Journal of Applied Engineering and Technological Science
Vol 7(2) 2026: 1352-1369

CROSS-DOMAIN FAKE REVIEWS IDENTIFICATION BASED ON DEEP LEARNING NEURAL NETWORK WITH ROLLING COLLABORATIVE TRAINING

Irham Aryandi Basir[1*], Yuliant Sibaroni[2]
School of Computing, Telkom University, Indonesia[1,2]
irhamaryandi@student.telkomuniversity.ac.id[1], yuliant@telkomuniversity.ac.id[2]

Received: 13 May 2025, Revised: 17 December 2025, Accepted: 10 January 2026
**Corresponding Author*

ABSTRACT
Identifying fake reviews in the current digital era has become an interesting study, especially in cross-domain context. This process is based on the limitations of the situation, where not all review domains have labels, and the labeling process takes a long time. In the context of machine learning, cross-domain learning involves learning from data and prediction processes, using data from different domains. Identifying fake reviews, several previous studies have conducted cross-domain, and the results of these studies indicate that there are still several issues in detecting fake reviews. The main problem with cross-domain methods is the difficulty of the model in understanding the differences in characteristics between domains, such as differences in language style, word structure in reviews, and the context present in the reviews. The main problem with the cross-domain method is the difficulty of the model in understanding the differences in characteristics between domains, such as differences in language style, word structure in reviews, and the context present in reviews. Based on these issues, this research adopts an approach to identify fake reviews using the Convolutional Neural Network and Bidirectional Long Short Term Memory models, utilizing the Multi Feature Rolling Collaborative Training (MRCT) algorithm with data from Yelp dan Amazon. The experimental results show that by conducting two scenarios, Scenario-1 provides better performance with an accuracy of 98.59%, while scenario-2 is only capable of providing an accuracy performance of 79.64%. Additionally, by using multi-features, the model experienced a 24.96% improvement in detecting fake reviews across domains. Based on these results, it can be seen that the use of multi-features and rolling collaborative training with the CNN-BiLSTM model works effectively in identifying fake reviews across domains.
Keywords : *Cross-Domain Classification, Deep Neural Network, Multi-Feature, Rolling Collaborative Training, Identification Fake Reviews.*

1. Introduction

As e-commerce grows, online transactions have become commonplace, making reviews an important source of information for consumers before making a decision (Manaskasemsak et al., 2021). Several studies indicate that over 80% of e-commerce users read reviews before deciding to make a transaction (Hajek et al., 2020; Mohawesh et al., 2021). This situation has led to the use of fake reviews, resulting in incorrect decision-making and impacting the brand's reputation in the future. In 2007, the process of identifying fake reviews was first introduced by Jindal and Liu, who categorised fake reviews into three groups: untruthful opinions, reviews that provide comments based solely on the brand's image, and non-reviews that are irrelevant to the product and are advertisements. (Alsubari et al., 2021; Gupta et al., 2024; Mewada & Dewang, 2023). To date, efforts to identify fake reviews have been developed using a variety of approaches. In a textual context (Budhi et al., 2021), feature behavioral (J. Liu et al., 2024), as well as the use of deep learning models (Deshai & Bhaskara Rao, 2023). However, these approaches are limited to use within the same domain. With the increasingly massive and widespread growth of review data, performance will decline if used in different domains due to the varying characteristics, structures, and styles of language in each domain.

Identifying reviews in cross-domain has been carried out by Wei (2020). Comprehensively utilising textual features by combining the Stimuli Organism Response (S-O-R) method, Linguistic Inquiry and Word Count (LIWC) and Word2Vec. Ren (2025) applies multi-level generics that extract text information at various levels (word-sentence-document). The study shows that the use of cross-domain in identifying fake reviews is limited to the utilisation of

textual features. This becomes less effective if the two domains have significant differences in sentence context. To address this research gap, a further approach is proposed by utilising an additional feature, namely behavioral features to identify of fake reviews.

Behavioral features are an approach that utilises non-textual features, such as the number of ratings, number of reviews, percentage of positive reviews, and others (J. Liu et al., 2024; Mohawesh et al., 2021). The use of behavioral features in identifying fake reviews shows good results, as in the study conducted by Zhang (2023) and Liu (2024). The combination of textual features and behavioral features resulted in an accuracy of over 80%. Although the use of both features showed good results, to strengthen the identification performance results, a dynamic Rolling Collaborative Training (RCT) algorithm was used so that the model obtained information iteratively (Wang et al., 2020). In addition, deep learning neural network approaches such as Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) were also applied in this study. The CNN architecture uses convolutional layers that are capable of capturing local features, while BiLSTM utilises sequential principles that assess data in two directions, enabling it to capture global patterns in text review data. Based on this, the focus of this research is to develop cross-domain identification of fake reviews using a multi-feature approach that utilises dynamic training processes using a hybrid CNN and BiLSTM architecture.

Therefore, this study aims to: (1) Investigate the effectiveness of combining textual features and behavioral features, (2) Develop cross-domain identification of fake reviews using the MRCT algorithm and CNN-BiLSTM deep learning model, (3) Evaluate the model using k-fold cross-validation by comparing the accuracy, precision, recall, and F1-score results with other existing methods. This study uses a textual and behavioral approach, which has not been applied in cross-domain identification research. In addition, the use of rolling collaborative training as an adaptive algorithm and the use of a hybrid CNN-BiLSTM model allows the model learning process to be continuously updated, enabling a more accurate identification process. Based on this, the contribution of this research focuses on the identification of fake reviews, especially across domains, which is expected to provide theoretical benefits in the process of identifying fake reviews and can be used practically on e-commerce platforms and other systems that utilise review features.

2. Literature Review

2.1. Fake Reviews Identification Method

The increasing use of fake reviews to boost revenue has become an issue, where fake reviews can lead consumers to make wrong decisions. Currently, there have been many studies that examine this issue using various methods, such as unsupervised manner (Li et al., 2019), machine learning (Budhi et al., 2021; W. Liu et al., 2019; Martens & Maalej, 2019), and even a deep learning approach (Barushka & Hajek, 2019; Mohawesh et al., 2024; Neisari et al., 2021). The research shows that with the various methods used, the performance achieved has been at a very good level, as seen in the research by Budhi (2021), which produced a performance of 89.7%, as well as research conducted by Mohawesh (2024) achieved a result of 93.15%. However, this result only applies to single domain usage. This will differ if applied to different domains, with conditions of higher data growth and more varied data diversity, requiring a more general and adaptive approach to the changes that occur. Therefore, this study proposes a cross-domain approach.

2.2. Cross-Domain Classification in Fake Reviews

Cross-domain classification is a prediction process that generally identifies a domain based on the learning process that has been conducted on another domain (Mohawesh et al., 2021; Wei et al., 2020). This technique has been widely applied in several cases, such as identifying fake news (W. Tang et al., 2023), sentiment analysis (Kong et al., 2023; H. Ren et al., 2022), as well as text spam detection (Dhaka & Mehrotra, 2019). Meanwhile, in the case of fake reviews, this study has been conducted by several researchers previously using various approaches. Hernández (2017), in his study on cross-domain identification of fake reviews combining LDA (Latent Dirichlet Allocation), LIWC (Linguistic Inquiry and Word Count), and WSM (Word-Space Model), he showed results of 52.95% obtained using the Naive Bayes model on DeRev, OpSpam,

Abortion, Bestfriend, and Death Penalty data.

Wei (2020) using SOR, which groups words into three categories: stimulate, organism, and response. The results of the experiment show that, using several machine learning models, an average accuracy of 60.93% was obtained for the Hotel, Restaurant, and Doctor data. The same domain was also studied by Ren (2025) utilising multi-level generic features yields the best results at 83%. Based on several studies, it shows that the use of cross-domain in identifying fake reviews has increased from year to year, but its use is limited to textual contexts. On the other hand, there are still behavioral features that need to be considered in identifying fake reviews, especially cross-domain. Therefore, this study uses a multi-feature approach that utilises textual and behavioral features for detecting fake reviews.

