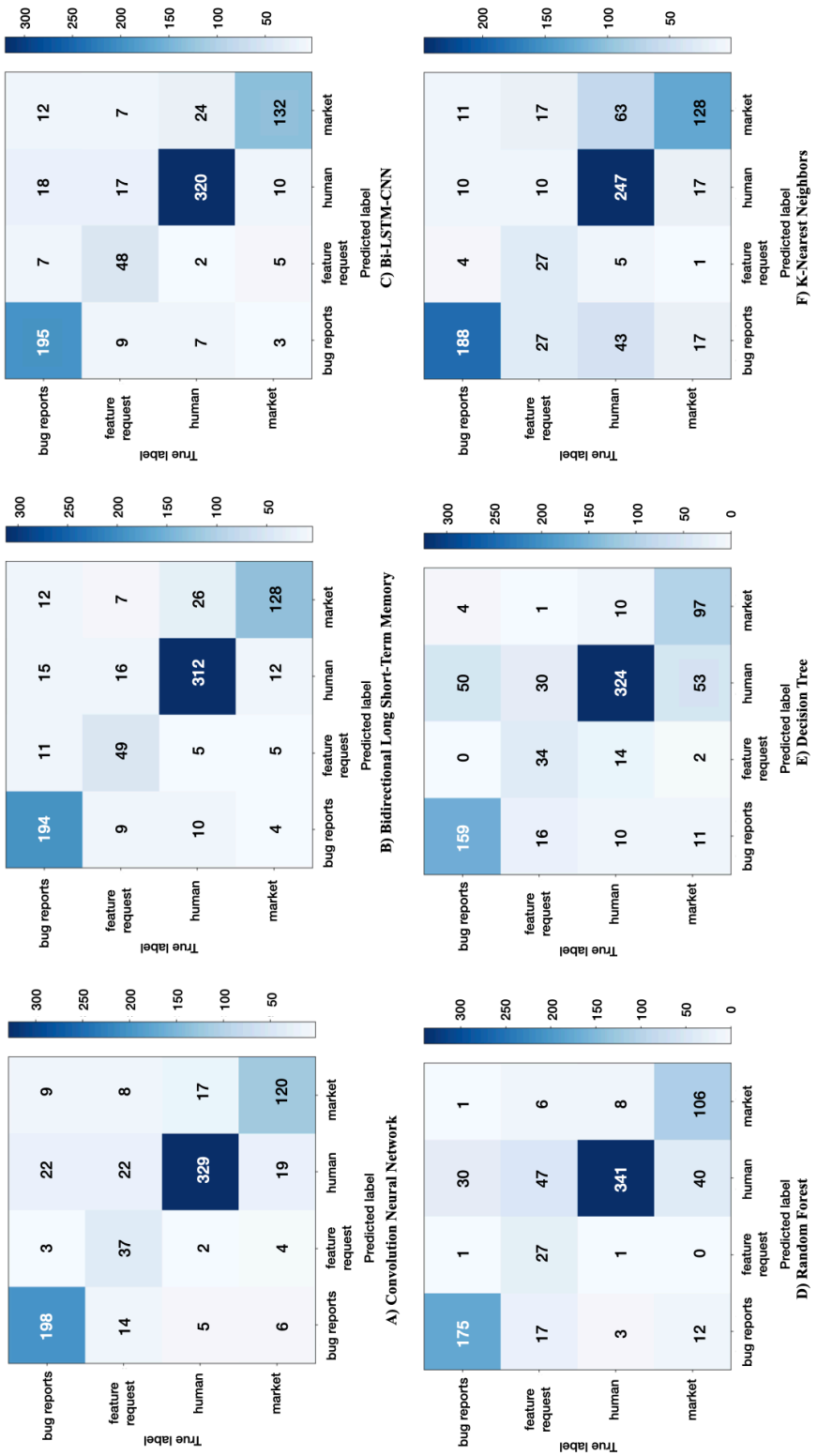



**Figure 9** Confusion matrix of all models

### 4.2 Comparison of performance results on imbalanced data and balanced data

The table 6 illustrates the performance scores after SMOTE oversampling on the training data. All performance scores, particularly accuracy, precision, recall, and F1-score, did not increase in our study. The accuracy of Bi-LSTM-CNN trained on the original training data is better than that trained on the balanced data set. Similarly, the F1-score of Bi-LSTM-CNN with the resampling technique has a decline of 5%. It seems, from the results, that the SMOTE technique does not work well in our multi-class problem.

