Effect of Word Embedding in Online Learning Tweet Multi-Classification with CNN and LSTM Variants

Marina Yusoff[1*, 2] Muhammad Syamil Ali[3], Nor Hapiza Mohd Ariffin[4], Mohammed Adam Kunna Azrag[1]

[1]Institute for Big Data Analytics and Artificial Intelligence (IBDAAI) Kompleks Al-Khawarizmi, Universiti Teknologi MARA (UiTM), 40450 Shah Alam, Selangor Darul Ehsan Malaysia

[2]College of Computing, Informatic and Mathematics, Kompleks Al-Khawarizmi, Universiti Teknologi MARA (UiTM), 40450 Shah Alam, Selangor Darul Ehsan, Malaysia

[3]2X Marketing Sdn Bhd, Jalan Wan Kadir, Taman Tun Dr Ismail, 60000 Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur

[4]MIS department, Faculty of Business, Sohar University, Oman

E-mail address: marina998@uitm.edu.com, syamil.ali@2x.marketing, nariffin@su.edu.om, kunna87mohammed@outlook.com

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: The paper examines the effectiveness of multi-class sentiment analysis strategies using deep learning methods for imbalanced and balanced datasets with and without word embeddings. Seven models, including Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), CNN-LSTM, LSTM-CNN, Bidirectional LSTM (BILSTM), CNN-BILSTM and Two layers CNN are compared. A dataset consisting of 23,168 tweets was gathered from online learning platforms between 2020 and 2021. The performance of sentiment categorization was evaluated based on accuracy, precision, recall, and F1-score. The study presents three main findings: (1) a comparison of the effectiveness of seven sentiment analysis algorithms, (2) the clear advantage of pre-trained Word2vec, and (3) the capability to achieve a balanced sentiment categorization using Twitter data. The LSTM-CNN model utilizing Word2vec word embedding outperformed several models, achieving an accuracy of 89.66% and a precision, recall, and F1-Score of 90.00% for the testing results. The experimental results confirmed that this methodology enhanced the accuracy of sentiment classification compared to standard methods and exhibited superior classification performance. The empirical research showed that the LSTM-CNN method was fast, efficient, and viable, making it a potentially better option for optimizing online learning rules. The study provides valuable insights to analytics professionals and academicians engaged in text analysis. It focuses on the performance evaluation of essential algorithms in sentiment classification, particularly emphasizing the data balancing technique in deep learning hybrid models.

Keywords: Deep Learning, Machine Learning, Online Learning, Sentiment Analysis, Multi-classification

1. Introduction

Social media can gather information and feedback on educational topics such as teacher performance, learning experience, and other course attributes. Social media influences the education field. It can provide meaningful information to people affected by educational policies [1]. Millions of people use that social media platform. For example, Twitter's social media platform currently has approximately 192 million active users, with 500 million tweets sent daily, equating to 5787 tweets per second [3]. As a result, it takes work to track users' overall opinions on social media topics. As a result, sentiment analysis can be a helpful tool in dealing with this issue. It focuses on the interpretation and classification of emotions in subjective data, and it is mainly used on textual data to detect sentiment in emails, survey responses, and social media [2]. Monitoring sentiment or opinion via social media is a viable option. Researchers have recently used sentiment analysis in online learning to improve teaching quality and learn about the relationships between teachers and students. A sentiment analysis system is used to help teachers better understand their students' moods and assist their teaching methods [3]. Various approaches have been developed and tested in sentiment analysis, but the two major approaches are machine learning and lexicon-based [4].

Machine learning approaches must be trained and tested on datasets for sentiment polarity prediction. Meanwhile, Lexicon-based sentiment polarity prediction does not involve training or testing with datasets and employs a pre-built list of words associated with a given sentiment [5]. Researchers use word embedding to extract information from unstructured text corpora [6, 7].

E-mail:marina998@uitm.edu.my

Word embedding can test unorganized text's semantic and syntactic patterns by showing words in a vector space [8].

DL methods, a subfield of machine learning, work better in sentiment analysis than other machine learning techniques because they can handle more complicated problems [9]. Some research looks at how people feel about movie reviews on Twitter using the Convolution Neural Network (CNN) and Long Short-Term Memory (LTSM) [10, 11] and social media datasets [12,13]. LSTM and CNN have shown superior performance to other machine learning methods because most tweet datasets are from benchmark data. Notably, CNN committed to providing effective local feature selection while the LSTM recurrent network focused on its sequential analysis of a long text. However, there are no findings on using CNN and LSTM for the online learning tweet dataset. As a result, the advantage of both methods has seen the potential for better performance with more evaluation of hyperparameter selection [14]. This research aims to automatically find Malaysian people's sentiments about online learning on the Twitter platform and identify the identify topics by Twitter users when expressing their emotions about online learning [15–17]. This study makes the following key contributions:

- Initiation of the data acquisition and preprocessing Preprocessing involved removing unrelated elements such as tabs, newlines, special characters, punctuation, numerals, repeated words, hashtag symbols, non-English characters, and unnecessary tabs and spaces from the original datasets.
- Exploratory research, such as keyword trend analysis and topic modeling, is carried out to understand the collected data better. Furthermore, feature extraction with and without pre-trained Word2Vec is performed.
- Initiatives that employ random oversampling methods and the Synthetic Minority Oversampling Technique (SMOTE).
- The extracted discriminative word embedding vectors are fed into seven CNN and LSTM combinations to classify Malaysian sentiments as positive, negative, or neutral. The proposed hybrid CNN and LSTM classification models improve accuracy by evaluating several hyperparameters, such as batch size and epoch.