2.3. Multi-Feature Fake Reviews Identification

Identifying fake reviews using multiple features is nothing new; many studies have already used this method. This multi-feature approach generally involves various features in a dataset, such as textual features (semantic, part of speech, bag of words, or word embedding) and behavioral features (length of review, rating deviation, number of ratings, or ratio of reviews). Budhi (2021), using 133 unique features from a combination of text features and behaviour features, the accuracy obtained for the MLP model was 93.70%, LR (Logistic Regression) 93.18%, and SVM 93.54%. Duma (2023), focusing on the rating aspect to detect fake reviews, it provides 99.5% accuracy. This shows that the use of additional features with the right feature selection does not result in a decrease in accuracy and will provide better performance. Based on this, this study uses several features, namely Word Embedding, Sentiment Score, Length of review, Maximum number of reviews, Average length of review, Percentage of positive reviews, Rating deviation, and Extreme rating behaviour.

2.4. Deep Learning and Rolling Collaborative Training on Fake Reviews Detection

The application of machine learning in identifying fake reviews is currently widely used. Classic models such as Logistic Regression (Le et al., 2022), Naive Bayes (Singhal & Kashef, 2023), Support Vector Machine (Budhi et al., 2021), Decision-tree (Elmogy et al., 2021) or Random Forest (Fathima Beevi et al., 2023) has shown excellent results. However, these models are limited to the manual use of feature engineering. Based on this issue, an approach that is capable of automatically adapting to feature extraction is needed. The process of automatic feature extraction in the concept of machine learning has been applied as a further development known as deep learning.

Identifying fake reviews using deep learning is no longer a novel concept, with numerous studies having applied it using a variety of algorithms. Barushka & Hajek (2019) which uses the CNN model, successfully detected fake reviews with good performance. The use of convolutional layers and max pooling layers, combined with the application of n-gram and skip-gram word embedding, was able to extract review text locally so that the model did not lose the main context in detecting fake reviews. Another study by Chen (2025) showed similar results in detecting fake reviews using BiLSTM combined with a positional attention mechanism. The use of a bidirectional sequential principle was able to understand the context of sentences in fake reviews globally. Based on these two studies, this study utilises the advantages of local understanding in CNN and global understanding in BiLSTM to improve performance in detecting fake reviews across domains. However, CNN and BiLSTM algorithms generally run statically, which poses a challenge in the frequent occurrence of domain shifting. Therefore, a Rolling Collaborative Training (RCT) approach, which is more adaptive and dynamic, is used.

Rolling Collaborative Training is a method that utilises an iterative training process, enabling the model to perform dynamic updates. This method has been implemented by Wang. (2020)The results of the study show a 16.57% improvement in performance compared to typical deep learning models. This improvement is due to the RCT method being able to reduce the impact of data changes on the classification process by adjusting model parameters. Based on these results, this study uses a combination of CNN-BiLSTM and the RCT method to detect fake reviews across domains.

3. Research Methods

This study utilised a cross-domain approach by employing a deep learning model (CNN-BiLSTM) combined with the rolling collaborative training method. The utilisation of pattern recognition by CNN and two-way global context in BiLSTM plays a role in improving the model's ability to recognise differences in review characteristics between domains. In addition, using the iterative and adaptive MRCT principle allows for a more dynamic model learning process and reduces domain shifting. Therefore, this study uses these approaches to detect fake reviews across domains. In general, the design can be seen in Fig. 1, which starts with data collection, identifying text and non-text features, followed by preprocessing to tidy up the data, then feature extraction, model training, and finally review identification by dividing reviews into two classes (fake or real).

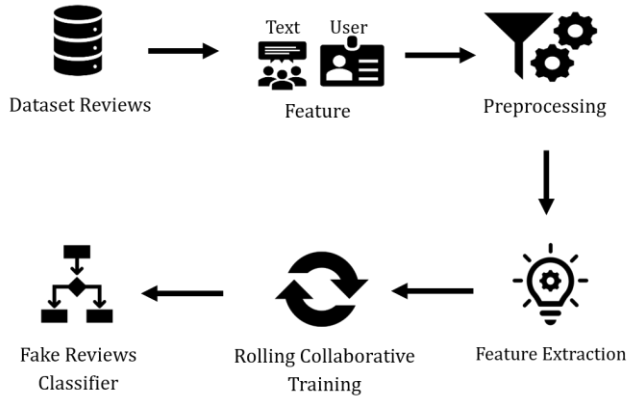

Fig. 1. Design Process

3.1 Preprocessing

This process involves data cleaning and tidying up the data in several stages. The preprocessing stage carries out several processes such as, cleaning text, apply lower case for text reviews, Remove Punctuation, Remove StopWords, Lemmatizing and Tokenizer. The preprocessing steps are explained as follows.

1. Perform the clean text process and tidy up the text by expanding the words. In addition, it also deletes link URLs, removes numbers in text both integer and decimal, reduces repetitive punctuation symbols into one such as "*good*!!" to "*good!*", removes multiple spaces in each sentence, as well as removes new lines in sentences and makes them single (Shinde et al., 2023).
2. The lower casing process is carried out on each letter, so as to create a uniformity of shape in the review text. This process is essential in text classification, as it can reduce execution time and improve model efficiency (Qayyum et al., 2023).
3. Remove punctuation, this process aims to clean up irrelevant punctuation symbols that often cause noise the teks review (Budhi et al., 2021; Qayyum et al., 2023).
4. Remove stop words, aims to remove any words that do not provide much information or words that are supportive words for other words, such as "the", "is", "that", "what" and others. This process is necessary because these words are the most frequently occurring and very common in review texts (Budhi et al., 2021; Shinde et al., 2023).
5. Lemmatise, this process converts every word with the same meaning into its basic form or root, such as "runs", "running", "ran" into "run". This is necessary to increase the efficiency of the learning process in machine learning (Raheem & Chong, 2024).
6. Tokenizer, the last preprocessing is to tokenize each word in the review text by separating each word into a token, so that the model can be more efficient in learning text reviews word by word (Qayyum et al., 2023; Raheem & Chong, 2024).

After all the preprocessing on both domains is carried out, clean review text data in the form of tokens is obtained to be used in the feature extraction process.

3.2 Feature Extraction

This process draws on textual features and behavioral features. The two features are used to classify according to the process design that has been made in Fig. 1. Feature extraction aims

to extract text data and behavioral data into a numeric representation so that machine learning models can process both features. This feature extraction process produces five behavioral features, such as maximum number of reviews, average length of reviews, rating deviation, percentage of positive reviews, and extreme rating behaviour. Textual features consist of review length, sentiment score, and word embedding.

3.2.1. Behavioral Feature

The behavioral feature is a representation of the statistical significance of a user based on the reviews he has made by calculating the length of reviews, the average length of reviews, the Bottom-ranked reviews ratio, the Top-ranked reviews ratio, and the Extreme rating behavioral.

a. Length of Review

Length of review is the number of words contained in the review text, This number of words is an important aspect in the identification of fake reviews. The number of words in a review can be a reference to whether the review is classified as fake or real, in general the number of words in fake reviews tends to be less than the number of words in real reviews (Mewada & Dewang, 2023; Mohawesh et al., 2021).

b. Maximum Number of Reviews.

Based on several previous studies, it has been shown that each individual or e-commerce user will make a review no more than one review per day, while 75% of groups or individuals who create fake reviews will do it at least five times a day (Mohawesh et al., 2021; Tang et al., 2020).

$$Max_a = \frac{MaxRev(a)}{Max(MaxRev)} \quad (1)$$

The results of the study are to find out whether the review is a fake review or not, it can be seen from the number of reviews made by individuals or groups, using the formula (1). Maximum number of reviews Type equation here. (Max_a) defined as the ratio of review uploads made by a person in a certain time bracket by calculating the maximum number of reviews per ID ($MaxRev$) compared to the maximum reviews of the whole ID ($Max(MaxRev)$).

c. Average length of reviews

The average length of reviews ($AvgR_a$) is the total number of words in a review (r) From each reviewer, where several studies show that 85% of spammers will create fake reviews with an average sentence length of less than 135 words, while 90% of genuine reviews are written longer and more detailed with an average review length of 200 words (He et al., 2022; Mewada & Dewang, 2023; Mohawesh et al., 2021).

$$AvgR_a = \begin{cases} 1, & len(r(a)) < X = 150 \\ 0, & otherwise \end{cases} \quad (2)$$

A review can be categorized as a fake or spam review based on the length of the review, this is in accordance with some previous studies that stated that fake reviews are more concise and explain in detail about the product being reviewed.

