Sentiment analysis of Indonesian hotel reviews: from classical machine learning to deep learning

Retno Kusumaningrum [a,1,*], Iffa Zainan Nisa [a,2], Rizka Putri Nawangsari [a,3], Adi Wibowo [a,4]

[a] Department of Informatics, Universitas Diponegoro, Semarang, Indonesia
[1] retno@live.undip.ac.id; [2] iffazainannisa@students.undip.ac.id; [3] rizkaputrinawangsari@gmail.com; [4] bowo.adi@live.undip.ac.id
* corresponding author

ARTICLE INFO

Article history
Received September 7, 2021
Revised October 13, 2021
Accepted October 13, 2021
Available online November 30, 2021

Keywords
Sentiment Analysis
Word2Vec
Convolutional Neural Network
Classical Machine Learning
Hotel Reviews

ABSTRACT

Currently, there are a large number of hotel reviews on the Internet that need to be evaluated to turn the data into practicable information. Deep learning has excellent capabilities for recognizing this type of data. With the advances in deep learning paradigms, many algorithms have been developed that can be used in sentiment analysis tasks. In this study, we aim to compare the performance of classical machine learning algorithms—logistic regression (LR), naïve Bayes (NB), and support vector machine (SVM) using the Word2Vec model in conjunction with deep learning algorithms such as a convolutional neural network (CNN) to classify hotel reviews on the Traveloka website into positive or negative classes. Both learning methods apply hyperparameter tuning to determine the parameters that produce the best model. Furthermore, the Word2Vec model parameters use the skip-gram model, hierarchical softmax evaluation, and the value of 100 vector dimensions. The highest average accuracy obtained was 98.08% by using the CNN with a dropout of 0.2, Tanh as convolution activation, softmax as output activation, and Adam as the optimizer. The findings from the study demonstrate that the integration of the Word2Vec model and the CNN model obtains significantly better accuracy than other classical machine learning methods.

1. Introduction

The rapid evolution of the Internet and information technology has resulted in extensive advances in e-commerce platforms such as online travel agencies (OTAs). Traveloka is an OTA that specializes in hotel reservations, airline tickets, train tickets, and other complementary services. Furthermore, Traveloka enables users to rate and review the services they have booked through the site. The reviews can be used as an indicator of the quality of services and a source of information for the service provider [1], [2]. However, a large number of reviews make analysis difficult for service providers. Therefore, sentiment analysis is required to process the data and analyze existing reviews to classify them into positive and negative reviews automatically.

Sentiment analysis is a technique that examines user thoughts expressed through various media, including print media, social media, blogs, and websites, to determine user satisfaction and their perception of an issue [3]. Sentiment analysis of reviews is used in various contexts, such as product reviews [4], mobile applications [5], films [6], travel destinations [7], restaurants [8], and hotels. Sentiment analysis is performed at tiered levels, namely document-level [9]–[11], sentence-level [12]–[14], and aspect-level [15]–[17]. The document-level analysis provides the benefits of presenting a

comprehensive polarity and evaluating a larger amount of material [18]. As a result, we investigate the sentiment analysis model of hotel reviews at the document level.

Most sentiment analysis studies use English as the target language. However, as Indonesian and English have different syntax, it is important to understand how utilizing Indonesian in hotel reviews affects the reader. There have been numerous studies on sentiment analysis of texts in the Indonesian language. The majority of these studies use classical machine learning techniques such as naïve bayes (NB) [13][19]–[21], logistic regression (LR) [19], k-nearest neighbors (KNN) [20], maximum entropy (MaxEnt) [21], and support vector machine (SVM) [19]–[21]. The NB model outperformed KNN and SVM in the personality classification task conducted by [20]. However, by applying the synthetic minority oversampling technique (SMOTE) in each class, LR was transformed into a superior model with a g-mean score of 81.65% [19]. SMOTE can overcome unequal-dataset problems. Moreover, it is challenging to choose a specific feature extraction to include in a given model using classical machine-learning methods. The model generates imperfect results if features are missing or incomplete and causes problems if there are too many features [11].

While deep learning generally performs appropriately for a large set of data, it is not effective when the dataset is small. A CNN was used as the deep learning approach in sentiment analysis of the Indonesian language [22], [23]. In comparison to long short-term memory (LSTM), gated recurrent unit (GRU), and recurrent neural network (RNN) algorithms, the CNN model can locate important features from the varied meanings of the word and has a faster computing time [24].

When examining an object's properties, unique attributes that can be referred to as features are given. Both classical and deep learning methods are unable to analyze input data in the form of text or strings, necessitating the use of numbers as inputs [11]. If the received object is a string or text, feature extraction is required to turn the text into vectors that represent a word, especially in the word embedding process.

Word2Vec is an alternative method for generating vector spaces from a corpus. One of the most significant advantages of the Word2Vec model is that it represents characteristics as dense vectors rather than sparse ones that allow it to overcome the problem of synonyms and homonyms, which is common in natural language processing problems [25]. Based on research conducted by [25], for Indonesian reviews, the Word2Vec model for sentiment analysis of hotel reviews shows the best accuracy in the skip-gram model architecture, hierarchical softmax for evaluation methods, and a value of 100 for vector dimensions. Therefore, based on previous research, this study aims to validate the efficacy of a CNN when compared to classical machine learning algorithms, such as LR, NB, and SVM, using the Word2Vec model.