The paper is structured as follows: Section 2 covers word embedding, random sampling, and deep learning (DL) research. Section 3 elaborates on the setting of our research and highlights online learning. Section 4 describes the experimental design and technique. Section 5 presents and discusses our significant findings. The paper's conclusion in Section 6 outlines the future scope of work.

2. Material and methods

A. Data Collection

Using the snscrap Python module, 23168 tweets from Twitter in Malaysia have been successfully scraped. Used keywords related to online learning number 23. The keywords are "online school," "online class," "online learning," "pdpr," "odl," "distance learning," "online teaching," "teach online," "e-learning," "online education," "study online," "online course," "edtech," "online school," "online learning," "online lecture," "remote learning," and "blended learning." The scraped tweets are all bilingual in Malay and English. During the pandemic, the tweet data ranged from 2020 to 2021. With references to previous articles, websites, and brainstorming, much study has been done to identify appropriate keywords connected to online learning. In this work, 23 different Twitter keywords are associated with online learning. Some of the keywords used in this article to collect relevant tweets include "online class," "online learning," "digital learning," "online training," "online lecture," and "remote learning.".

B. Data Preprocessing

Data preprocessing requires comprehensive cleaning to assure data quality. The dataset was extensively cleansed during data preprocessing. Several steps involve data preprocessing, including link address removal, lowercasing, number, and special character removal, stop words removal, emojis and emotions removal, tokenization, lemmatization, and labeling.

C. Data Balancing

The presence of imbalanced datasets would affect the class's skewness and lead to bias in the training dataset. An unbalanced distribution class would ignore the minority class in a classification process. Resampling data with random oversampling can potentially duplicate minority classification features [18,19]. Its goal is to keep the existing features. SMOTE is a classic oversampling method that has recently gained popularity for assisting in the solution of different data types in many domains. It was invented by [20]. SMOTE can generate synthetic examples to convert a minority class into a balanced class along the class decision boundary aspect [21]. In this regard, SMOTE selects a close attribute or feature from the feature space by introducing a boundary line, and a new sample is generate**d** along the same line. The calculation procedure on the work of [20, 22].

D. Model Establishment

The architecture of the proposed CNN-LSTM in single and hybrid variants is presented in this section. We build a single CNN and LSTM, as shown in Fig. 1 (a) and (b). The CNN model determines the sentiment of sentences based on the local features extracted by the convolution layer. Fig.1 demonstrates the architecture of the LSTM (b). Fig. 1 (c), (d), and (e) depicts a CNN-LSTM bidirectional LSTM (BILSTM) and CNN-LSTM, respectively. Local feature learning in bidirectional sequences in this model. As shown in Fi 1 (f), we extend the model by embedding the CNN with BILSTM. The model works in the same way as CNN-LSTM. The difference is that bidirectional sequences learn from the CNN's pooling layer output. Furthermore, we should contrast with 2-layer CNN, as shown in Fig. 1(g).

E. Performance Evaluation

The performance evaluation relies on a confusion matrix table, commonly used to depict a classification model's performance. True positive (TP) and true negative (TN) are instances where the predictions are accurate. The four primary performance measures assessed are accuracy, precision, recall, and F1-Score. Accuracy is a straightforward and intuitive way to measure performance. The calculation for accuracy (A), precision (P), recall (R), and F1-Score (F) are in Equations (1), (2), (3), and (4).

$$A = TP+TN/TP+FP+FN+TN \quad (1)$$

$$P = TP/TP+FP \quad (2)$$

$$R= TP/TP+FN \quad (3)$$

$$F = 2*(R*P) / (R+P) \quad (4)$$

3. COMPUTATIONAL RESULTS

The results reflect the oversampling method for balanced text data based on 32 and 64 batch sizes using a trainable embedding vector at 0.9 splits. Fig. 2 (a), (b), (c), and (d) illustrate results from the experiment on batch equal to 32 and with trainable ability at the embedding layer. It resulted in a good accuracy of 94% for training accuracy by using 90% of the text data. Meanwhile, it needs more validation accuracy. The difference between loss and loss validation is about 0.14 and 0.40, respectively. As shown in Fig 2 (e), (f), (g), and (h), the performance is shown to be less accurate and about similar in terms of loss value to a batch size equal to 32. However, the computational time is seen to increase.