d. Rating Deviation

Rating deviation calculation (RDa) aim to find out the deviation value between rating reviews made by an individual on a product to the average rating ($\bar{r_t}$) value of the product (Mewada & Dewang, 2023; Mohawesh et al., 2021; Xiang et al., 2023).

$$RD_a = \frac{|\bar{r_t}(a) - r_t(a)|}{4} \quad (3)$$

This feature is based on several studies where every spammer has a habit of giving high ratings to the products they review; so that by calculating the rating deviation, it is expected to identify fake reviews.

e. Percentage of Positive Review

Based on previous studies where each spammer will give a rating (r_t) for each product they review, so that by calculating the percentage of positive reviews (PPR_a) then it will make it easier to detect fake reviews (Mewada & Dewang, 2023; Mohawesh et al., 2021).

$$PPR_a = \frac{\sum_{k=1}^{|R(a)|} |\{r_t(a) \in [4,\ 5]\}}{|R(a)|} \quad (4)$$

This calculation was carried out only by paying attention to the rating of reviews that were rated four and five stars, this was based on the results of analysis on Yelp data found from 85% of reviewers who made fake reviews 80% of the reviews gave ratings between four and five (Mewada & Dewang, 2023).

f. Extreme Rating Behavioral

The last feature extraction is extreme rating behavioral (ER_t), this extraction will be carried out for each review by grouping the reviews into two categories, namely those who get an extreme rating score of one and five, while the rest are categorized as non-extreme with get a rating score with a score range of two to four (Mewada & Dewang, 2023; Mohawesh et al., 2021).

$$ER_t(a) = \begin{cases} 1, & r_{t(a)\in\{1,\ 5\}} \\ 0, & r_{t(a)\in\{2,\ 3,\ 4\}} \end{cases} \quad (5)$$

Process Extracting is based on characteristics of fake reviewers that often give ratings (r_t) quite extreme in each review (Mohawesh et al., 2021).

3.2.2. Textual Feature

Textual feature is the result of extracting text reviews in each domain used in this study. To obtain information from these texts, feature extraction is carried out by looking for the length of review, sentiment feature, and word embedding.

a. Sentiment Feature

Sentiment feature (S_r) in the identification of fake reviews is one of several factors that need to be considered, this can be seen in Saumya research (2018), Wang (2020), and Birim (2022) which uses a sentimental approach in detecting spam opinions, especially in a review. The process of calculating the sentiment classification in a review can be seen in the following equation (6).

$$S_r(a) = \begin{cases} 1, & r_t(a) \in \{4,\ 5\}\ and\ (C_s \geq 0.8) \\ -1, & r_t(a) \in \{1,\ 2\}\ and\ (C_s \leq -0.8) \\ 0, & otherwise \end{cases} \quad (6)$$

The calculation of sentiment score is carried out using the VADER tool. The tool provides a polarity sentiment Compound Score value (C_s) between -1 to 1. A value of -1 indicates extreme negative and 1 indicates extreme positive. Furthermore, this compound score will be categorized according to equation (6). Sentiment features are part of semantic features that measure or detect the meaning of a sentence This sentiment feature assesses sentences into categories, namely positive and negative (Birim et al., 2022; Mewada & Dewang, 2023; Mohawesh et al., 2021).

b. Word Embedding

The process of word embedding is a neural network process that converts words or sentences into numerical vectors (Barushka & Hajek, 2019; Hajek et al., 2020; Mohawesh et al., 2021). Word embedding is different from the general method of representing words into vectors, word embedding can represent an arrangement of words into a vector even though the set of words has a limited number of words (Mohawesh et al., 2021). Based on previous research each review length has an average of 200 words, utilizing optimal word embedding so that it can reduce computing time. In addition, word embedding is able to capture semantics and compatibility between words so that the use of word embedding is more advanced in representing features (Y. Liu et al., 2022). This research will represent reviews into a numerical array so that it can be processed to predict fake reviews by utilizing the FastText method which gives better results from wor2vec (Mohawesh et al., 2021).

The FastText model is an advanced level of the word2vec model that utilizes information from subwords in a text that can be used as word embedding (Asudani et al., 2023; Mohawesh et al., 2023). Another advantage is that FastText is lighter in memory usage compared to BERT, which is a more complex and heavier transformer. In addition, the use of the n-gram principle in FastText provides the advantage of being able to generate vectors for new words, unlike GloVe, which is based on global word co-occurrences. FastText was developed by Facebook as a technique to generate word vectors based on the morphology of the word represented in the form of n-grams (Raheem & Chong, 2024), therefore, FastText pretrained with 300 dimensions was used in this study.

3.3 CNN-BiLSTM Model

This process begins by converting text into numerical representations using Quad Fastext embedding, which produces 400-dimensional vectors. These vectors are then read by a convolutional layer to find numerical patterns for each feature. To perform all these processes, several parameters are applied, such as using 128 filters with a kernel size of 3 and an activation function by ReLu. After the convolutional layer maps these numerical patterns, the next step is to reduce the dimensions and identify the features that influence the feature map through the max-pooling layer. After the max-pooling process, the numerical vectors have changed and undergone dimensional changes that represent local patterns in the text. After obtaining the output from the max-pooling process, the next process is carried out on the BiLSTM block.

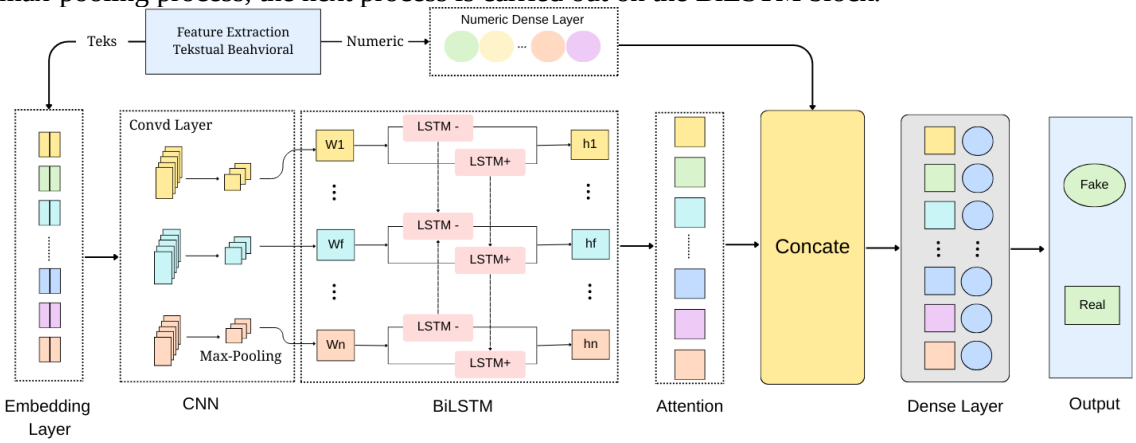

Fig. 2. Proposed Model Architecture

Convolutional Neural Network (CNN) has become a crucial model in machine learning programming, where CNN is very popular in the process of computer vision (Alzubaidi et al., 2021; Mohawesh et al., 2023). However, with the development of CNN studies, it is no longer used solely for computer vision. Deshei, in his study, explains that the utilization of CNN in text-based computation yields very good results in classification (Deshai & Bhaskara Rao, 2023). In addition, by using CNN in the neural network space based on text, semantic information with high-dimensional features can be detected effectively (Hajek et al., 2020; Mohawesh et al., 2021). The result is caused by the CNN structure, which consists of several layers of neurons resembling the structure of the human and animal brain (Alzubaidi et al., 2021). Basically, CNN is very similar to the multi-layer perceptron (MLP) model, which consists of 3 main layers: the convolutional layer, pooling layer, and fully connected layer, as shown in Fig.3.

