JOIV : Int. J. Inform. Visualization, 9(2) - March 2025 761-769

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Identifying Fraud Sellers in E-Commerce Platform

Lovesh Anand [a], Hui-Ngo Goh [a,*], Choo-Yee Ting [a], Albert Quek [a]

[a] *Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, Cyberjaya, Malaysia*

Corresponding author: [*]hngoh@mmu.edu.my

***Abstract*—Identifying fake reviews in e-commerce is crucial as they might impact buyers' purchasing decisions and overall satisfaction. This work investigates the effectiveness of machine learning and transformer-based models for detecting fake reviews on the Amazon Fake Review Labelled Dataset. The dataset contains 20,000 computer-generated and 20,000 original reviews across various product categories with no missing value. In this study, machine learning and transformer-based models were compared, revealing that transformer-based models outperformed in detecting fake reviews, achieving an accuracy of 98% with the DistilBERT model. Additionally, this work too examines the impact of word embedding on machine learning models in enhancing fake review detection accuracy. The results show that the word embedding model Word2Vec displays notable improvements, achieving accuracies of 92% with SVM and 90% with Random Forest and Logistic Regression. Furthermore, a comparison study was carried out on comparing transformer models from previous work, which utilized the same full dataset; it was found that the DistilBERT model produced comparable accuracy despite its lighter architecture. In summary, this study underscores the effectiveness of transformer-based models and machine learning models in detecting fake reviews while at the same time highlighting the importance of word embedding techniques in enhancing the performance of machine learning models. This work is hoped to contribute to combating fake reviews and fostering trust in e-commerce platforms.**

***Keywords*—E-commerce; fake reviews; transformer-based models; Amazon fake review labelled dataset; machine learning.**

Manuscript received 29 Sep. 2024; revised 29 Nov. 2024; accepted 10 Mar. 2025. Date of publication 31 Mar. 2025.

I. Introduction

In the realm of online commerce, the prevalence of smartphone usage has propelled platforms like Amazon, eBay, Taobao, and Shopee to serve millions, if not billions, of users worldwide [1]. These platforms have become integral in connecting customers with online sellers and offline factories, contributing significantly to the global economy. Key to their success is the reputation ranking system, with Amazon generating around 475 billion dollars and Alibaba 1 trillion dollars in gross merchandise volume (GMV) in 2020 alone [1]. However, challenges arise from dishonest practices, particularly the proliferation of fake reviews, which can mislead buyers and distort market dynamics. As such, this work endeavors to develop a fake review detection model integrated with sentiment analysis to combat fraudulent behaviors on e-commerce platforms, specifically focusing on Amazon. By identifying and flagging fake reviews, the aim is to foster a trustworthy and scam-free environment, ultimately enhancing the buying experience for consumers.

Between the rise of online shopping, the reliance on product reviews as a decision-making tool has surged, with studies indicating that a significant majority of consumers trust these reviews [2]. However, this reliance has been exploited by unscrupulous sellers who engage in deceptive practices to manipulate ratings and reviews. Such practices undermine consumer trust and distort market competition [3]. Consequently, there has been a growing interest in utilizing automated methods within the Natural Language Processing (NLP) field to detect and combat fake reviews [4]. Although various models, including machine learning and transformer models, have been developed for this purpose, there remains a notable gap in research concerning the integration of word embeddings. Based on the current methods for fake review detection, it is found that researchers have not explored different external word embeddings to improve the accuracy of fake review detection for machine learning methods. For instance, the frequent word embedding used for machine learning models based on the existing work is Term Frequency-Inverse Document Frequency (TF-IDF). To the best of my knowledge, word embeddings were not explicitly stated in most of the papers. One of the papers even used readability features. Therefore, to address the gap, external word embeddings like GloVe, Word2Vec, TF-IDF, and Count

Vectorizer are added to this work to experiment with whether the combination of external word embeddings in machine learning classifiers could potentially enhance the accuracy of fake review detection.