2. Method

This study is divided into five fundamental processes: data collection, preprocessing, word embedding, classification model generation, and performance evaluation. The process of classification model generation uses several classical machine learning and deep learning methods. The research methodology used in this study is shown in Fig. 1.

2.1. Data Collection

The data used in this study were obtained from the hotel review data of the Traveloka website. Reviews of hotels in several cities were crawled using Scrapy (https://scrapy.org) and Selenium (https://selenium-python.readthedocs.io) libraries, and over 2500 hotel reviews of such crawled data were used in this study. These data were then manually labeled as positive and negative reviews, with a total of 1250 for each label. Data labeling is determined by considering the composition of positive and negative words in the review based on the document level. Some examples of labeled data are listed in Table 1.

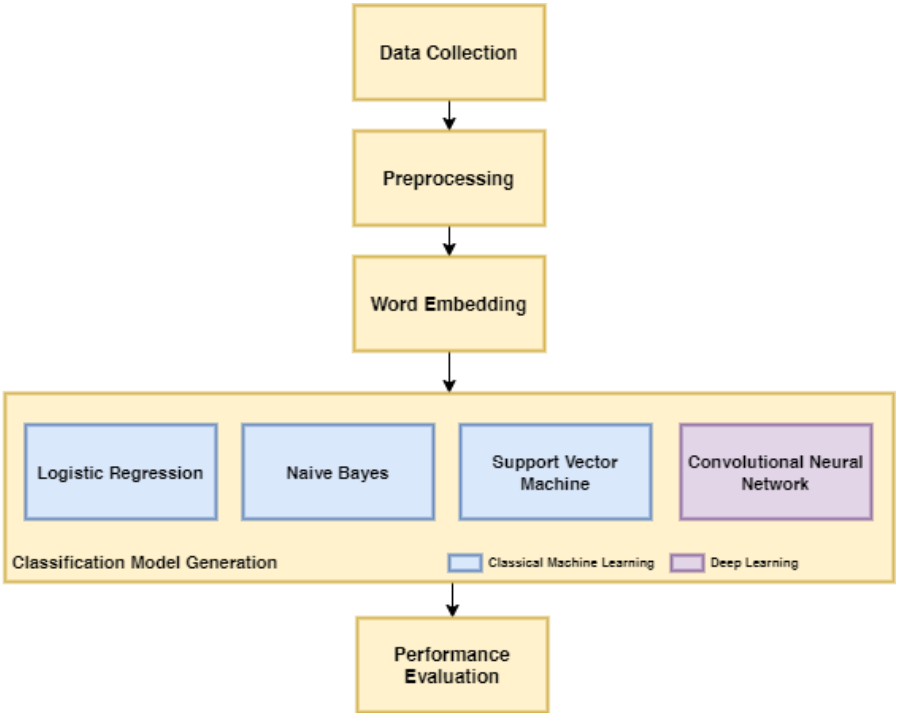

Fig. 1. Research Methodology

Table 1. Examples of labeled data

Review	Label
The bed and pillows stink ("*Tempat tidur dan bantalnya bau*")	Negative
WiFi and hot water are off, even though the conditions there are very cold ("*WiFi dan air panas mati, padahal kondisi disana sangat dingin*")	Negative
The atmosphere is cool, the facilities are ok, thank you, next time I will definitely come here again ("*Suasananya adem, fasilitas ok, thanks ya, next pasti kesini lagi*")	Positive
The service is okay, the view is ok, the rooms are ok, the location is a bit inside so prepare for food, especially if you bring children ("*Pelayanan oke, view ok, kamar ok, lokasi agak masuk ke dalam jadi persiapkan makanan apalagi klo membawa anak-anak*")	Positive

2.2 Preprocessing

Preprocessing is the cleaning and preparation of the text for analysis [26]. Preprocessing in this study comprises case folding, tokenization, stop-word removal, stemming, and padding.

a. Case folding is used to uniformize all of the text into a lowercase form to make processing easier [27]. In this study, the procedure used for converting all characters into lowercase letters is well-structured and straightforward. There are no unstructured words or sentences that contain a mixture of lowercase and uppercase characters.

b. The tokenization process eliminates punctuation, tags, emoticons, and numbers from each sentence to produce an array of words. Dots (.), commas (,), special characters (!), etc., are used to remove punctuation. While emoticons are included as symbols, numbers are removed, and each sentence is segmented into an array of words that are then stored.

c. Stop-word removal is a process used to remove words that appear frequently but do not have a specific meaning [28]. The stop-word dictionary was stored in a flat-file with an extension (.txt) during the removal process. The list of words was obtained from https://github.com/stopwordsiso/stopwords-

id. The stop-words dictionary in this application is adapted to the needs of the system while extracting information so that the information extraction process runs optimally.

d. Stemming is the process of removing affixes to obtain a basic word. For example, the words 'offered', 'offering', and 'offer' will change to the basic form of 'offer' after going through the stemming process. In this study, a literature stemmer was used.

e. The padding process adds the word "<PAD />" to all documents whose length is less than the maximum length of the longest document in the array of stemming results. The addition of the word "</ PAD>" ensures that all documents to be processed have the same length and facilitate the next process.