**Table 6** The performance scores after SMOTE technique

| Model | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| CNN | 0.846 | 0.842 | 0.844 | 0.836 |
| Bi-LSTM | 0.834 | 0.834 | 0.834 | 0.834 |
| Bi-LSTM-CNN | **0.846** | **0.848** | **0.846** | **0.845** |
| CNN* | 0.729 | 0.757 | 0.749 | 0.735 |
| Bi-LSTM* | 0.783 | 0.821 | 0.783 | 0.796 |
| Bi-LSTM-CNN* | 0.779 | 0.826 | 0.779 | 0.795 |

*SMOTE technique

Table 7 displays the performance of different classification models across various report types following the application of the SMOTE technique to tackle class imbalance issues. Generally, there's an improvement in the precision of deep learning models concerning the human class. More precisely, the precision of CNN, Bi-LSTM, and Bi-LSTM-CNN models increased by approximately 1.1%, 4.1%, and 3.5%, respectively. However, it's noteworthy that the application of the SMOTE technique did not yield performance enhancements for any other class.

**Table 7** The performance scores of each class after SMOTE technique

| Model | Class | Precision | Recall | F1 |
|---|---|---|---|---|
| CNN* | bug reports | 0.830 | 0.716 | 0.754 |
| | feature request | 0.320 | 0.370 | 0.325 |
| | human | 0.873 | 0.810 | 0.839 |
| | market | 0.629 | 0.752 | 0.683 |
| Bi-LSTM* | bug reports | 0.888 | 0.784 | 0.830 |
| | feature request | 0.374 | 0.591 | 0.455 |
| | human | 0.925 | 0.827 | 0.872 |
| | market | 0.720 | 0.784 | 0.750 |
| Bi-LSTM-CNN* | bug reports | 0.889 | 0.791 | 0.836 |
| | feature request | 0.354 | 0.557 | 0.433 |

| | | | |
|---|---|---|---|
| human | 0.932 | 0.813 | 0.868 |
| market | 0.717 | 0.826 | 0.767 |

Figure 10 depicts the confusion matrix of the deep learning model following the integration of the SMOTE technique. The findings demonstrate that Bi-LSTM-CNN attained the highest number of accurate predictions, correctly classifying 653 instances. To break it down, it accurately predicted 183 instances for bug reports, 49 for feature requests, 302 for human concerns, and 119 for market-related issues. These results imply that the effectiveness of the SMOTE technique may be limited in addressing our multi-class problem. The observation is backed up by some evidence from Padurariu, and Breaban (2019) which suggests that SMOTE tended not to work well on more complex embedding spaces like those generated by Glove, The small sample size severe imbalance ratio also affect the working of SMOTE.