The research questions include evaluating the effectiveness of machine learning and transformer models in detecting fake reviews, identifying fraudulent sellers, and assessing the impact of word embeddings on accuracy. The objectives are to compare model effectiveness, identify sellers, and evaluate word embedding effects. The selected classifiers are Support Vector Machine (SVM), Logistic Regression (LR), and Random Forest (RF) as they are well-studied and effective in detecting fake reviews. Although BERT is commonly used, this study opts for DistilBERT due to time constraints, as it provides similar functionality.

This integrated approach aims to contribute to the creation of a more transparent and trustworthy online shopping environment, benefiting both consumers and legitimate sellers alike. The rest of the section is structured as follows. In Section 2, we discuss the proposed framework, whereas Section 3 points out the results obtained upon experimenting with the framework. Finally, Section 4 concludes the whole work.

II. Materials and Method

In this section, we examined the distinction between fake and genuine reviews. As the flow diagram in Figure 1 shows, we follow a systematic approach.

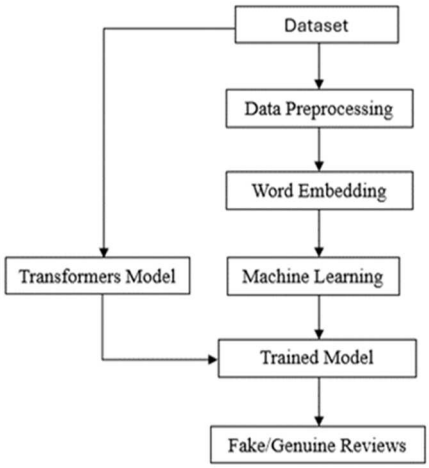

Fig. 1 Proposed Framework Diagram

A. Dataset

A fake review dataset from an Open Science Framework (OSF) by [5] is used for this work. The dataset consists of various product reviews from different categories like Books, Clothing, Electronics, Home and Kitchen, Kindle Store, Movies and TV, Pet Supplies, Sports and Outdoors, Tools, Toys, and Games. This dataset contains 20 K computer-generated (CG) reviews and 20 K original reviews (OR). The dataset also has information on the product ratings that scales from 1 to 5. There are a total of 40432 rows with four attributes in the dataset. The attributes present in the dataset are mentioned earlier, which are “category,” “rating,” “label,” and “text_.” The label indicates whether the review is CG or OR. Table I shows the sample dataset that is used for this work.

TABLE I
Sample Dataset

Category	Rating	Label	Text_
Home_and_Kitchen_	5	CG	Love this! Well made, sturdy, and very comfortable. I love it!Very pretty
Sports_and_Outdoors_5	4	CG	The hat fit a little tight on the head, but I'm not sure if the size fits.
Electronics_5	3	OR	I bought this for the TTL ... then I realised that the best results can be achieved only in Manual ...
Movies_and_TV_5	2	OR	Not very good. It sounds okay but not great choice of songs. Also it's made from photographs not film so not real footage.
Books_5	1	CG	This a a great book and an easy read. I will keep my eyes peeled for the next book

B. Data Preprocessing

When dealing with reviews, they must be pre-processed to make them suitable for feeding into the models. Without these preprocessing steps, the performance of trained models will be affected, and the computational costs will increase as well [2]. The preprocessing steps involved are as follows:

1) Stop words Removal: In text classification, the approach of removing stop words is made to remove common words such as "the," "a," "am," and "our" that do not significantly contribute any meaning to the texts. These words are known as stop words. Removing stop words allows us to focus on the essential words that improve our understanding of the text [2] [6] [7]. Hence, to focus on the important words in the review text, the stop words have been removed in this project. The NLTK stop-word library from Python was used to remove all the stop words that are present in the textual reviews.

2) Punctuation Removal: Punctuation and special characters are like noise in text analysis, making it more difficult to classify or interpret the text [2]. Punctuation, such as commas and exclamation marks, helps humans understand the ideas and sentiments expressed in a text. However, punctuation does not contribute significantly to machines providing better classification performance [7]. Therefore, to improve the fake review detection performance, punctuation was removed in this project. Python's regular expressions, ' re' and' string', were used to remove the punctuation.