2.3 Word Embedding Using Word2Vec Model

To produce a bag of words (BoW), the repository is converted into a vector shape. To convert the bag of word representations into vectors, the Word2Vec model is used. This study uses the best Word2Vec model based on the research performed by [25] using the following parameters: the skip-gram architecture model, the hierarchical softmax evaluation method, and 100 vector dimensions.

2.4 Classification Model Generation

In this study, several classical machine learning and deep learning methods were applied. The classical machine learning methods used were Logistic Regression (LR), Naïve Bayes (NB), and Support Vector Machine (SVM), and the deep learning method used was a Convolutional Neural Network (CNN). LR is an algorithm derived from a linear regression-based approach for predicting probability-dependent variables. This method performs well for binary classification tasks [29]. A logistic function or sigmoid function is used in this method to map a predicted value to a probabilistic number between 0 and 1. The probability (p) of the review is determined as positive (+1) or negative class (-1) based on the input (x) and learn coefficient (w). This calculation involves exponential (e) raised to the power of the dot product of the transpose learn coefficient from each feature (h) in the given input.

The NB classifier is a classification that uses the probability method proposed by Thomas Bayes. This method is used to predict future opportunities based on past experience. Compared to other methods, although this method is relatively simple and effective, it is rather sensitive to feature selection [30]. The SVM, introduced by Vapnik, is a method used to create hyperplanes in certain dimensional spaces. The purpose of this method is to find the best hyperplane that provides the maximum margin distance [30]. CNNs are deep learning techniques that are typically used for classification, segmentation, and object detection. The CNN architecture used in this study is shown in Fig. 2.

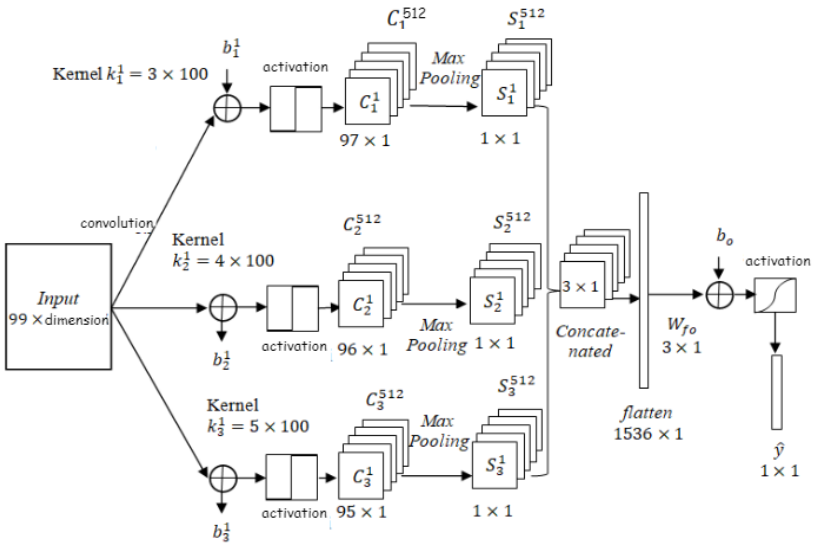

Fig. 2. CNN Architecture

The input of the CNN architecture is an array of BoW preprocessing results that have been processed using the Word2Vec model with a size of 99 × 100. Then, the input is processed in a convolution layer that uses the filter size = (3, 4, and 5) and activation of the linear unit rectifier (ReLU) and tanh convolution, taking into consideration the combination of parameters being tested. C_1^1 is the result of the input that has been convoluted with a size filter of 3 × dimensions, resulting in a size of 97 × 1 with a total of 512 features. C_2^1 is the result of a convoluted input with a 4 × dimension size filter, resulting in a size of 96 × 1 with a total of 512 features. C_3^1 is the result of a convoluted input with a 5 × dimension size filter, resulting in a size of 95 × 1 with a total of 512 features.

The results of $\boldsymbol{C_1^1}$, $\boldsymbol{C_2^1}$, and $\boldsymbol{C_3^1}$ are carried out by the pooling process with the max-pooling function and the size of the pool size, 97 × 1, 96 × 1, and 95 × 1, respectively, to produce a 1 × 1 matrix with 512 features. The result of the pooling layer is concatenated so that it measures 3 × 1, and the number of features is 512. The results of the concatenate are flattened such that the shape changes into a vector so that it can be processed in the fully connected layer. In the fully connected layer using dropout values of 0.2, 0.5, and 0.7, the activation of softmax and sigmoid output is dependent on the combination of parameters being tested. The number of output nodes was adapted to the combination of the parameters being tested.

2.5 Evaluation

K-fold cross-validation was used with a fold value of 10 to divide the data into training and testing. 10-Fold Cross-validation can prevent biased values of performance metrics [13]. The data were divided into training and testing data. Training data were used to conduct the training process to generate a classification model that was later tested using the testing data. This process was repeated until the fold value reached 10. The performance of the models was evaluated using the accuracy, precision, recall, and f-measure. These results were compared and analyzed using several classification model scenarios generated using classical machine learning and deep learning methods.