Table 1 shows the results from the best model, which was reached by all 32 models using trainable and non-trainable oversampling methods. So, the results show how well the DL models work when 100 epochs and 32 batch sizes are added to an experiment with and without words that can be trained. Random oversampling was used to make the balanced data pattern. According to the results, all models work well with the hyperparameter settings described in the initial part of the experiment. Each model achieved an accuracy level between 87% and 90% when trained on embeddable words. With an accuracy of 89.66%, the LSTM-CNN model is the most accurate of the bunch. It also has the best accuracy, memory, and F1-Score of 0.90. The second-highest percentage was achieved by CNN-LSTN, at 89.18%, with accuracy and recall scores of 0.89 and an F1-Score of 0.89.

Table 2 shows the results on 64 batch normalization values and Word2vec with and without training. The LSTM-CNN model, which combines CNN and LSTM, does better than all the other models, as shown by the data. There has been a 1% to 2% drop in the accuracy of the CNN, LSTM, and BILSTM models, but the Precision, Recall, and F1 scores still give steady results with an average of 88%. It works the same way for 32 and 64-person groups. The effectiveness of CNN and two-layer CNN models is 85%. Reports say this works as well as a batch size of 32.

4. DISCUSSION

We presented seven different CNN and LSTM variants, including standard CNN and LSTM models, hybrid CNN and LSTM models, and a single LSTM and convolutional neural network (CNN) model for sentiment analysis. The results demonstrated that the LSTM-CNN model with Word2vec word implanting outperformed the other models, achieving an impressive 89.66% accuracy and 90.00% precision, recall, and F1-Score.

A. Effect of word embedding and trainable ability

Using Word2vec as an embedding word is sufficient to attain optimal classification performance, both with and without pre-trained weight. Word2vec, founded at Google [23], is a word embedding algorithm that employs a neural network to discover a transformation word vector and produce word embeddings of superior quality. It is a method for quickly learning the sentiment of text based on the assumption of semantic relationships between vectors [24]. The effectiveness of the word embedding technique was verified through supplementary analyses that juxtaposed the performance of our deep learning models with that of machine learning approaches. Compared to ML models, both pre-trained and untrained weights implemented in Word2vec exhibit superior performance. Word2vec enhances the model's performance in text sentiment analysis tasks and has evident benefits.