Table 1 - Model Configuration

Component	Layer/Parameter	Configuration
Embedding Layer	FastText General	Embed_dim=100, frozen
	FastText Domain Training	Embed_dim=100, trainable
	FastText Domain Testing	Embed_dim=100, trainable
	FastText Train + Test	Embed_dim=100, trainable
CNN Block	Conv1D	128 filters, kernel=3
	Max-Pooling	Pool_size=2
	Dropout	Rate=0.3
BiLSTM Layer	Bidirectional LSTM	128 × 2 directions
Attention & Pooling	Attention Mechanism	Self-attention
	Dense (Text Branch)	64 units, ReLU
Numeric Branch	Dense Layer 1	256 units, ReLU
	Dense Layer 2	128 units, ReLU
	Dense Layer 3	64 units, ReLU
Classifier	Dense Layer 1	256 units, ReLU
	Dense Layer 2	128 units, ReLU
	Dense Layer 3	64 units, ReLU
	Dense Layer 4	32 units, ReLU
	Output Layer	2 units, Softmax

Training Configuration	Loss Function	Focal Loss or CCE
	Optimizer	Adam
	Regularization	Dropout (0.2-0.3)

This process begins by converting text into numerical representations using Quad FastText embedding, which produces 400-dimensional vectors. These vectors are then read by a convolutional layer to find numerical patterns for each feature. To perform all these processes, several parameters are applied, such as using 128 filters with a kernel size of 3 and an activation function by ReLu. After the convolutional layer maps these numerical patterns, the next step is to reduce the dimensions and identify the features that influence the feature map through the max-pooling layer. After the max-pooling process, the numerical vectors have changed and undergone dimensional changes that represent local patterns in the text. After obtaining the output from the max-pooling process, the next process is carried out on the BiLSTM block.

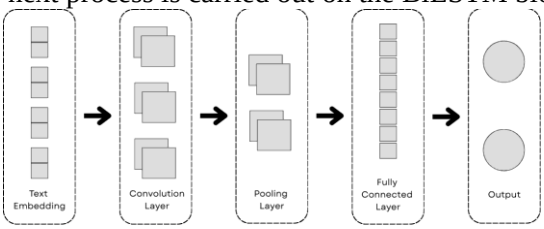

Fig. 3. CNN Architecture

The Bidirectional Long Short Term Memory (Bi-LSTM) model is designed to overcome the limitations of LSTM in making predictions based solely on previous data (Shinde et al., 2023). LSTM itself is a fine-tuned model of the Recurrent Neural Network (RNN) model that requires a prediction process based on data advantages by utilizing convolutional layers to capture local patterns, especially in data with relatively high dimensions often found in text data. On the other hand, Bi-LSTM is capable of understanding data patterns globally by analyzing data in both directions, thus with this capability, Bi-LSTM can comprehensively understand the context of the data.

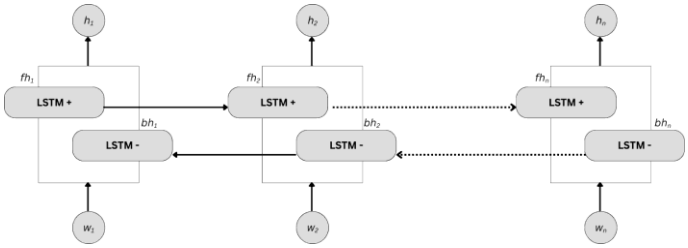

Fig. 4. BiLSTM Architecture

Based on the architecture used in Fig. 4, after the vector goes through the max-pooling process in the next CNN layer, it is processed by a BiLSTM consisting of one layer. The parameters used in the BiLSTM layer consist of 128 units for both directions. With these parameters, the BiLSTM can maintain the sequential structure that produces a 256-dimensional output, a combination of the two LSTM directions. Next, after the BiLSTM process is complete, the next process is to perform an attention mechanism that reduces the output from the 256-unit BiLSTM process to 64 units with the ReLU activation function. By combining CNN-BiLSTM, the model can capture patterns locally and globally in a sequential manner.

3.4 Rolling Collaborative Training

The process of rolling collaborative training can be seen in Figure 5. This method implements an iterative and dynamic retraining process, makes the model more adaptive with each iteration as it updates information. This approach offers an additional learning model to accurately predict the following data, as supported by the research conducted by Wang (2020).

Algorithm 1 Rolling Collaborative Domain-Adaptive Training

Input:

DA ← Domain A dataset

DB ← Domain B dataset
DB_test ← 20% portion of Domain B
kA ← 80% portion of Domain A used for similarity search
kB_train ← 10% portion of Domain B (based on |DA_sim|)
max_iter ← 15

Output:

Adapted CNN-BiLSTM model M
Evaluation metrics on DB_test

Step 1 — Similarity-Based Selection

DA_80 ← Take Random Subset (DA, kA)
sim_scores ← Compute Similarity (DA_70, DB)
DA_sim ← Select Most Similar (DA_70)

Step 2 — Splitting Domain B

n ← |DA_sim|
DB_train_small ← First(n * 0.10) samples of DB
DB_pool ← Next (samples of DB - DB_train_small)

Step 3 — Construct Training Data

BaseTrain ← DA_sim ∪ DB_train_small
BalancedTrain ← Borderline-SMOTE (BaseTrain.numeric + BaseTrain.behaviour)
Extract textual (T1), behavioral (T2) features

Step 4 — Initialize Model

M ← CNN-BiLSTM-Quad-FastText()
Train(M, BalancedTrain)

Step 5 — Rolling Collaborative Training (Iterative Pooling)

iter ← 1
while iter ≤ max_iter do
    Predictions ← M.predict(DB_pool)
    Selected_p ← Choose Fake Samples (Predictions)
    Selected_n ← Choose Real Samples (Predictions)
    DB_pool ← Remove (DB_pool, Selected_p ∪ Selected_n)
    NewBatch ← Selected_p ∪ Selected_n
    TrainBatch ← BalancedTrain ∪ NewBatch
    Train(M, TrainBatch)
    iter ← iter + 1
end while