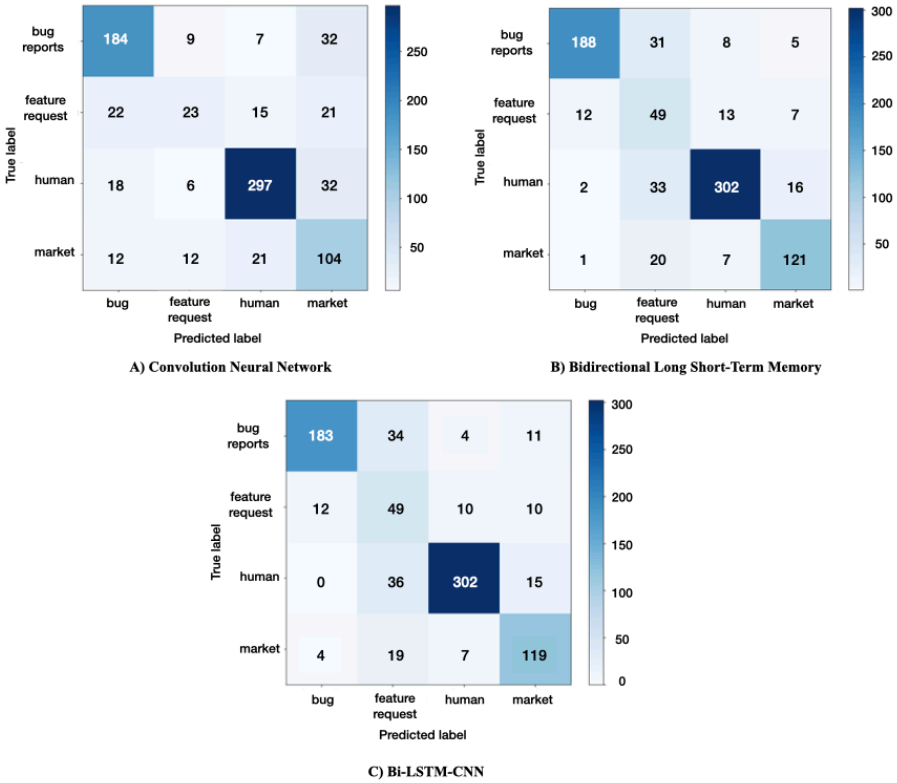
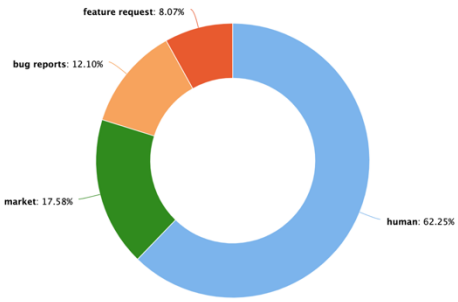


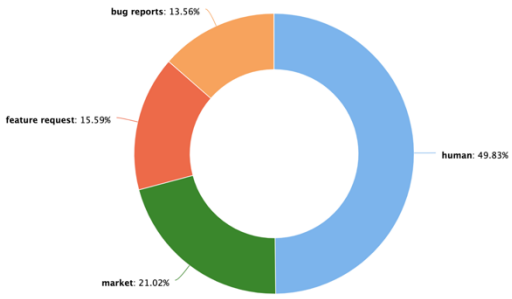

**Figure 10** Confusion Matrix for Deep Learning model after SMOTE technique

### 4.3 Model Deployment

In terms of practical implications, the proposed automatic text classification could be used to characterise problems in software requirements and business domains. In this section, the unseen user reviews from food delivery apps namely, Deliveroo and Grabhub are tested with the proposed model. The results presented in Figure 11 and Figure 12 summarised the composition of user concerns in a month. The two doughnut charts illustrate how different categories of user concerns contributed to the food delivery apps percentages from 1 Dec 2022 to 31 Dec 2022. Overall, human concerns were the most significant user concerns sector in both Deliveroo and Grabhub apps, while the feature request concerns (8.07%) and bug report concerns (13.56%) contributed the least to user concerns in Deliveroo and Grabhub apps respectively. In general, feature request concerns in Grabhub apps are almost twice as higher as in Deliveroo apps while other user concerns are slightly different. Overall, this indicates that all these apps had major problems with service quality such as the interactions with the providers. Users are also dissatisfied with driver behaviour, late orders or the cancellation of the order. Overall, the classification of the types of user concerns gives an overview of the users' dissatisfactions over the software and could hopefully be beneficial in prioritising software updates.