3. Results and Discussion

3.1. Research Data

This study used data from over 2500 hotel reviews. The details of the data distribution were as follows: 1250 positive-labeled review data and 1250 negative-labeled review data. Review data were obtained using the crawling process on the Traveloka website. This study was implemented using Python 3 with the help of the library for training the Word2Vec model and the library for the formation of the LR, NB, SVM, and CNN methods. In addition, hyperparameter tuning in classical machine learning also uses the GridSearchCV library. The performance of the model was measured using K-fold cross-validation. The number of K used was 10 such that the training data was divided by 9-fold and the remaining one was a test with the number of classes balanced on each fold.

3.2. Research Scenarios

Three scenarios were designed to achieve the research objectives. In Scenario 1, classical machine learning was implemented, including LR, NB, and SVM. In Scenario 2, the CNN was implemented using three distinct architectures. Subsequently, the outcomes of scenarios 1 and 2 were compared in scenario 3.

a. Hyperparameter Tuning for Scenario 1

As mentioned previously, the three classical machine learning algorithms used are LR, NB, and SVM. Each model has a different hyperparameter tuning. The hyperparameters of each algorithm are listed in Table 2.

Table 2. Hyperparameters of Classical Machine Learning

LR		NB		SVM	
Parameters	***Values of Parameter***	***Parameters***	***Values of Parameter***	***Parameters***	***Values of Parameter***
solver	newton-cg, lbfgs, liblinear	var_smoothing	Log space (0, -9, 100)	kernel	polynomial, rbf, sigmoid, linear
penalty	none, l1, l2, elasticnet			gamma	1, 0.1, 0.01, 0.001, 0.0001
C	0.1, 1, 10, 100, 1000			C	0.1, 1, 10, 100, 1000

b. Hyperparameter Tuning for Scenario 2

As previously stated, the CNN was implemented using three distinct architectures. Each architecture has a different hyperparameter tuning. The hyperparameters of each architecture are listed in Table 3.

Table 3. Hyperparameters of CNN

Architecture 1		Architecture 2		Architecture 3	
Parameters	***Values of Parameter***	***Parameters***	***Values of Parameter***	***Parameters***	***Values of Parameter***
dropout	0.2, 0.5, 0.7	dropout	0.2, 0.5, 0.7	dropout	0.2, 0.5, 0.7
convolutional activation	ReLU, Tanh	convolutional activation	ReLU, Tanh	convolutional activation	ReLU, Tanh
output activation	sigmoid	output activation	sigmoid, softmax	output activation	sigmoid, softmax
optimizer	SGD	optimizer	SGD	optimizer	Adam
number of outputs	one node	number of outputs	two nodes	number of outputs	two nodes

3.3. Results and Analysis

a. Scenario 1

In scenario 1, the test results for hyperparameter tuning using the GridSearchCV library are listed in Table 4. The best model of the LR method was obtained when using newton-cg for the solver parameter, l2 for the penalty, and C with a value of 100, resulting in a 54.4% of accuracy. Meanwhile, the best model of the NB method was obtained when using var smoothing with a value of 0.285, resulting in a 53.8% of accuracy. And lastly, the best model of the SVM method was obtained when using the rbf kernel, gamma value 1, and C with a value of 1000, resulting in a 54.2% of accuracy. LR is the optimal model in scenario 1, as evident from its accuracy, precision, recall, and f-measure when compared to NB and SVM.

Table 4. Test Results in Scenario 1

Methods	Accuracy	Precision	Recall	F-Measure	Best Parameters
LR	**54.4%**	**55%**	**54%**	**54%**	Solver: 'newton-cg'; Penalty: 'l2'; C: 100
NB	53.8%	54%	52%	47%	Var_smoothing: 0.285
SVM	54.2%	54%	53%	52%	Kernel: 'rbf'; Gamma: 1; C: 1000

b. Scenario 2

In Scenario 2, the test results for hyperparameter tuning are listed in Table 5. For Architecture 1 (CNN 1), the highest accuracy was 95.68%. This value was obtained by using a dropout value of 0.2 and the activation of the Tanh convolution. Meanwhile, the highest accuracy for Architecture 2 (CNN 2)

was 94.08%, which was obtained using a 0.2 dropout value, Tanh convolution activation, and softmax output activation. Lastly, the highest accuracy for Architecture 3 (CNN 3) was 98.08%. This value was obtained using a dropout value of 0.2, the Tanh convolution's activation, and the softmax output's activation.

Table 5. Test Results in Scenario 2

Methods	Accuracy	Precision	Recall	F-Measure	Best Parameters
CNN 1	95.68%	95.72%	95.68%	95.68%	Dropout: 0.2 Conv. Activation: Tanh Output activation: Sigmoid Optimizer: SGD Number of outputs: one node
CNN 2	94.08%	94.48%	94.08%	94.07%	Dropout: 0.2 Conv. Activation: Tanh Output activation: Sigmoid Optimizer: SGD Number of outputs: two nodes
CNN 3	**98.08%**	**98.09%**	**98.08%**	**98.08%**	Dropout: 0.2 Conv. Activation: Tanh Output activation: Softmax Optimizer: Adam Number of outputs: two nodes

The effect of changing the CNN parameter on the performance evaluation values (accuracy, precision, recall, and f-measure) is described in further detail.