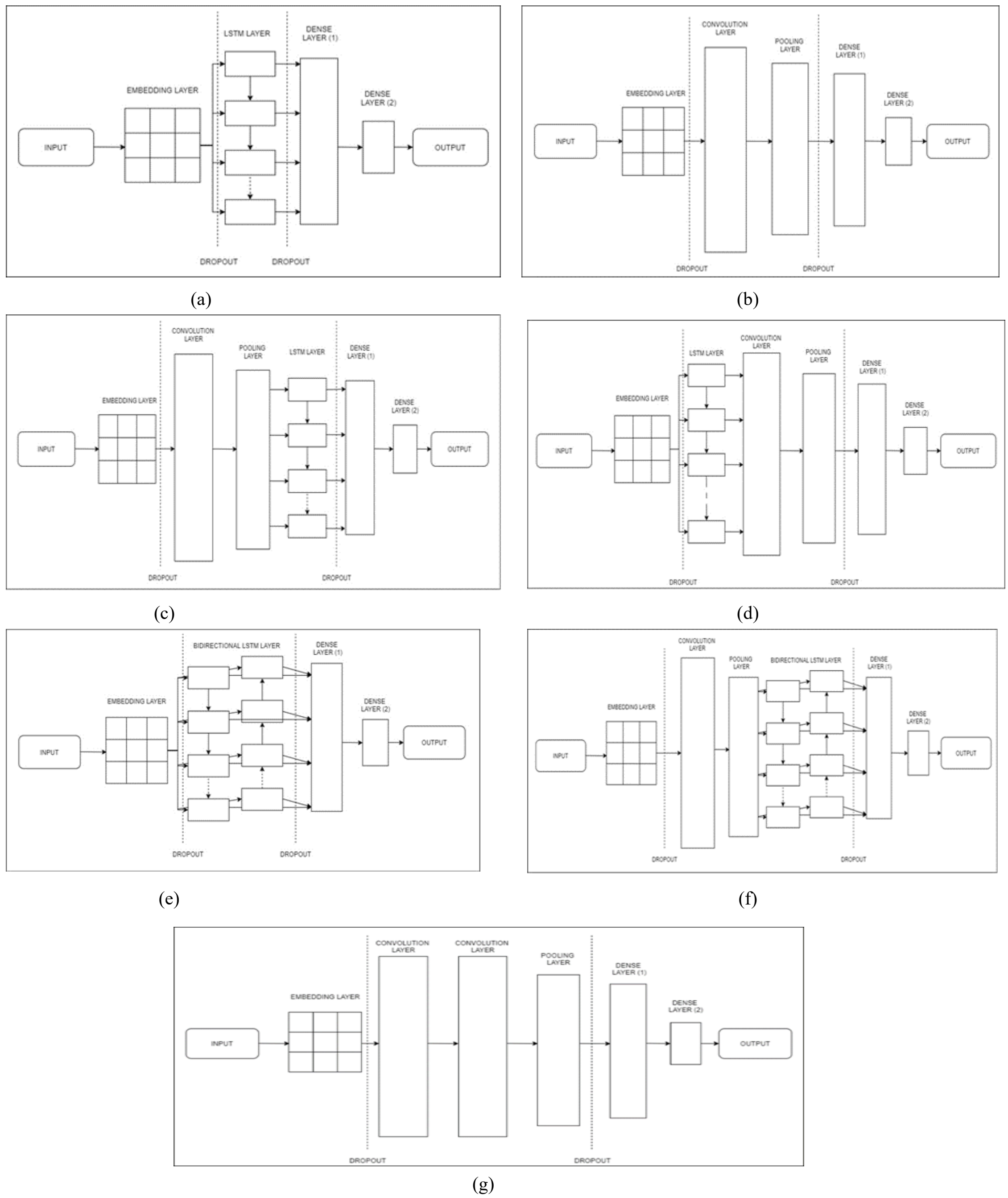

Figure 1. Architecture of CNN and LSTM and its variants (a) Single LSTM (b) Single LSTM (c) CNN-LSTM (d) LSTM-CNN (e) Bidirectional LSTM (f) CNN-Bidirectional LSTM (g) 2 Layer CNN

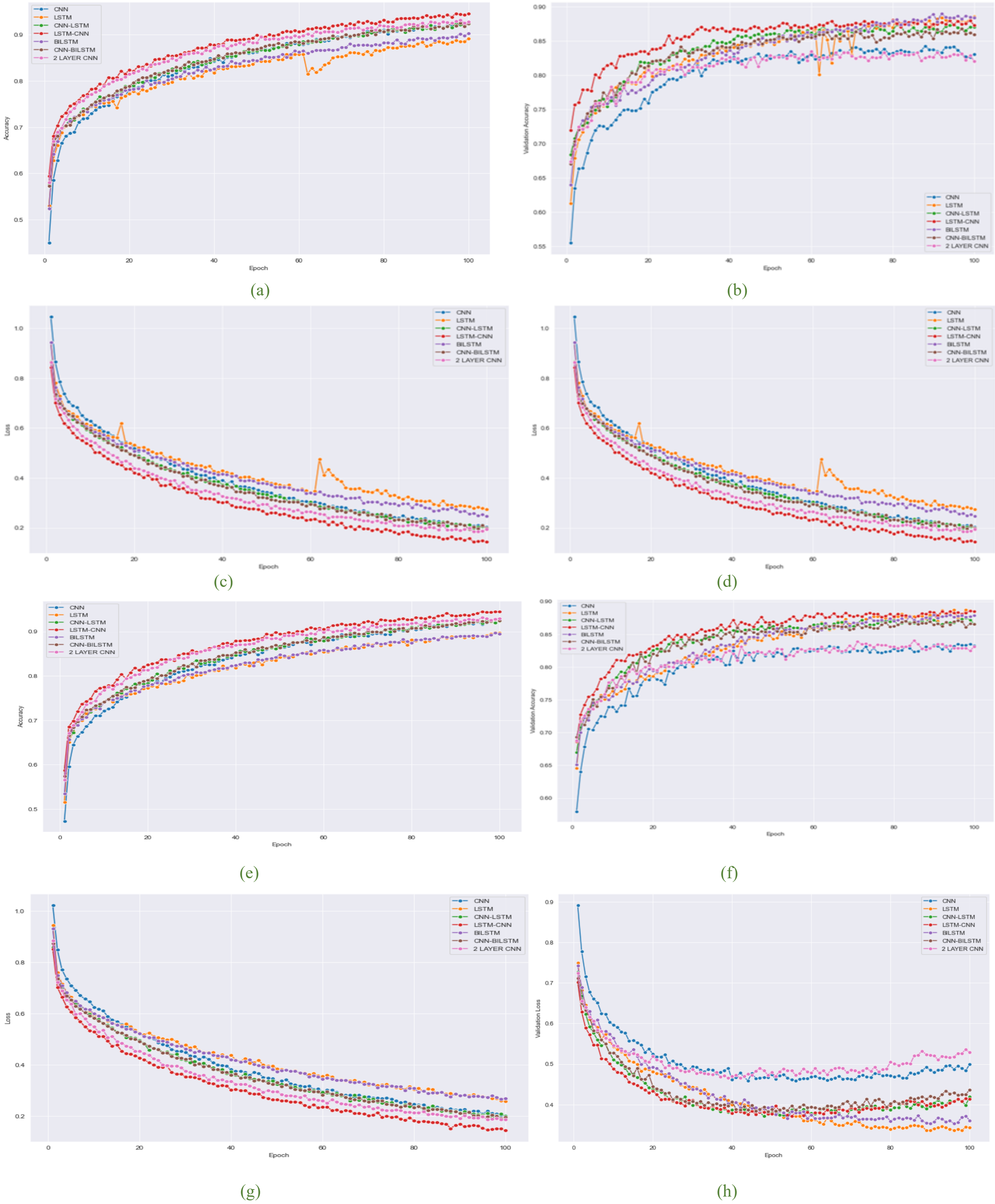

Figure 2. Computational results for balanced data on 0.9 split strategy without trainable at embedded layer (a) training accuracy for batch 32, (b) validation accuracy batch 32, (c) loss for batch 32 (d) validation loss for batch 32 (e) training accuracy for batch 64 (f) validation accuracy for (g) loss for batch 64 (h) validation loss for batch 64