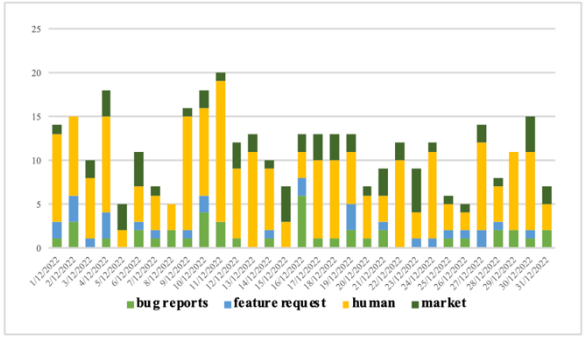



**A) The pie chart of predicted user concerns for Deliveroo apps**

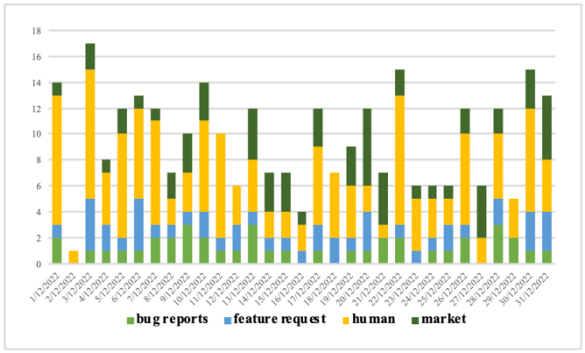



**B) The pie chart of predicted user concerns for Grabhub apps**

**Figure 11** The pie chart of predicted user concerns




A) The stacked bar chart of predicted user concerns for Deliveroo apps




B) The stacked bar chart of of predicted user concerns for Grabhub apps

**Figure 12** The stacked bar chart of predicted user concerns

## 5 Conclusion and Future Work

In this paper, we studied text classification using deep learning approaches to classify user concerns for food delivery applications. The user reviews on Google Play and App store were collected and manually labelled into four categories: bug report, human, market, and feature request. The results of this study revealed that Bi-LSTM-CNN, a combination of Bi-LSTM and CNN models, attained an accuracy of 0.846, which is superior to individual network architectures. In addition, the combination of Bi-LSTM and CNN was able to capture the semantic sentence in the text sequences and outperformed the traditional machine learning models. Moreover, we tried to increase the performance scores by using SMOTE technique to increase the training data size; nonetheless, the results reported that the re-sampling technique did not significantly improve the accuracy, precision and F1-score. This conceptual framework can be a guideline for classifying the aspect of user concerns which are crucial factors in evaluating the effectiveness of both software management and sharing economy applications. Besides, our architecture can hopefully be adapted to other domains with minor modifications.

For our future work, we will apply the new techniques of word embeddings, such as Bi-directional Encoder Representations from Transformers (BERT) to generate contextualized word embeddings on a large corpus of text. For the problem of data class

imbalance, we will apply the focal loss function to handle class imbalance text. Furthermore, due to the fact that there are an abundance of unlabelled user concerns, we might consider adopting the self-supervised learning technique for leveraging the unlabeled data in the training.

## References

S. Albawi, et al. (2017) 'Understanding of a convolutional neural network', *2017 international conference on engineering and technology (ICET)*, Ieee, pp. 1-6.

D. Allen (2015) 'The sharing economy', *Institute of Public Affairs Review: A Quarterly Review of Politics and Public Affairs, The*, Vol. 67 No. 3, pp. 24-27 (Access 2015).

J.P. Allen (2017) 'Technology and inequality case study: The sharing economy', *Technology and Inequality*, Springer, pp. 121-135.

Z. Bao and Y. Zhu (2021) 'Why customers have the intention to reuse food delivery apps: evidence from China', *British Food Journal*, (Access 2021).

P. Bhuvaneshwari, et al. (2022) 'Sentiment analysis for user reviews using Bi-LSTM self-attention based CNN model', *Multimedia Tools and Applications*, Vol. 81 No. 9, pp. 12405-12419 (Access 2022).

S. Bird and E. Loper (2004) 'NLTK: the natural language toolkit', Association for Computational Linguistics.

Y. Chen (2015) *Convolutional neural network for sentence classification*. University of Waterloo.

A. Ciurumelea, et al. (2017) 'Analyzing reviews and code of mobile apps for better release planning', *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, IEEE, pp. 91-102.