1) Effect of Dropout on Performance Evaluation Values

Based on the graph in Fig. 3, it can be concluded that dropout is directly proportional to the performance evaluation values, whereby the greater the dropout, the smaller the performance evaluation values obtained. This is due to the fact that the greater the dropout value, the more units are wasted, such that more semantic meaning of the text is lost.

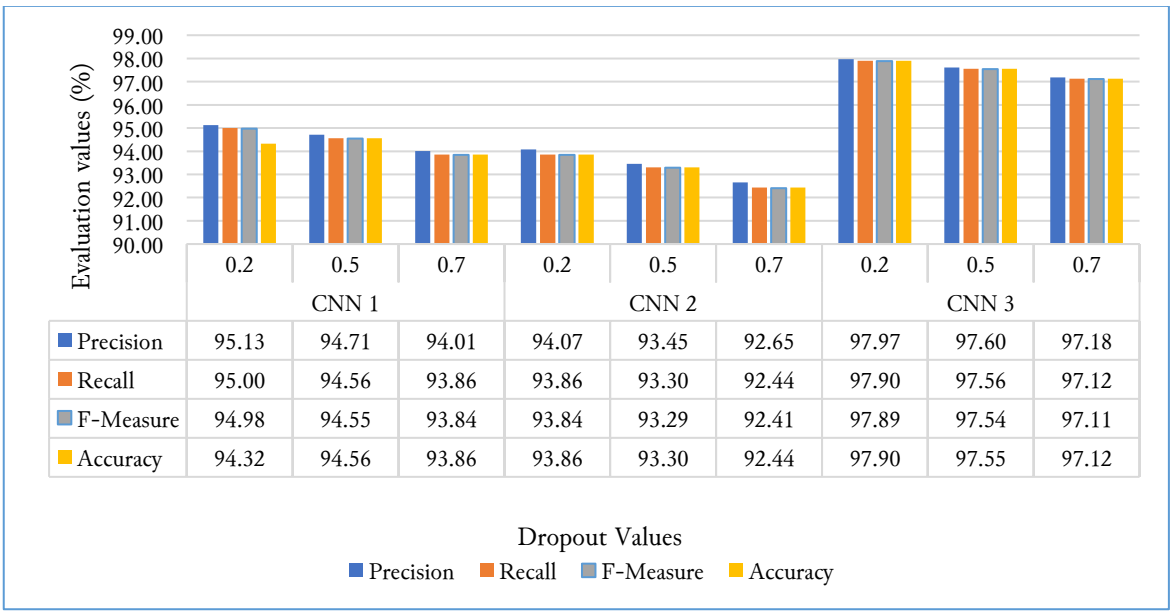

Fig. 3. Effect of Dropout on Performance Evaluation Values

2) Effect of Convolution Activation on Performance Evaluation Values

Fig. 4 shows that Tanh generates better performance evaluation values when compared to ReLU. This is due to the fact that ReLU turns off several units by giving a function gradient = 0 so that when updating the weight, it is less than optimal.