The embedding layer's role is to feed the vectors converted from text to a numerical representation value. There are several ways to perform embedding tasks, such as word2vec and BERT. We first evaluated the embedding method and the results using all seven models. Further work on Word2Vec is then performed by considering trainable and non-trainable in the embedding layer. In this case, the embedding layer is evaluated with trainable and without trainable to see how weight values affect learning the pattern of the text data. If it is run with pre-trained weight, the model utilized the weight values the same way during training and in dense layer learning activity. It would lead to the effect of trainable in the embedding layer, which would offer efficiency at representing a similar word in the dense layer.

B. Effect of hyperparameter tuning during training and validation

This section compares all seven DL model outputs to state-of-the-art approaches. CNN feature extraction and association [25]. Many literatures discuss hyperparameter tuning to select the optimum CNN model. Random oversampling improves sentiment analysis accuracy, especially with online learning of Twitter datasets. All DL models improved by 10% in accuracy and lost -0.02. Oversampling balances a dataset by randomly copying class data [26]. In terms of the LSTM model, it was reported that LSTM has its flaws, which means it needs many resources and time to build the model [27]. LSTM has quite similar behavior to feed-forward neural networks, and it is susceptible to overfitting and problematically implementing the dropout algorithm to handle the issue [28]. In this work, after carefully identifying appropriate parameters, the results are as expected and demonstrate an improvement, as indicated in Table 3. In this case, the 100 epochs and 32 batch sizes used have significantly improved the classification accuracy of the LSTM model. The results are competitive with single-layer CNN and 2 Layer models. CNN and its hybrid with LSTM models have shown good accuracy and loss performance. CNN may benefit from local and higher-level text patterns [28]. The hybrid LSTM-CNN model outperformed the Single Layer CNN, Single Layer LSTM, CNN-LSTM, Bidirectional LSTM, CNN-Bidirectional LSTM, and 2 Layer CNN models. The LSTM layer's strength is learning word patterns sequentially from embedding vectors, and CNN creates feature maps and sentiment class prediction. In addition, the LTSM-CNN had the smallest value of the loss function and an acceptable computational time for the model's training.

However, the result is competitive to CNN-Bidirectional LSTM in experiment 5 for accuracy, loss, and computational time for training. The performance of CNN-Bidirectional LSTM could be due to the additional layer of the LSTM layer, which learns the sequence of its input and provides a new set of encoding outputs. These experiments also explain that training the model to capture the long-term text in a sequence way and then capture the local features provides better learning than training the model by only capturing local features and training the model with only sequences of text. Training the model by capturing the local features and then learning the long-term dependencies from the local features is inefficient because the CNN layer does not capture the sequentially from text and thus may not provide the best information for the LSTM layer to learn.

C. Impact of CNN and LSTM architecture

This occurs because the features that the LSTM and CNN models learn from come straight from the model's global average pooling layer, where the average of each feature map is calculated, and the resulting vector is inputted into the layer. Because optimizing any parameter within the global average pooling layer is not necessary. Overfitting is less of an issue, and the model's performance is improved because of this one-of-a-kind feature of the global average pooling layer. In the end, it is safe to say that when it comes to online learning sentiment analysis using text data from tweets, the LSTM-CNN and CNN-LSTM models with an oversampling method give a more solid and trustworthy classification model. This happens because the combined LSTM and CNN models' learned features come straight from the model's global average pooling layer, which takes an average of all feature maps and feeds the resulting vector into the model. The global average pooling layer improves the model's performance and overcomes the overfitting issue. Ultimately, it is safe to say that when generalizing LSTM-CNN and CNN-LSTM models to online learning tweets text data for sentiment analysis, the oversampling technique produces a more robust and dependable classification model. Prior to utilizing the dataset in the DL model, it is necessary to standardize every tweet within the dataset to the same format.