D. Curry 'Food Delivery App Revenue and Usage Statistics (2022)' [online] https://www.businessofapps.com/data/food-delivery-app-market/.

M.S.N.M. Danuri, et al. (2022) 'The Improvement of Stress Level Detection in Twitter: Imbalance Classification Using SMOTE', *2022 IEEE International Conference on Computing (ICOCO)*, IEEE, pp. 294-298.

S. Elbagir and J. Yang (2019) 'Twitter sentiment analysis using natural language toolkit and VADER sentiment', *Proceedings of the international multiconference of engineers and computer scientists*, pp. 16.

S. Ghannay, et al. (2016) 'Word embedding evaluation and combination', *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 300-305.

C. Guthrie, et al. (2021) 'Online consumer resilience during a pandemic: An exploratory study of e-commerce behavior before, during and after a COVID-19 lockdown', *Journal of Retailing and Consumer Services*, Vol. 61, pp. 102570 (Access 2021).

E. Hindocha, et al. (2019) 'Short-text Semantic Similarity using GloVe word embedding', *Int. Res. J. Eng. Technol*, Vol. 6, pp. 553-558 (Access 2019).

S. Hochreiter and J. Schmidhuber (1997) 'Long short-term memory', *Neural computation*, Vol. 9 No. 8, pp. 1735-1780 (Access 1997).

H. Hoehle and V. Venkatesh (2015) 'Mobile application usability', *MIS quarterly*, Vol. 39 No. 2, pp. 435-472 (Access 2015).

R. Islam, et al. (2010) 'Mobile application and its global impact', *International Journal of Engineering & Technology (IJEST)*, Vol. 10 No. 6, pp. 72-78 (Access 2010).

B. Jang, et al. (2020) 'Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism', *Applied Sciences*, Vol. 10 No. 17, pp. 5841 (Access 2020).

Y. Kim (2014) 'Convolutional Neural Networks for Sentence Classification', *arXiv preprint arXiv:1408.5882*, (Access 2014).

F. Koto (2014) 'SMOTE-Out, SMOTE-Cosine, and Selected-SMOTE: An enhancement strategy to handle imbalance in data level', *2014 international conference on advanced computer science and information system*, IEEE, pp. 280-284.

A. Kulkarni, et al. (2020) 'Foundations of data imbalance and solutions for a data democracy', *data democracy*, Elsevier, pp. 83-106.

L.-C. Kung and G.-Y. Zhong (2017) 'The optimal pricing strategy for two-sided platform delivery in the sharing economy', *Transportation Research Part E: Logistics and Transportation Review*, Vol. 101, pp. 1-12 (Access 2017).

E.-Y. Lee, et al. (2017) 'Factors influencing the behavioral intention to use food delivery apps', *Social Behavior and Personality: an international journal*, Vol. 45 No. 9, pp. 1461-1473 (Access 2017).

C. Li, et al. (2017) 'Deep memory networks for attitude identification', *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 671-680.

D. Liang and Y. Zhang (2016) 'AC-BLSTM: asymmetric convolutional bidirectional LSTM networks for text classification', *arXiv preprint arXiv:1611.01884*, (Access 2016).

P. Liu, et al. (2015) 'Fine-grained opinion mining with recurrent neural networks and word embeddings', *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1433-1443.

N. Madnani (2007) 'Getting started on natural language processing with Python', *XRDS: Crossroads, The ACM Magazine for Students*, Vol. 13 No. 4, pp. 5-5 (Access 2007).

S. Mehrolia, et al. (2021) 'Customers response to online food delivery services during COVID-19 outbreak using binary logistic regression', *International journal of consumer studies*, Vol. 45 No. 3, pp. 396-408 (Access 2021).

C. Muangmee, et al. (2021) 'Factors determining the behavioral intention of using food delivery apps during COVID-19 pandemics', *Journal of Theoretical and Applied Electronic Commerce Research*, Vol. 16 No. 5, pp. 1297-1310 (Access 2021).