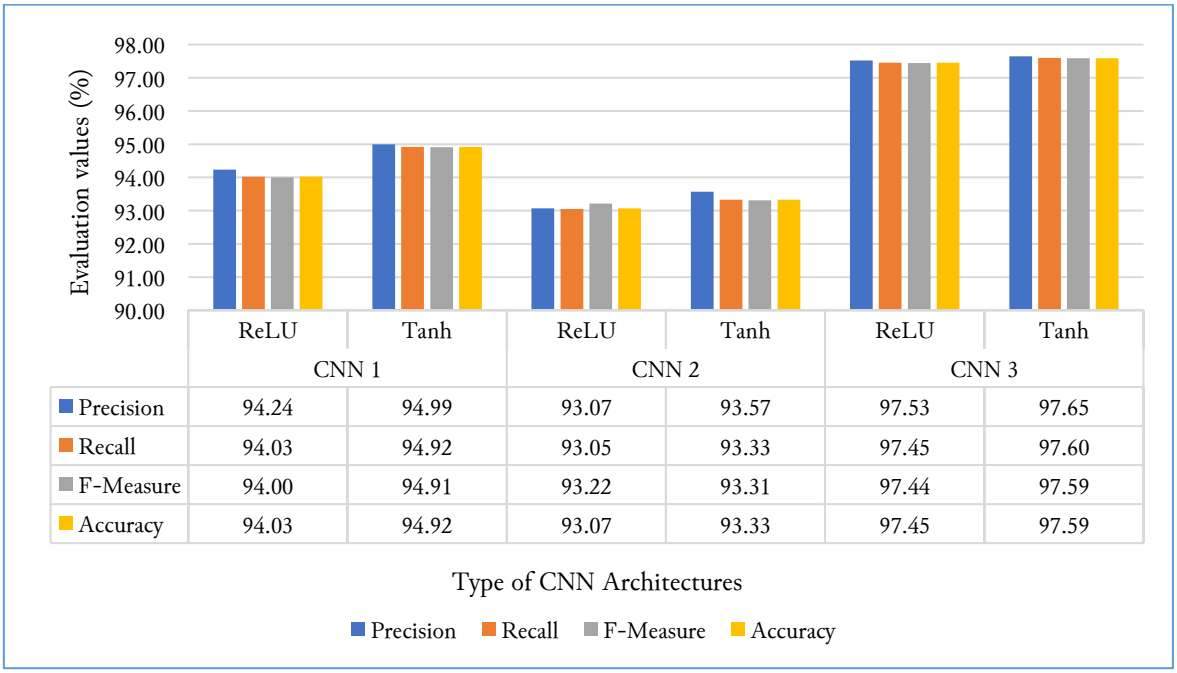

Fig. 4. Effect of Convolution Activation on Performance Evaluation Values

3) Effect of Output Activation on Performance Evaluation Values

Based on the graph in Fig. 5, it is evident that the performance evaluation values from the activation of the softmax output are better than sigmoid. This is since softmax is better at classifying multi-classification than binary classification, whereas CNN's 2 and 3 use multiple output nodes (2).

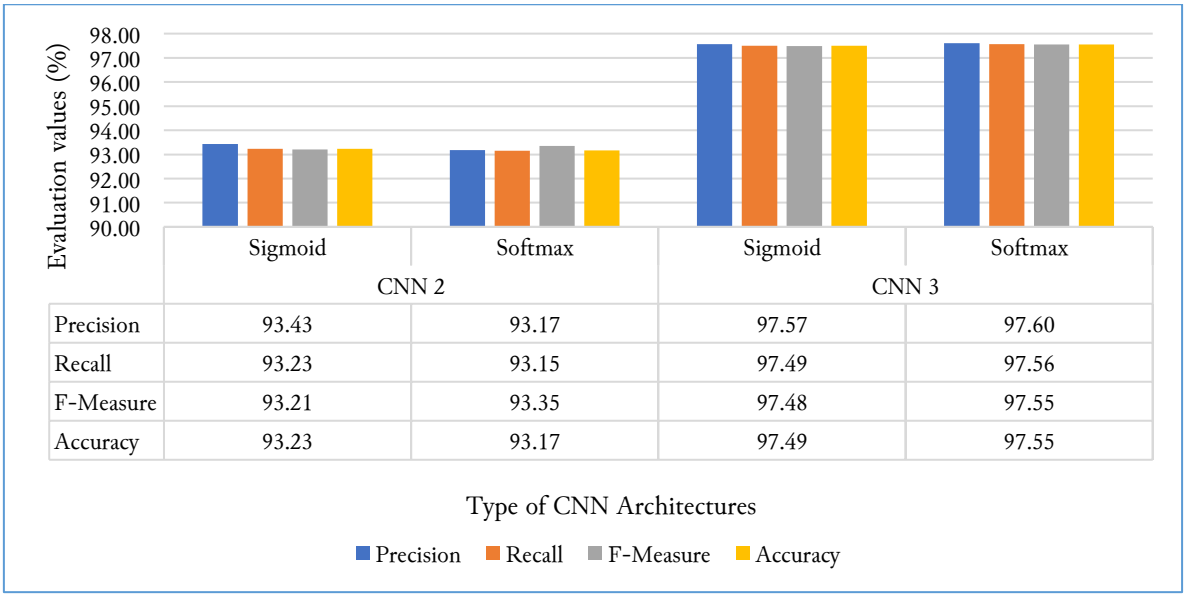

Fig. 5. The Effect of Output Activation on Performance Evaluation Values

c. Scenario 3

Based on the best performance evaluation values from scenarios 1 and 2, it can be concluded that CNN 3 produces a higher performance evaluation value than LR. This proves that the deep learning method outperforms classical machine learning in the sentiment classification task of hotel reviews in Indonesian because deep learning methods learn deeper during the mapping of raw input data into feature representation by utilizing a layered algorithmic structure. This comparison is presented in Table 6. Classical machine learning produces imperfect results due to the method's incomplete or too many features.

Table 6. Test Results in Scenario 3

	Methods	Accuracy	Precision	Recall	F-Measure
Scenario 1	LR	54.4%	55%	54%	54%
Scenario 2	CNN 3	98.08%	98.09%	98.08%	98.08%

Furthermore, CNN 3's Performance evaluation in terms of polarity, which is positive or negative, is presented in Table 7. It can be seen that negative reviews have better recall and f-measure values than positive reviews, although they are not too far away. This shows that negative reviews tend to be well predicted by this model. Some misclassifications occur in reviews that have a balanced comparison of words containing positive and negative meanings, such as "Old hotel, spacious rooms, standard cleanliness only" ("*Hotel tua, kamar luas, kebersihan standar saja*"), "Good only for cleanliness can be improved" ("*Bagus cuma untuk kebersihan bisa ditingkatkan*"), "Strategic hotel location. Quite comfortable and clean, only the room is relatively small." ("*Lokasi hotel strategis. Lumayan nyaman dan bersih, cuman ruangannya relatif kecil*"). Therefore, further research can be developed by adding a neutral class.