E-mail:marina998@uitm.edu.my

TABLE 1. RESULTS FROM THE BEST MODEL OBTAINED BY ALL MODELS WITH 32 BATCH SIZE

	Class Type	Trainable Word Embedding				Without Trainable Word Embedding			
		P	***R***	***F***	***A (%)***	***P***	***R***	***F***	***A (%)***
CNN	0	0.91	0.86	0.88	88.36	0.88	0.82	0.85	85.27
	1	0.81	0.87	0.84		0.78	0.81	0.80	
	2	0.93	0.93	0.93		0.90	0.92	0.91	
	Avg/Total	0.89	0.88	0.88		0.85	0.85	0.85	
LSTM	0	0.94	0.83	0.88	88.70	0.93	0.85	0.89	87.84
	1	0.80	0.90	0.84		0.80	0.86	0.83	
	2	0.94	0.94	0.94		0.92	0.93	0.92	
	Avg/Total	0.89	0.89	0.88		0.88	0.88	0.88	
CNN-LSTM	0	0.93	0.86	0.89	**89.18**	0.94	0.82	0.87	87.53
	1	0.82	0.88	0.85		0.79	0.88	0.83	
	2	0.94	0.94	0.94		0.92	0.93	0.92	
	Avg/Total	0.89	0.89	0.89		0.88	0.88	0.88	
LSTM-CNN	0	0.93	0.83	0.88	87.95	0.92	0.88	0.90	**89.66**
	1	0.79	0.89	0.84		0.85	0.86	0.85	
	2	0.94	0.92	0.93		0.93	0.96	0.94	
	Avg/Total	0.88	0.88	0.88		0.90	0.90	0.90	
BILSTM	0	0.93	0.82	0.87	87.43	0.93	0.85	0.89	87.81
	1	0.78	0.88	0.83		0.79	0.87	0.83	
	2	0.93	0.93	0.93		0.92	0.92	0.92	
	Avg/Total	0.88	0.88	0.88		0.88	0.88	0.88	
CNN-BILSTM	0	0.91	0.85	0.88	87.81	0.93	0.83	0.87	87.47
	1	0.80	0.85	0.83		0.79	0.87	0.83	
	2	0.93	0.93	0.93		0.92	0.93	0.92	
	Avg/Total	0.88	0.88	0.88		0.88	0.87	0.88	
2 LAYER CNN	0	0.92	0.84	0.88	88.02	0.90	0.83	0.87	85.95
	1	0.80	0.88	0.84		0.78	0.84	0.81	
	2	0.94	0.92	0.93		0.90	0.91	0.90	
	Avg/Total	0.88	0.88	0.88		0.86	0.86	0.86	

E-mail:marina998@uitm.edu.my

TABLE 2. RESULTS FROM THE BEST MODEL OBTAINED BY ALL MODELS WITH 64 BATCH SIZE

DL Model	Trainable Word Embedding					Without Trainable Word Embedding			
	Class Type	***P***	***R***	***F***	***A (%)***	***P***	***R***	***F***	***A (%)***
CNN	0	0.91	0.86	0.89	88.05	0.90	0.81	0.85	85.03
	1	0.80	0.87	0.83		0.76	0.85	0.80	
	2	0.94	0.91	0.92		0.91	0.90	0.90	
	Avg/Total	0.88	0.88	0.88		0.86	0.85	0.85	
LSTM	0	0.95	0.83	0.88	88.39	0.93	0.85	0.89	88.11
	1	0.79	0.89	0.84		0.81	0.85	0.83	
	2	0.93	0.94	0.93		0.91	0.94	0.92	
	Avg/Total	0.89	0.89	0.89		0.88	0.88	0.88	
CNN-LSTM	0	0.93	0.85	0.89	**89.05**	0.94	0.81	0.87	87.05
	1	0.81	0.89	0.85		0.78	0.88	0.83	
	2	0.94	0.93	0.93		0.91	0.94	0.93	
	Avg/Total	0.89	0.89	0.89		0.88	0.87	0.87	
LSTM-CNN	0	0.92	0.84	0.88	87.74	**0.92**	**0.88**	**0.90**	89.00
	1	0.78	0.88	0.83		0.83	0.86	0.84	
	2	0.94	0.91	0.93		0.92	0.93	0.93	
	Avg/Total	0.88	0.88	0.88		0.89	0.89	0.89	
BILSTM	0	0.92	0.81	0.87	86.71	0.92	0.87	0.89	88.00
	1	0.77	0.86	0.82		0.81	0.84	0.83	
	2	0.92	0.93	0.92		0.91	0.92	0.92	
	Avg/Total	0.87	0.87	0.87		0.88	0.88	0.88	
CNN-BILSTM	0	0.91	0.84	0.88	88.02	0.92	0.86	0.89	88.02
	1	0.80	0.86	0.83		0.81	0.85	0.83	
	2	0.93	0.94	0.93		0.91	0.93	0.92	
	Avg/Total	0.88	0.88	0.88		0.88	0.88	0.88	
2 LAYER CNN	0	0.93	0.81	0.86	87.77	0.88	0.83	0.86	85.17
	1	0.79	0.89	0.83		0.79	0.81	0.80	
	2	0.94	0.94	0.94		0.89	0.92	0.91	
	Avg/Total	0.88	0.88	0.88		0.86	0.85	0.85	

E-mail:marina998@uitm.edu.my

D. Effect of batch sizes

The batch size defines the number of samples that will be transmitted through the network. For gradient descent, the batch size is a hyperparameter that determines how many training samples must be processed before the model's internal parameters are changed. Batch sizes 32 and 64 are commonly used in mini-batch gradient descent. Gradient descent's batch size hyperparameter determines how much training data is processed before updating the model's internal parameters. An example of a batch would be a for-loop that makes predictions by iterating over a set of samples. The batch is completed by comparing the predictions to the expected output variables and calculating the error. The update procedure is employed to enhance the model based on this error, for instance, by descending the error gradient.