K. O'Shea and R. Nash (2015) 'An introduction to convolutional neural networks', *arXiv preprint arXiv:1511.08458*, (Access 2015).

C. Padurariu and M.E. Breaban (2019) 'Dealing with data imbalance in text classification', *Procedia Computer Science*, Vol. 159, pp. 736-745 (Access 2019).

D. Pagano and W. Maalej (2013) 'User feedback in the appstore: An empirical study', *2013 21st IEEE international requirements engineering conference (RE)*, IEEE, pp. 125-134.

J. Pennington, et al. (2014) 'Glove: Global vectors for word representation', *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532-1543.

G. Pigatto, et al. (2017) 'Have you chosen your request? Analysis of online food delivery companies in Brazil', *British Food Journal*, (Access 2017).

A. Ray, et al. (2019) 'Why do people use food delivery apps (FDA)? A uses and gratification theory perspective', *Journal of Retailing and Consumer Services*, Vol. 51, pp. 221-230 (Access 2019).

K. Roberts (2016) 'Assessing the corpus size vs. similarity trade-off for word embeddings in clinical NLP', *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*, pp. 54-63.

V. Rupapara, et al. (2021) 'Impact of SMOTE on imbalanced text features for toxic comments classification using RVVC model', *IEEE Access*, Vol. 9, pp. 78621-78634 (Access 2021).

N. Senthil Kumar and N. Malarvizhi (2020) 'Bi-directional LSTM–CNN combined method for sentiment analysis in part of speech tagging (PoS)', *International Journal of Speech Technology*, Vol. 23 No. 2, pp. 373-380 (Access 2020).

F. Sjahroeddin (2018) 'The role of ES-Qual and food quality on customer satisfaction in online food delivery service', *Prosiding Industrial Research Workshop and National Seminar*, pp. 551-558.

B. Solomon (2015) 'America's most promising company: Instacart, the $2 billion grocery delivery app', *Forbes, January*, Vol. 21, (Access 2015).

Statista 'Number of downloads of leading food delivery and takeout apps in the United Kingdom (UK) in 2021' [online] https://www.statista.com/statistics/1298186/food-delivery-app-downloads-uk/.

B. Sumathi (2020) 'Grid search tuning of hyperparameters in random forest classifier for customer feedback sentiment prediction', *International Journal of Advanced Computer Science and Applications*, Vol. 11 No. 9, (Access 2020).

W. Sutherland and M.H. Jarrahi (2018) 'The sharing economy and digital platforms: A review and research agenda', *International Journal of Information Management*, Vol. 43, pp. 328-341 (Access 2018).

W. Wang and J. Gang (2018) 'Application of convolutional neural network in natural language processing', *2018 International Conference on Information Systems and Computer Aided Education (ICISCAE)*, IEEE, pp. 64-70.

G. Williams, et al. (2020) 'Modeling user concerns in sharing economy: the case of food delivery apps', *Automated Software Engineering*, Vol. 27 No. 3, pp. 229-263 (Access 2020).

J. Wirtz, et al. (2019) 'Platforms in the peer-to-peer sharing economy', *Journal of Service Management*, (Access 2019).

H. Wu, et al. (2020) 'Review of Text Classification Methods on Deep Learning', *Computers, Materials \& Continua*, Vol. 63 No. 3, pp. 1309--1321 http://www.techscience.com/cmc/v63n3/38877 (Access 2020).

R. Xu, et al. (2015) 'Word embedding composition for data imbalances in sentiment and emotion classification', *Cognitive Computation*, Vol. 7, pp. 226-240 (Access 2015).

L. Yang and A. Shami (2020) 'On hyperparameter optimization of machine learning algorithms: Theory and practice', *Neurocomputing*, Vol. 415, pp. 295-316 (Access 2020).

W. Zhao, et al. (2020) 'The study on the text classification for financial news based on partial information', *IEEE Access*, Vol. 8, pp. 100426-100437 (Access 2020).