Step 6 — Final Evaluation

FinalMetrics ← Evaluate(M, DB_test)
return M, FinalMetrics

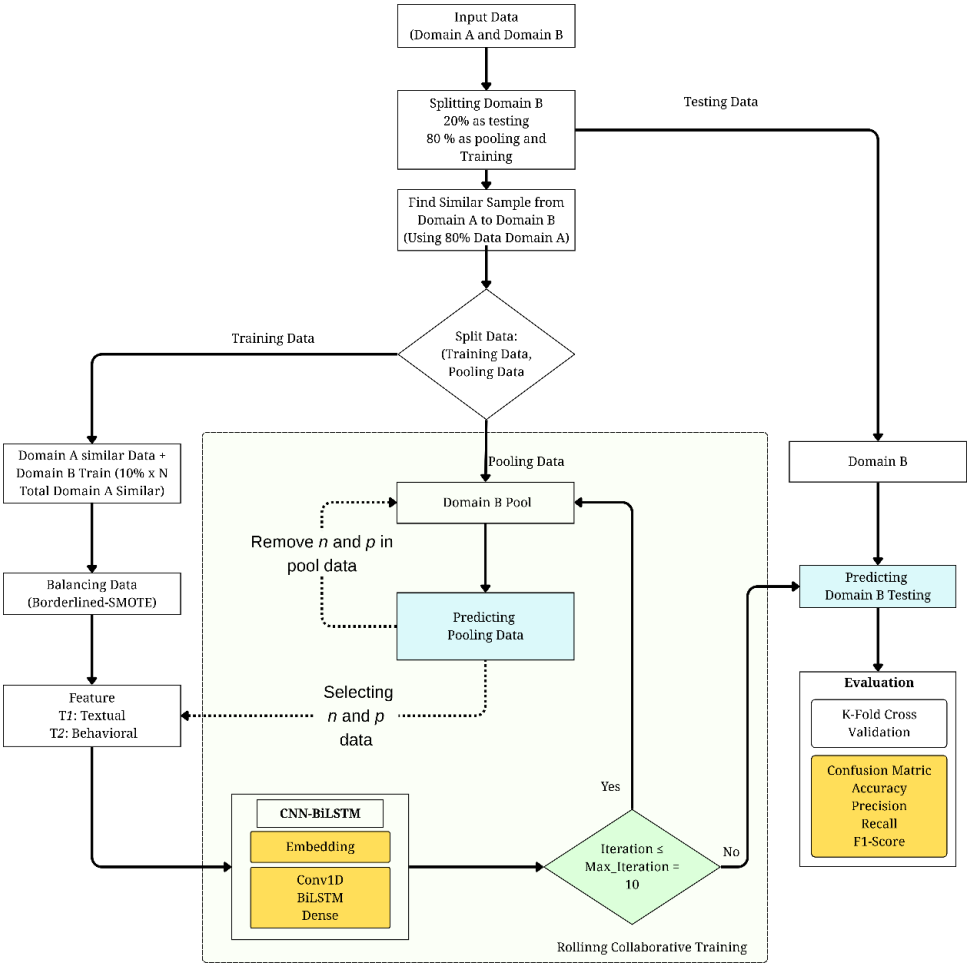

Fig. 5. Rolling Collaborative Training

1. Input data domain A (DA) and Domain B (DB).
2. Divide Domain B into two parts: 20% for testing (DB_test) and 80% for pooling (DB_Pool) and train small (DB_train_small).
3. After defining the domain for training and the domain for testing, the next step is to find the similarity (DA_sim) between DA and DB using 80% DA.
4. Separation of L1 textual and L2 Behavioral features.
5. Extraction of data features into L1 textual data and L2 Behavioral data.
6. Divide the domain B data into three data groups, dimana jumlah datanya disesuaikan dengan total (n) data similar (DA_sim) :
   - (n x 10%) training (DB_train_small)
   - data pooling (DB_pool)
7. Furthermore, the data from the Extraction results will be trained using the CN BiLSTM model, which is used to predict domain B.
8. Conducting a training process by combining 10% training (DB_train_small) data from domain B and similar data from domain A.
9. Predicting data pooling (DB_pool).
10. Selecting n+p data on predicted data pooling.
11. Add the selected n and p data to the training data.
12. Delete n and p data on data pooling.
13. Re-training with updated data.
14. The iteration process from steps 8 to 12 continues until Max Iteration (max_iter) is reached.
15. Evaluate the model using testing data (DB_test).

After all the processes are completed, the last step is to evaluate by calculating the accuracy, precision, recall, and f1-score values based on the confusion matrix value, comparing the suitability of real data and prediction results to determine the accuracy of the method used.

3.5 Performance Measurement

3.5.1 *K-Fold Cross Validation*

As an effort to measure whether the model used is a good model. In this study, k-fold cross-validation is used considering that in the dataset used, there is a dataset that is imbalanced so it is necessary to carry out an under-sampling process. In practice, the condition of data reviews is imbalanced where the number of fake data and regular reviews actually has a different number, where this condition can be seen from the entire number of reviews on a domain or data set in general only has a percentage of 10% to 15% of the total data (Martens & Maalej, 2019).

K-Fold Cross Validation (CV) is a machine learning method that is generally used for validation or evaluation in training, where in the process CV will divide the entire dataset into two parts, namely training data and testing data (Lainder & Wolfinger, 2022). In addition, CV will also divide the data into several partitions according to the number of *k* that has been determined, where from the entire data one partition will be taken to become data testing and the other k-1 partition will be used as training by using k-fold cross-validation (Lainder & Wolfinger, 2022; Pal & Patel, 2020).

3.5.2 Confusion Matrix

Table 2 - Confusion Matrix

		Real	
		Non-Fake	Fake
Prediction	Non-Fake	TN	FN
	Fake	FP	TP

The process of evaluating the model using k-fold cross-validation pays attention to four metrics that must be used, namely accuracy, precision, recall, and f1-score. To obtain the values of these four metrics, a confusion matrix is used which is a matrix of prediction results and actual data states

$$Accuracy = \frac{TN+TP}{TN+TP+FN+FP} \quad (7)$$

$$Precission\ (Fake) = \frac{TP}{TP+FP} \quad (8)$$

$$Recall\ (Fake) = \frac{TP}{TP+FN} \quad (9)$$

$$F1 = 2 \times \frac{Precission \times Recall}{Precission + Recall} \quad (10)$$

The confusion matrix in Table 2 is a paired table consisting of the amount of prediction data and the amount of actual data, which are grouped into four categories, namely True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP). In the case of fake reviews, identification TN shows that the data predicted to be genuine reviews and in fact genuine reviews. Furthermore, TP explained that the data that has been successfully predicted and the actual situation is fake reviews, while the data classified as FN is review data whose prediction results are classified as genuine reviews but are actually fake reviews, and the last category, namely FP, shows data whose prediction results are classified as fake reviews but the actual situation is a real review. Of these four categories, it is used to find the accuracy, precision, recall, and F1-score scores. Accuracy shows the comparison of the overall predicted data to the actual situation compared to the overall data. Precision is the amount of data that is correctly predicted of the overall prediction. A recall indicates the amount of data in the dataset that was correctly predicted. F1-Score is harmonic average of precision and recall that has a value range of 0 to 1, which aims to assess the level of accuracy of unbalanced data.

4. Results and Discussions

4.1 Dataset Description

The dataset used in this study is a Yelp and Amazon. Yelp data consists of three subgroups, namely YelpChi, YelpNYC and YelpZip (Rayana & Akoglu, 2015), is a dataset that contains

reviews of Restaurant and hotel, while Amazon dataset (Hussain et al., 2020), is data containing reviews from Amazon.com that have been labeled using crowdsourcing using Amazon Mechanical Turk (AMT) which consists of five categories, namely Clothing Shoes and Jewelry, Home and Kitchen, Sports and Outdoors, Toys and Games, Cell Phones and Accessories. The total amount of data from all datasets are 110.000 data, in detail can be seen in Table 3.

Table 3 - Dataset Statistic

Dataset	Category	Fake	Real	Total
Amazon	Clothing Shoes and Jewelry	7.857	2.143	10.000
	Home and Kitchen	7.949	2.051	10.000
	Sports and Outdoors	8.310	1.690	10.000
	Toys and Games	7.073	2.927	10.000
	Cell Phones and Accessories	8.362	1.638	10.000
Yelp	YelpChi	2.844	17.156	20.000
	YelpNYC	2.105	17.895	20.000
	YelpZip	2.734	17.266	20.000
Total		19.479	82.529	110.000

Both datasets were selected using random sampling, with data from 2010 to the present year. A total of 10,000 data points were taken from Amazon, while 20,000 data points were selected from Yelp for each category. The use of these two datasets in this study was due to the wealth of more complete data that enabled cross-domain identification of fake reviews. The time of review creation, user ID, and rating are very important in the application of multi-features that require non-textual and textual data. In addition, the condition of the dataset originating from very different domains can justify the proposed method in this study, which is whether the two datasets can identify fake reviews well. Based on the results of the data description, several conditions in the two datasets were found that need to be considered, such as the distribution of labels and the distribution of review lengths.

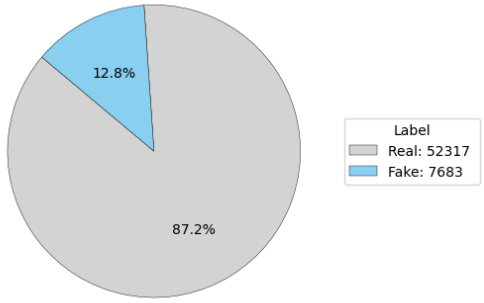

Fig. 6. Yelp Label Percentage

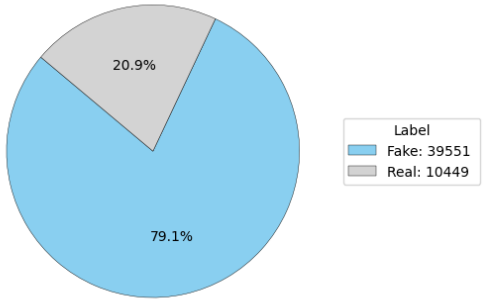

Fig. 7. Amazon Label Percentage

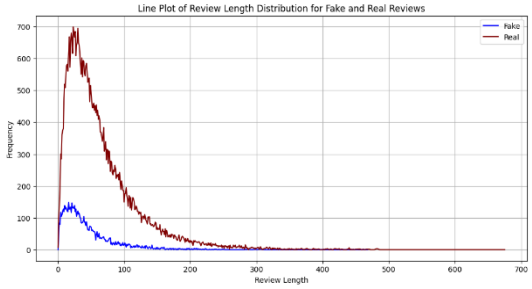

Fig 8. Yelp Length Reviews Distribution

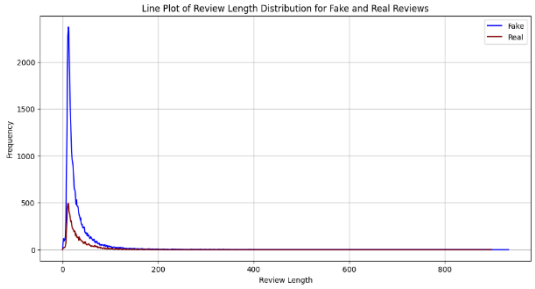

Fig. 9. Amazon Length Reviews Distribution

The label distribution results as shown in Fig. 6 and Fig. 7 indicate differences in distribution between the two datasets. The distribution in the Yelp data shown in Fig. 6 indicates that 87.2% of reviews are labelled as real and 12.8% as fake. This 7:1 ratio indicates that for every 100 reviews in the Yelp data, there are 13 fake reviews. This contrasts with the Amazon data, which tends to have more fake reviews than real reviews, as shown in Fig. 7, where the ratio

reaches 79.1%. Based on the length of reviews, both datasets have similar patterns, as seen in Fig. 8 and Fig. 9, where the distribution of review lengths is centred in the 0-200 word range.