Table 7. CNN 3's Performance evaluation in terms of polarity

	Precision	Recall	F-Measure	Accuracy
Positive	98.18%	97.92%	98.04%	
Negative	98%	98.24%	98.11%	98.08%
Average	98.09%	98.08%	98.08%	

4. Conclusion

This study centered on measuring the efficacy of a CNN at sentiment analysis when compared to classical machine learning algorithms, such as LR, NB, and SVM, using the Word2Vec model. Based on the test results presented, it can be concluded that the CNN deep learning method outperforms the classical machine learning method with an accuracy of 98.08%. This is attributable to deep learning methods that learn deeper during the mapping of raw input data into feature representation by utilizing a layered algorithmic structure, whereas classical machine learning generates imperfect results because of incomplete or too many features generated by the method. The best parameters of the CNN model architecture were obtained using a dropout value of 0.2, activation of the Tanh convolution, activation of softmax output, and Adam's optimizer. This is because a dropout value close to 0 gives better accuracy than a dropout value close to 1; the higher the dropout value, the more units are wasted such that the greater semantic meaning of the text is wasted. As a result, the activation of the convolution Tanh provides a better average value of accuracy compared to the activation of the ReLU convolution when using the stochastic gradient descent (SGD) optimizer. Adam's optimizer is able to improve the accuracy of the SGD optimizer by 4% because Adam generates satisfactory results in practice and is advantageous when compared to other stochastic optimization methods. Furthermore, an increase in the number of output nodes from one node to two nodes can improve the accuracy by 2.4%. This is due to the fact that the greater the number of nodes in the CNN method, the better it is at updating weights that affect the accuracy of the prediction. The use of softmax activation gives better results at the two output nodes

because the formula divides by 1 so that the results range from 0 to 1; therefore, the results are more realistic and are well suited for multi-class classification.

Acknowledgment

The authors thank the Directorate of Research and Development, under the Ministry of Research and Technology/National Agency for Research and Innovation, Indonesia, for supporting this research.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding statement. This work was supported by the Directorate of Research and Development, under the Ministry of Research and Technology/National Agency for Research and Innovation, Indonesia [grant number 257-20/UN7.6.1/PP/2021].

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

[1] N. Akhtar, N. Zubair, A. Kumar, and T. Ahmad, "Aspect based Sentiment Oriented Summarization of Hotel Reviews," in *Procedia Computer Science*, 2017, vol. 115, pp. 563–571, doi: 10.1016/j.procs.2017.09.115.

[2] D. Anand and D. Naorem, "Semi-supervised Aspect Based Sentiment Analysis for Movies Using Review Filtering," in *Procedia Computer Science*, 2016, vol. 84, pp. 86–93, doi: 10.1016/j.procs.2016.04.070.

[3] E. Wahyudi and R. Kusumaningrum, "Aspect Based Sentiment Analysis in E-Commerce User Reviews Using Latent Dirichlet Allocation (LDA) and Sentiment Lexicon," in *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*, 2019, pp. 1–6, doi: 10.1109/ICICoS48119.2019.8982522.

[4] Rahul, V. Raj, and Monika, "Sentiment Analysis on Product Reviews," in *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2019, pp. 5–9, doi: 10.1109/ICCCIS48478.2019.8974527.

[5] Indriati, A. Kusyanti, and D. Zakia, "Sentiment Analysis in the Mobile Application Review Document Using the Improved K-Nearest Neighbor Method," in *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*, 2019, pp. 332–337, doi: 10.1109/SIET48054.2019.8986037.

[6] F. R. Saputra Rangkuti, M. A. Fauzi, Y. A. Sari, and E. D. L. Sari, "Sentiment Analysis on Movie Reviews Using Ensemble Features and Pearson Correlation Based Feature Selection," Nov. 2018, doi: 10.1109/SIET.2018.8693211.

[7] I. P. Windasari and D. Eridani, "Sentiment analysis on travel destination in Indonesia," in *2017 4th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, 2017, pp. 276–279, doi: 10.1109/ICITACEE.2017.8257717.

[8] R. A. Laksono, K. R. Sungkono, R. Sarno, and C. S. Wahyuni, "Sentiment Analysis of Restaurant Customer Reviews on TripAdvisor using Naïve Bayes," in *2019 12th International Conference on Information Communication Technology and System (ICTS)*, 2019, pp. 49–54, doi: 10.1109/ICTS.2019.8850982.

[9] R. Moraes, J. F. Valiati, and W. P. Gavião Neto, "Document-level sentiment classification: An empirical comparison between SVM and ANN," *Expert Systems with Applications*, vol. 40, no. 2, pp. 621–633, Feb. 2013, doi: 10.1016/j.eswa.2012.07.059.

[10] A. Tripathy, A. Anand, and S. K. Rath, "Document-level sentiment classification using hybrid machine learning approach," *Knowledge and Information Systems*, vol. 53, no. 3, pp. 805–831, Dec. 2017, doi: 10.1007/s10115-017-1055-z.

[11] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "Sentiment Analysis Using Word2vec and Long Short-Term Memory (LSTM) for Indonesian Hotel Reviews," in *Procedia Computer Science*, 2021, vol. 179, pp. 728–735, doi: 10.1016/j.procs.2021.01.061.

[12] A. R. Naradhipa and A. Purwarianti, "Sentiment classification for Indonesian message in social media," Apr. 2012, doi: 10.1109/ICCCSN.2012.6215730.