E. Effect of balanced and imbalanced data

Problems with unbalanced data have sparked new ideas for how to fix them, particularly in machine learning and deep anchoring, which could improve the way text data behavior is classified. By duplicating examples from the minority class in the training dataset, Random oversampling addresses the imbalanced dataset problem and, for some models, reduces overfitting. Nevertheless, the hybrid CNN and LSTM variations model outperformed the classic machine learning model. At long last, the suggested classifier model, which combines CNN and LSTM versions, can be considered a cutting-edge method for text classification. When compared to another deep learning-based model, the results show that CNN-LSTM and LSTM-CNN outperform it in terms of classification accuracy.

5. CONCLUSION

Many aspects of online learning that affect how academic institutions and students engage with one another must be clarified. As a result, studying online student feedback has made the creation of an effective perception categorization and computing model a highly significant subject. Traditional machine learning and the current sentiment analysis approaches should focus more on keyword polarity classification and be more accurate regarding other forms of sentiment. Consider the problems with the current sentiment analysis methodology, such as its inability to scale and identify features. We conducted fine-grained sentiment classification and suggested hybrid CNN-LSTM and LSTM-CNN architectures to calculate public perceptions of online learning based on dependency parsing. Word embeddings provide input to our model, using convolutional layers to extract local characteristics. After the convolutional model, the output is passed to an LSTM model to understand the word sequence's long-term relationships. Lastly, a classifier layer is applied. Following the LSTM-CNN, BILSTM, CNN-BILSTM, a single LSTM, two layers of CNN, and a single CNN model based on accuracy, recall rate, and F-value for classifying the perception tendency of Twitter texts were the models shown in the simulation experiments. The small sample size and variety of emotions require additional work to ensure that our findings apply to a broad audience. Future research will look at additional types of cases, how different perceptions change throughout online learning, and how to optimize the best-fit model for feature selection. The goal is to learn perception based on dependency parsing and perform fine-grained sentiment classification.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support from the Universiti Teknologi MARA (UiTM), the Institute for Big Data Analytics and Artificial Intelligence (IBDAAI), Universiti Teknologi MARA, Malaysia.

REFERENCES

[1] Omar, M. S., Njeru, A., & Yi, S. (2017, June). The influence of twitter on education policy making. In *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)* (pp. 133-136). IEEE.

[2] Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), e1253.

[3] Clarizia, F., Colace, F., De Santo, M., Lombardi, M., Pascale, F., & Pietrosanto, A. (2018, January). E-learning and sentiment analysis: a case study. In *Proceedings of the 6th international conference on information and education technology* (pp. 111-118).

[4] Aung, K. Z., & Myo, N. N. (2017, May). Sentiment analysis of students' comment using lexicon based approach. In *2017 IEEE/ACIS 16th international conference on computer and information science (ICIS)* (pp. 149-154). IEEE.

[5] Dhaoui, C., Webster, C. M., & Tan, L. P. (2017). Social media sentiment analysis: lexicon versus machine learning. *Journal of Consumer Marketing*.

[6] Onan, A. (2021). Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. *Concurrency and Computation: Practice and Experience*, *33*(23), e5909.

[7] Ghosh, S., Chakraborty, P., Cohn, E., Brownstein, J. S., & Ramakrishnan, N. (2016, October). Characterizing diseases from unstructured text: A vocabulary driven word2vec approach.

E-mail:marina998@uitm.edu.my

In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (pp. 1129-1138).

[8] Agarwal, N., Sikka, G., & Awasthi, L. K. (2020). Enhancing web service clustering using Length Feature Weight Method for service description document vector space representation. *Expert Systems with Applications*, *161*, 113682. Ali, F., Kwak, D., Khan, P., El-Sappagh, S., Ali, A., Ullah, S., ... & Kwak, K. S. (2019). Transportation sentiment analysis using word embedding and ontology-based topic modeling. *Knowledge-Based Systems*, *174*, 27-42.

[9] Chen, Q., & Sokolova, M. (2021). Specialists, scientists, and sentiments: Word2Vec and Doc2Vec in analysis of scientific and medical texts. *SN Computer Science*, *2*(5), 1-11.

[10] Jain, K., & Kaushal, S. (2018, August). A comparative study of machine learning and deep learning techniques for sentiment analysis. In *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 483-487). IEEE.

[11] Adak, A., Pradhan, B., & Shukla, N. (2022). Sentiment Analysis of Customer Reviews of Food Delivery Services Using Deep Learning and Explainable Artificial Intelligence: Systematic Review. *Foods*, *11*(10), 1500.

[12] Rodrigues, J. C., Rodrigues, J. T., Gonsalves, V. L. K., Naik, A. U., Shetgaonkar, P., & Aswale, S. (2020, February). Machine & deep learning techniques for detection of fake reviews: A survey. In *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)* (pp. 1-8). IEEE.

[13] Lin, B., Zampetti, F., Bavota, G., Di Penta, M., Lanza, M., & Oliveto, R. (2018, May). Sentiment analysis for software engineering: How far can we go?. In *Proceedings of the 40th international conference on software engineering* (pp. 94-104).

[14] Alharbi, A. S. M., & de Doncker, E. (2019). Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information. *Cognitive Systems Research*, *54*, 50-61.