4.2 Experiments Result

This section explains the results of research that has been carried out on three cross-domain scenarios, namely classifying the identification of fake reviews by combining textual and behavioral features, the next experiment is to identify based on textual features, and the last experiment is comparing with another model. The three experiments identified fake reviews cross-domain by applying two scenarios of each combination of domains such as Yelp as training data and Amazon as testing data, the same applies in the opposite direction, as shown in Table 4 below.

Table 4 - Experiment Scenario

Scenario	Training Domain	Testing Domain
Scenario 1	Yelp	Amazon
Scenario 2	Amazon	Yelp

The combination of scenarios was conducted to examine whether there is an improvement in performance in fake review classification using the proposed method. In addition, reciprocal testing was performed to determine which dataset is more representative in the cross-domain process.

4.2.1 Multi-Feature

The results of the cross-domain multi-feature fake review identification experiment shown in Table 4 demonstrate excellent performance in scenario-1 (Yelp → Amazon), as indicated by a five-fold average accuracy of 98.59%. These results indicate that the proposed method, using Yelp data for training, is capable of handling cross-domain problems, namely differences in characteristics between the two domains. In addition, the model is also capable of correctly predicting fake reviews from the entire data predicted on Amazon data by 99.86%, as seen in the precision results. This explains that if there are 10,000 data predicted as fake reviews, the model only experiences 14 data prediction errors. Similarly, the recall score only experienced a prediction error of 1.64% of the total data labelled as fake. Therefore, from the precision and recall results, the harmonic average accuracy provided by the model is 99.10%, indicating that with unbalanced data, the model proposed in this study is capable of performing well in detecting fake reviews, where the model does not tend to predict data on certain labels.

Table 5 - Performance Metrics Multi-Feature

Scenario-1 (Yelp → Amazon)						
Matric	**Fold-1**	**Fold-2**	**Fold-3**	**Fold-4**	**Fold-5**	**Mean**
Accuracy	0.9855	0.9840	0.9870	0.9915	0.9815	0.9859
Precision	0.9994	0.9987	0.9994	0.9981	0.9974	0.9986
Recall	0.9823	0.9810	0.9842	0.9912	0.9791	0.9836
F1-Score	0.9908	0.9898	0.9917	0.9946	0.9882	0.9910
Scenario-2 (Amazon → Yelp)						
Matric	**Fold-1**	**Fold-2**	**Fold-3**	**Fold-4**	**Fold-5**	**Mean**
Accuracy	0.7900	0.7996	0.7983	0.8079	0.7863	0.7964
Precision	0.2161	0.2194	0.2294	0.2556	0.2012	0.2243
Recall	0.2443	0.2215	0.2443	0.2597	0.2240	0.2388
F1_score	0.2294	0.2204	0.2366	0.2576	0.2120	0.2312

Unlike scenario-1, scenario-2 shows the opposite result, as seen in Table 5, where the accuracy value using Amazon data as training data only obtained 79.64% of the average of the five folds tested. This is clarified by the precision and recall values of 22.43% and 23.88%, respectively. Based on these two measurements, it can be seen that the proposed model is only capable of correctly predicting fake review data, which is less than 2,400 data out of 10,000 data, both from the overall data predicted to be fake reviews and from the overall data labelled as fake. The reason for these low measurement results is that the model failed to recognise the data properly in the Yelp domain, which has different data and label characteristics, as shown in Fig.

6 and Fig. 7, where the two data sets have very different label distributions. Therefore, the results of scenario 2 show that the use of Amazon data, which has a higher proportion of fake labels compared to real labels, is not yet capable of identifying fake reviews across domains on the target Yelp domain.

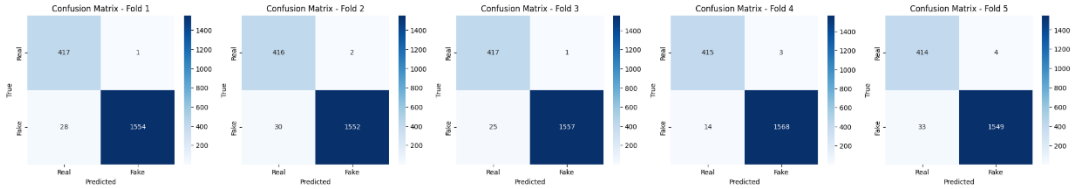

Fig. 10. Confusion Matrix Multi-feature Scenario-1 (Yelp – Amazon)

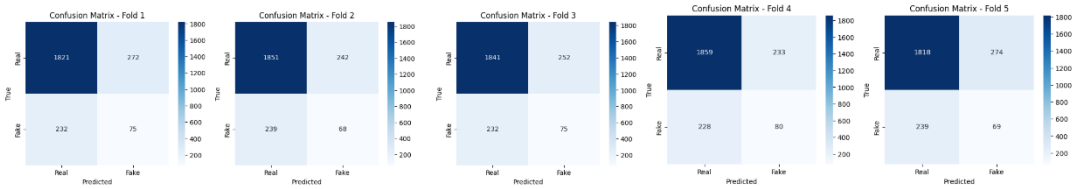

Fig. 11. Confusion Matrix Multi-feature Scenario-2 (Amazon – Yelp)

These differences can also be seen in the confusion matrix results in Fig. 10 and 11. In Scenario-1, with a total of 10,000 test data points and five-fold evaluation, the model produced 130 false negatives from 7,910 fake review data points. Similarly, there were 11 false positives from 7,791 data points predicted to be fake reviews. Meanwhile, in scenario 2, there were 1,273 false positives from 1,640 predicted fake data and 1,170 false negatives from 1,537 fake data. These measurements show that in scenario-2, from all fake reviews in the Yelp data, the model was only able to correctly guess 69 data. Based on these measurement results, in terms of accuracy, precision, recall, F1-score, and confusion matrix, scenario-1, which utilised Yelp data as training data, was better at identifying fake reviews across domains than scenario-2, which utilised Amazon data. The results of these two scenarios also show that the proposed method only worked well in scenario-1 and experienced a decline in performance when applied to scenario-2.

4.2.2 Textual Feature

This experiment was conducted to observe the identification process in the proposed method, which only uses textual features. The results of the experiment can be seen in Table 6. Based on the measurement matrix results in the table, the average accuracy value in scenario-1 is 73.63%, while in scenario-2 it is 66.31%. Although the accuracy performance results for these two scenarios are not significantly different, there are considerable differences overall, as seen in the precision, recall, and F1-score measurement results. Based on the results of the five-fold evaluation, the average f1-score value in scenario-1 was 82.28%, obtained from precision and recall values of 87.8% and 77.51%. In contrast to scenario-1, the measurement results for scenario 2 only achieved an f1-score of 26.45% with precision and recall values of 18.36% and 47.30%, respectively. The results of both scenarios show that cross-domain identification using textual features yields good results when applied to scenario-1, where the scenario was able to accurately predict 7,363 data points out of 10,000. Meanwhile, in scenario-2, although the overall accuracy obtained results that were not much different, the harmonic still could not predict fake reviews well, where the model tended to predict data labelled as real.