[13] S. Kurniawan, R. Kusumaningrum, and M. E. Timu, "Hierarchical Sentence Sentiment Analysis Of Hotel Reviews Using The Naïve Bayes Classifier," in *2018 2nd International Conference on Informatics and Computational Sciences (ICICoS)*, 2018, pp. 1–5, doi: 10.1109/ICICOS.2018.8621748.

[14] N. Farra, E. Challita, R. A. Assi, and H. Hajj, "Sentence-Level and Document-Level Sentiment Mining for Arabic Texts," Dec. 2010, doi: 10.1109/ICDMW.2010.95.

[15] L. P. Manik *et al.*, "Aspect-Based Sentiment Analysis on Candidate Character Traits in Indonesian Presidential Election," Nov. 2020, doi: 10.1109/ICRAMET51080.2020.9298595.

[16] S. Gojali and M. L. Khodra, "Aspect based sentiment analysis for review rating prediction," Aug. 2016, doi: 10.1109/ICAICTA.2016.7803110.

[17] A. N. Azhar, M. L. Khodra, and A. P. Sutiono, "Multi-label Aspect Categorization with Convolutional Neural Networks and Extreme Gradient Boosting," Jul. 2019, doi: 10.1109/ICEEI47359.2019.8988898.

[18] D. I. Af'idah, R. Kusumaningrum, and B. Surarso, "Long Short Term Memory Convolutional Neural Network for Indonesian Sentiment Analysis towards Touristic Destination Reviews," in *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2020, pp. 630–637, doi: 10.1109/iSemantic50169.2020.9234210.

[19] W. Satriaji and R. Kusumaningrum, "Effect of Synthetic Minority Oversampling Technique (SMOTE), Feature Representation, and Classification Algorithm on Imbalanced Sentiment Analysis," in *2018 2nd International Conference on Informatics and Computational Sciences (ICICoS)*, 2018, pp. 1–5, doi: 10.1109/ICICOS.2018.8621648.

[20] B. Y. Pratama and R. Sarno, "Personality classification based on Twitter text using Naive Bayes, KNN and SVM," in *2015 International Conference on Data and Software Engineering (ICoDSE)*, 2015, pp. 170–174, doi: 10.1109/ICODSE.2015.7436992.

[21] E. Lunando and A. Purwarianti, "Indonesian social media sentiment analysis with sarcasm detection," in *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2013, pp. 195–198, doi: 10.1109/ICACSIS.2013.6761575.

[22] A. Cahyadi and M. L. Khodra, "Aspect-Based Sentiment Analysis Using Convolutional Neural Network and Bidirectional Long Short-Term Memory," in *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, 2018, pp. 124–129, doi: 10.1109/ICAICTA.2018.8541300.

[23] A. Ilmania, Abdurrahman, S. Cahyawijaya, and A. Purwarianti, "Aspect Detection and Sentiment Classification Using Deep Neural Network for Indonesian Aspect-Based Sentiment Analysis," in *2018 International Conference on Asian Language Processing (IALP)*, 2018, pp. 62–67, doi: 10.1109/IALP.2018.8629181.

[24] A. Ligthart, C. Catal, and B. Tekinerdogan, "Systematic reviews in sentiment analysis: a tertiary study," *Artificial Intelligence Review*, Mar. 2021, doi: 10.1007/s10462-021-09973-3.

[25] R. P. Nawangsari, R. Kusumaningrum, and A. Wibowo, "Word2vec for Indonesian sentiment analysis towards hotel reviews: An evaluation study," in *Procedia Computer Science*, 2019, vol. 157, pp. 360–366, doi: 10.1016/j.procs.2019.08.178.

[26] M. Mhatre, D. Phondekar, P. Kadam, A. Chawathe, and K. Ghag, "Dimensionality reduction for sentiment analysis using preprocessing techniques," in *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, 2017, pp. 16–21, doi: 10.1109/ICCMC.2017.8282676.

[27] Y. A. Putra and M. L. Khodra, "Deep learning and distributional semantic model for Indonesian tweet categorization," in *2016 International Conference on Data and Software Engineering (ICoDSE)*, 2016, pp. 1–6, doi: 10.1109/ICODSE.2016.7936108.

[28] S. Twinandilla, S. Adhy, B. Surarso, and R. Kusumaningrum, "Multi-Document Summarization Using K-Means and Latent Dirichlet Allocation (LDA) – Significance Sentences," *Procedia Computer Science*, vol. 135, 2018, doi: 10.1016/j.procs.2018.08.220.

[29] M. al Omari, M. Al-Hajj, N. Hammami, and A. Sabra, "Sentiment Classifier: Logistic Regression for Arabic Services' Reviews in Lebanon," in *2019 International Conference on Computer and Information Sciences (ICCIS)*, 2019, pp. 1–5, doi: 10.1109/ICCISci.2019.8716394.

[30] H. Hasanli and S. Rustamov, "Sentiment Analysis of Azerbaijani twits Using Logistic Regression, Naive Bayes and SVM," in *2019 IEEE 13th International Conference on Application of Information and Communication Technologies (AICT)*, 2019, pp. 1–7, doi: 10.1109/AICT47866.2019.8981793.