[15] Priyadarshini, I., & Cotton, C. (2021). A novel LSTM–CNN–grid search-based deep neural network for sentiment analysis. *The Journal of Supercomputing*, *77*(12), 13911-13932.

[16] Colón-Ruiz, C., & Segura-Bedmar, I. (2020). Comparing deep learning architectures for sentiment analysis on drug reviews. *Journal of Biomedical Informatics*, *110*, 103539.

[17] Kumar, N., & Yadav, S. C. (2023). A Hybrid Schema: LSTM-BiLSTM with Attention Mechanism to Predict Emotion in Twitter Data. *International Journal of Computing and Digital Systems*, *14*(1), 1-xx.

[18] Venugopalan, M., Gupta, D., & Bhatia, V. (2021). A supervised approach to aspect term extraction using minimal robust features for sentiment analysis. In *Progress in Advanced Computing and Intelligent Engineering* (pp. 237-251). Springer, Singapore.

[19] Saad, Shihab Elbagir, and Jing Yang. "Twitter sentiment analysis based on ordinal regression." *IEEE Access* 7 (2019): 163677-163685.

[20] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

[21] Duan, Feng, Shuai Zhang, Yinze Yan, and Zhiqiang Cai. "An Oversampling Method of Unbalanced Data for Mechanical Fault Diagnosis Based on MeanRadius-SMOTE." *Sensors* 22, no. 14 (2022): 5166.

[22] Elreedy, D., & Atiya, A. F. (2019). A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance. *Information Sciences*, *505*, 32-64.

[23] Hossain, Eftekhar, Omar Sharif, Mohammed Moshiul Hoque, and Iqbal H. Sarker. "Sentilstm: a deep learning approach for sentiment analysis of restaurant reviews." In *International Conference on Hybrid Intelligent Systems*, pp. 193-203. Springer, Cham, 2020.

[24] Cambria, Erik, Soujanya Poria, Devamanyu Hazarika, and Kenneth Kwok. "SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings." In *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1. 2018.

[25] Liao, Shiyang, Junbo Wang, Ruiyun Yu, Koichi Sato, and Zixue Cheng. "CNN for situations understanding based on sentiment analysis of twitter data." *Procedia computer science* 111 (2017): 376-381.

[26] Brownlee, Jason. *Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python*. Machine Learning Mastery, 2020.

[27] Reddy, Dyapa Sravan, Lakshmi Prasanna Reddy, Kandibanda Sai Santhosh, and Virrat Devaser. "SEO Ministration: Content Related Tag Suggestion Using One vs Rest." (2021).

[28] Kim, Tae-Young, and Sung-Bae Cho. "Particle swarm optimization-based CNN-LSTM networks for forecasting energy consumption." In *2019 IEEE congress on evolutionary computation (CEC)*, pp. 1510-1516. IEEE, 2019.

Marina Yusoff is currently a deputy director and senior fellow researcher at the Institute for Big Data Analytics and Artificial Intelligence (IBDAAI) and Associate Professor of Computer and Mathematical Sciences, Universiti Teknologi MARA Shah Alam, Malaysia. She has a Ph.D. in Information Technology and Quantitative Sciences (Intelligent Systems). She previously worked as a Senior Executive of Information Technology in SIRIM Berhad, Malaysia. She is most interested in multidisciplinary research, artificial intelligence, nature-inspired computing optimization, and data analytics. She applied and modified AI methods in many research and projects, including recent hybrid deep learning, particle swarm optimization, genetic algorithm, ant colony, and cuckoo search for many real-world problems, medical and industrial projects. Her recent projects are data analytic optimizer, audio, and image pattern recognition. She has many impact journal publications and contributes as an examiner and reviewer to many conferences, journals, and universities' academic activities.

Muhammad Syamil Ali, a dedicated Data Associate at 2x Marketing, is committed to enhancing business marketing outcomes through his expertise in data processes and analytics. Holding a degree in Intelligent System Engineering, Syamil actively applies his skills to bridge gaps, ensuring a steadfast commitment to work quality for improved business results.

Nor Hapiza Mohd Ariffin obtained a Bachelor of Science (Hons) degree in Computer Science in 1994, a master's degree in information technology in 2001, and a Ph. D in Information Systems in 2010 from Universiti Kebangsaan Malaysia. She is an Assistant Professor at the Faculty of Business, Sohar University, Oman. She has published several indexed articles and books nationally and internationally on e-learning, information systems, blockchain, and artificial intelligence. She is currently teaching various topics on e-business, systems analysis, and project management.

Mohammed Adam Kunna Azrag received the B.Sc. degree (Hons.) in computer studies from the National Ribat University (Sudan) in 2010, and the M.Sc. degree in computer science from Universiti Malaysia Pahang in 2015, and the Ph.D. degree in computer science from Universiti Malaysia Pahang in 2021. His research interests include optimization algorithms, data analysis, machine learning, bioinformatic, and data base. He is currently working at the Institute for big data analytics and artificial intelligence (IBDAAI), UiTM Shah Alam, Malaysia.