Table 6 - Performance Matric Textual Feature

Scenario-1 (Yelp → Amazon)						
Matric	**Fold-1**	**Fold-2**	**Fold-3**	**Fold-4**	**Fold-5**	**Mean**
Accuracy	0.7305	0.7425	0.7353	0.7345	0.7385	0.7363
Precision	0.8744	0.8813	0.8691	0.8703	0.8891	0.8768
Recall	0.7699	0.7794	0.7807	0.7807	0.7649	0.7751
F1-Score	0.8188	0.8272	0.8225	0.8231	0.8223	0.8228
Scenario-2 (Amazon → Yelp)						
Matric	**Fold-1**	**Fold-2**	**Fold-3**	**Fold-4**	**Fold-5**	**Mean**
Accuracy	0.6700	0.6679	0.6613	0.6512	0.6650	0.6631
Precision	0.1919	0.1907	0.1748	0.1706	0.1900	0.1836

Recall	0.4919	0.4919	0.4430	0.4448	0.4935	0.4730
F1_score	0.2761	0.2748	0.2507	0.2466	0.2744	0.2645

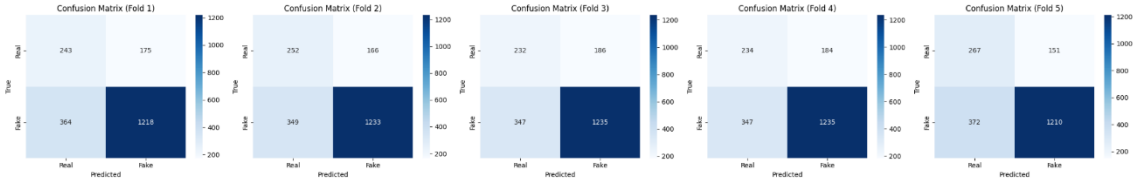

Fig. 12. Confusion Matrix Textual Feature Scenario-1 (Yelp – Amazon)

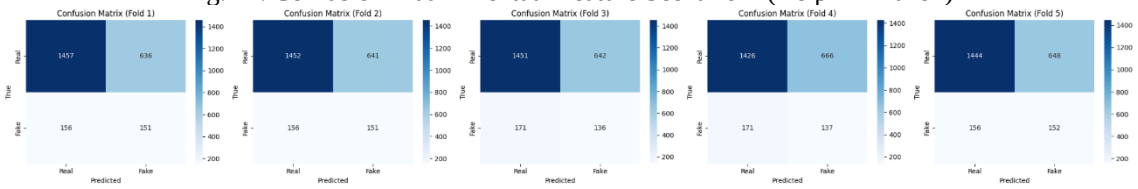

Fig. 13. Confusion Matrix Textual Feature Scenario-2 (Amazon – Yelp)

The confusion matrix results in Fig. 12 and Fig. 13 from both scenarios show that there were 1,779 false negative results in scenario-1 and 810 false negative results in scenario-2. Meanwhile, there are 862 false positives in scenario-1 and 3,233 false positives in scenario-2. These two measurements prove that the proposed method in scenario-1 made errors in predicting fake reviews as real reviews in 1,779 data points and predicting real reviews as fake reviews in 862 data points from 10,000 testing data points. Scenario 2 made errors in predicting fake reviews as real reviews in 810 cases and predicted 3,233 real reviews as fake reviews from 12,000 review data. Based on these conditions, from all the data labelled as fake in scenario 1, the method used only correctly predicted 6,131 fake reviews from 7,910 fake review data. Meanwhile, in scenario-2, it was only able to correctly predict 727 data points out of 1,537 fake review data points. The high failure rate of 52.7% in scenario-1 and 22.49% in scenario-2 shows that utilising textual features is not yet able to efficiently identify fake reviews across domains using the proposed method.

4.2.3 Model Comparison

The model comparison conducted in this study compares the results of experiments carried out using several methods from previous research, such as ST-MFLC (Cao et al., 2022), LSTM-RoBERTa (Mohawesh et al., 2024), and EUPHORIA (Andresini et al., 2022). The three methods used as comparison models will identify fake reviews across domains using Yelp data as training data and Amazon data as testing data, with a data ratio of 80% for the training process and 20% for testing data.

Table 7 – Comparison Model

Model	Feature-Based	Accuracy	Precision	Recall	F1-Score
ST-MFLC	Textual	0.2283	0.4582	0.4890	0.2702
LSTM-RoBERTa	Textual & Behavioral	0.3578	0.7696	0.2685	0.3981
EUPHORIA	Textual & Behavioral	0.4941	0.7925	0.4970	0.6109
Proposed Method	Textual	0.7363	0.8768	0.7751	0.8228
Proposed Method	Textual & Behavioral	0.9859	0.9986	0.9836	0.9910

The comparison results shown in Table 7 indicate that of the five models, the proposed method obtained better accuracy results than the other four models, where the method that applies rolling-collaborative training combined with the CNN-BiLSTM model provides an accuracy of 98.59%, utilising multi-features that use textual features and behavioural features. Meanwhile, models that apply similar methods, such as LSMT-RoBERTa and EUPHORIA, only achieve an accuracy of 49.41% and 35.78%. Another comparison is using textual features, where the ST-MFLC model achieves an accuracy of 22.83%, while when applied in the proposed method used in this study, the accuracy can reach 63.63%. Based on these results, it can be seen that for Yelp and Amazon data, the proposed method is better at detecting fake reviews, both textually and multi-feature, compared to several previous studies.

5. Conclusion

The comparison results shown in Table 6 indicate that of the five models, the proposed method obtained better accuracy results than the other four models, where the method that applies rolling-collaborative training combined with the CNN-BiLSTM model provides an accuracy of 98.59%, utilising multi-features that use textual features and behavioural features. Meanwhile, models that apply similar methods, such as LSMT-RoBERTa and EUPHORIA, only achieve an accuracy of 49.41% and 35.78%. Another comparison is using textual features, where the ST-MFLC model achieves an accuracy of 22.83%, while when applied in the proposed method used in this study, the accuracy can reach 63.63%. Based on these results, it can be seen that for Yelp and Amazon data, the proposed method is better at detecting fake reviews, both textually and multi-feature, compared to several previous studies.

In addition, the use of multi-features also supports the achievement of excellent performance results. These results are proven by comparing the accuracy of using text only and multi-features. The identification results using text features only achieved an accuracy of 73.63% in the Yelp domain as training data. This shows that there is a clear difference in accuracy between the use of multi-features and text-based features alone. The difference between the two approaches is 24.96%, which proves that the addition of behavioural features in identifying fake reviews, especially cross-domain, can improve the performance of the method used.

References

Alsubari, S. N., Deshmukh, S. N., Al-Adhaileh, M. H., Alsaade, F. W., & Aldhyani, T. H. H. (2021). Development of Integrated Neural Network Model for Identification of Fake Reviews in E-Commerce Using Multidomain Datasets. *Applied Bionics and Biomechanics*, *2021*. https://doi.org/10.1155/2021/5522574

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1). https://doi.org/10.1186/s40537-021-00444-8

Andresini, G., Iovine, A., Gasbarro, R., Lomolino, M., de Gemmis, M., & Appice, A. (2022). EUPHORIA: A neural multi-view approach to combine content and behavioral features in review spam detection. *Journal of Computational Mathematics and Data Science*, *3*, 100036. https://doi.org/10.1016/j.jcmds.2022.100036

Asudani, D. S., Nagwani, N. K., & Singh, P. (2023). Impact of word embedding models on text analytics in deep learning environment: a review. *Artificial Intelligence Review*, *56*(9), 10345–10425. https://doi.org/10.1007/s10462-023-10419-1

Barushka, A., & Hajek, P. (2019). Review Spam Detection Using Word Embeddings and Deep Neural Networks. *IFIP Advances in Information and Communication Technology*, *559*, 340–350. https://doi.org/10.1007/978-3-030-19823-7_28

Birim, Ş. Ö., Kazancoglu, I., Kumar Mangla, S., Kahraman, A., Kumar, S., & Kazancoglu, Y. (2022). Detecting fake reviews through topic modelling. *Journal of Business Research*, *149*, 884–900. https://doi.org/10.1016/j.jbusres.2022.05.081

Budhi, G. S., Chiong, R., & Wang, Z. (2021). Resampling imbalanced data to detect fake reviews using machine learning classifiers and textual-based features. *Multimedia Tools and Applications*, *80*(9), 13079–13097. https://doi.org/10.1007/s11042-020-10299-5

Cao, N., Ji, S., Chiu, D. K. W., & Gong, M. (2022). A deceptive reviews detection model: Separated training of multi-feature learning and classification. *Expert Systems with Applications*, *187*. https://doi.org/10.1016/j.eswa.2021.115977

Chen, J., Zhang, T., Yan, Z., Zheng, Z., Zhang, W., & Zhang, J. (2025). Attention-based BiLSTM with positional embeddings for fake review detection. *Journal of Big Data*, *12*(1). https://doi.org/10.1186/s40537-025-01130-9

Deshai, N., & Bhaskara Rao, B. (2023). Unmasking deception: a CNN and adaptive PSO approach to detecting fake online reviews. *Soft Computing*, *27*(16), 11357–11378. https://doi.org/10.1007/s00500-023-08507-z

Duma, R. A., Niu, Z., Nyamawe, A. S., Tchaye-Kondi, J., & Yusuf, A. A. (2023). A Deep Hybrid Model for fake review detection by jointly leveraging review text, overall ratings, and aspect ratings. *Soft Computing*, *27*(10), 6281–6296. https://doi.org/10.1007/s00500-023-07897-4

Elmogy, A. M., Tariq, U., Ibrahim, A., & Mohammed, A. (2021). Fake Reviews Detection using Supervised Machine Learning. *International Journal of Advanced Computer Science and Applications*, *12*(1), 601–606. https://doi.org/10.14569/IJACSA.2021.0120169

Fathima Beevi, P. S., Abdul Gafur, M., & Haroon, R. P. (2023). Identification of Fake Reviews: A Review. *Proceedings - 2023 International Conference on Innovations in Engineering and Technology, ICIET 2023*. https://doi.org/10.1109/ICIET57285.2023.10220907

Gupta, R., Jindal, V., & Kashyap, I. (2024). Recent state-of-the-art of fake review detection: a comprehensive review. In *Knowledge Engineering Review* (Vol. 39). Cambridge University Press. https://doi.org/10.1017/S0269888924000067

Hajek, P., Barushka, A., & Munk, M. (2020). Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. *Neural Computing and Applications*, *32*(23), 17259–17274. https://doi.org/10.1007/s00521-020-04757-2

He, L., Wang, X., Chen, H., & Xu, G. (2022). Online Spam Review Detection: A Survey of Literature. *Human-Centric Intelligent Systems*, *2*(1–2), 14–30. https://doi.org/10.1007/s44230-022-00001-3

Hussain, N., Turab Mirza, H., Hussain, I., Iqbal, F., & Memon, I. (2020). Spam Review Detection Using the Linguistic and Spammer Behavioral Methods. *IEEE Access*, *8*, 53801–53816. https://doi.org/10.1109/ACCESS.2020.2979226

Lainder, A. D., & Wolfinger, R. D. (2022). Forecasting with gradient boosted trees: augmentation, tuning, and cross-validation strategies: Winning solution to the M5 Uncertainty competition. *International Journal of Forecasting*, *38*(4), 1426–1433. https://doi.org/10.1016/j.ijforecast.2021.12.003

Le, T. K. H., Li, Y. Z., & Li, S. T. (2022). Do Reviewers' Words and Behaviors Help Detect Fake Online Reviews and Spammers? Evidence From a Hierarchical Model. In *IEEE Access* (Vol. 10, pp. 42167–42183). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2022.3167511

Liu, J., Quan, P., & Zhang, W. (2024). A Study on Fake Review Detection Based on RoBERTa and Behavioral Features. *Procedia Computer Science*, *242*, 1323–1330. https://doi.org/10.1016/j.procs.2024.08.131

Liu, Y., Wang, L., Shi, T., & Li, J. (2022). Detection of spam reviews through a hierarchical attention architecture with N-gram CNN and Bi-LSTM. *Information Systems*, *103*. https://doi.org/10.1016/j.is.2021.101865

Manaskasemsak, B., Tantisuwankul, J., & Rungsawang, A. (2021). Fake review and reviewer detection through behavioral graph partitioning integrating deep neural network. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-021-05948-1

Martens, D., & Maalej, W. (2019). Towards understanding and detecting fake reviews in app stores. *Empirical Software Engineering*, *24*(6), 3316–3355. https://doi.org/10.1007/s10664-019-09706-9

Mewada, A., & Dewang, R. K. (2023). A comprehensive survey of various methods in opinion spam detection. *Multimedia Tools and Applications*, *82*(9), 13199–13239. https://doi.org/10.1007/s11042-022-13702-5

Mohawesh, R., Bany Salameh, H., Jararweh, Y., Alkhalaileh, M., & Maqsood, S. (2024). Fake review detection using transformer-based enhanced LSTM and RoBERTa. *International Journal of Cognitive Computing in Engineering*, *5*, 250–258. https://doi.org/10.1016/j.ijcce.2024.06.001

Mohawesh, R., Xu, S., Springer, M., Jararweh, Y., Al-Hawawreh, M., & Maqsood, S. (2023). An explainable ensemble of multi-view deep learning model for fake review detection. *Journal of King Saud University - Computer and Information Sciences*, *35*(8). https://doi.org/10.1016/j.jksuci.2023.101644

Mohawesh, R., Xu, S., Tran, S. N., Ollington, R., Springer, M., Jararweh, Y., & Maqsood, S. (2021). Fake Reviews Detection: A Survey. In *IEEE Access* (Vol. 9, pp. 65771–65802).

Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2021.3075573

Pal, K., & Patel, B. V. (2020). Data Classification with k-fold Cross Validation and Holdout Accuracy Estimation Methods with 5 Different Machine Learning Techniques. *Proceedings of the 4th International Conference on Computing Methodologies and Communication, ICCMC 2020*, 83–87. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-00016

Qayyum, H., Ali, F., Nawaz, M., & Nazir, T. (2023). FRD-LSTM: a novel technique for fake reviews detection using DCWR with the Bi-LSTM method. *Multimedia Tools and Applications*, *82*(20), 31505–31519. https://doi.org/10.1007/s11042-023-15098-2

Raheem, M., & Chong, Y. C. (2024). E-Commerce Fake Reviews Detection Using LSTM with Word2Vec Embedding. *Journal of Computing and Information Technology*, *32*(2), 65–80. https://doi.org/10.20532/cit.2024.1005803

Rayana, S., & Akoglu, L. (2015). Collective opinion spam detection: Bridging review networks and metadata. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *2015-August*, 985–994. https://doi.org/10.1145/2783258.2783370

Ren, G., Wang, H., & Yang, Y. (2025). Cross-domain Fake Review Detection Based on Deep Learning MultiLevel Generic Features Extraction Fusion. *Informatica (Slovenia)*, *49*(18), 99–110. https://doi.org/10.31449/inf.v49i18.7071

Saumya, S., & Singh, J. P. (2018). Detection of spam reviews: a sentiment analysis approach. *CSI Transactions on ICT*, *6*(2), 137–148. https://doi.org/10.1007/s40012-018-0193-0

Shinde, S. A., Pawar, R. R., Jagtap, A. A., Tambewagh, P. A., Rajput, P. U., Mali, M. K., Kale, S. D., & Mulik, S. V. (2023). Deceptive opinion spam detection using bidirectional long short-term memory with capsule neural network. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-023-17348-9

Singhal, R., & Kashef, R. (2023). A Weighted Stacking Ensemble Model With Sampling for Fake Reviews Detection. *IEEE Transactions on Computational Social Systems*. https://doi.org/10.1109/TCSS.2023.3268548

Tang, X., Qian, T., & You, Z. (2020). Generating behavior features for cold-start spam review detection with adversarial learning. *Information Sciences*, *526*, 274–288. https://doi.org/10.1016/j.ins.2020.03.063

Wang, J., Kan, H., Meng, F., Mu, Q., Shi, G., & Xiao, X. (2020). Fake review detection based on multiple feature fusion and rolling collaborative training. *IEEE Access*, *8*, 182625–182639. https://doi.org/10.1109/ACCESS.2020.3028588

Wei, C. S., Hsu, P. Y., Huang, C. W., Cheng, M. S., & Prassida, G. F. (2020). Devising a Cross-Domain Model to Detect Fake Review Comments. *Communications in Computer and Information Science*, *1287*, 714–725. https://doi.org/10.1007/978-3-030-63119-2_58

Xiang, L., You, H., Guo, G., & Li, Q. (2023). Deep feature fusion for cold-start spam review detection. *Journal of Supercomputing*, *79*(1), 419–434. https://doi.org/10.1007/s11227-022-04685-z

Zhang, D., Li, W., Niu, B., & Wu, C. (2023). A deep learning approach for detecting fake reviewers: Exploiting reviewing behavior and textual information. *Decision Support Systems*, *166*. https://doi.org/10.1016/j.dss.2022.113911