3) Convert Text to Lowercase: A common method that is used to preprocess text in natural language processing is by converting all the letters to [8]. Lowercasing text is essential in text analysis since it enables us to compare the words

accurately and speeds up the analysis process [2]. By changing all the text to lowercase, we can make sure words with different capitalizations are considered the same, making it easier to find patterns and similarities in the data. For that reason, reviews were converted from uppercase to lowercase letters in this project. Python's lower () method was used to turn the words in the reviews into lowercase letters.

4) Lemmatization: Lemmatization is a method that reduces words to their most basic form. By having the basic forms, it simplifies the analysis of words in a sentence [7] [9]. Therefore, this project used lemmatization to ease the analysis of the words in reviews using the NLTK word lemmatizer Python library.

5) Tokenization: The tokenization process involves splitting the text into individual words or phrases called tokens. In this way, the machine better understands the words inside a sentence. It's similar to breaking down a sentence into smaller parts for the machine to understand easily. Tokenization is mainly done to improve the effectiveness and efficiency of the model training [2] [10]. For this purpose, tokenization was performed in this project with the help of NLTK word tokenizer library from Python.

C. Word Embeddings

A word embedding method uses numbers to teach machines what the words in a text mean. Words that have similar meanings or relationships have numbers closer to one another [11]. For instance, the number representation for the words 'shirt' and 'trousers' might be closer than those for 'shirt' and 'toys'. In this work, several word embeddings were used before feeding the data into the training model such as Count Vectorizer, Term Frequency - Inverse Document Frequency (TF-IDF), Word to Vector (Word2Vec), Global Vector (GloVe).

1) Count Vectorizer: Count Vectorizer is a way to convert documents of different lengths into fixed-length vector forms. This is done by representing each word as a column in a matrix and each sentence as a row in the matrix. In this way, text data can be converted into numerical representation, which helps feed into machine learning models [12]. Therefore, Count Vectorizer was used as one of the embeddings in this work together with machine learning models. The 'CountVectorizer' method from Python's sci-kit-learn library was used to transform the reviews into numerical representations. Table II shows how texts are represented in the document term matrix.

TABLE II
DOCUMENT TERM MATRIX

Words / Sentence	I	love	this	mobile	He
I love this mobile	1	1	1	1	0
He love mobile	0	1	0	1	1

2) TF-IDF: The TF-IDF method, an extension of the Count Vectorizer, utilizes word counts in a document term matrix to extract relevant information. It's crucial for converting text data into a mathematical format suitable for machine learning models [8] [9]. TF-IDF computation involves multiplying TF (Term Frequency) with IDF (Inverse Document Frequency). While TF measures word frequency in a document, it doesn't consider document length variations, leading to biased results [11]. IDF rectifies this bias by weighing rare but informative terms more than common ones, thus enhancing the extraction of meaningful information from documents [8] [11]. IDF is computed using a logarithmic function based on the ratio of total documents in a corpus to the number containing a specific term. In this work, TF-IDF is an embedding technique that converts reviews into a mathematical format compatible with machine learning models. The 'TfidfVectorizer' method from Python's scikit-learn library transforms reviews into vectors before applying machine learning models for fake review detection.

3) Word2Vec: Word2Vec, a neural network-based method introduced by Mikolov, Chen, Corrado, and Dean in 2013, aims to capture the semantic relationships between words. It comprises two versions: Continuous Bag of Words (CBoW) and Skip-Gram (SG). While SG predicts context words given a target word, CBoW predicts the target word based on existing context words [8]. In SG, the input layer feeds the target word to the projection layer, followed by the output layer containing context words. Conversely, CBoW takes all context words as input and passes them to the projection layer before reaching the output layer [8]. Word2Vec offers several advantages, including quickly generating high-quality word embeddings, even with large datasets. Moreover, it effectively handles out-of-vocabulary words and misspellings, making it suitable for real-world applications [13]. In this work, the CBoW model of Word2Vec was used to convert each word in the review into a list of numbers, where similar words yield similar lists. The Average Word2Vec method was then applied to average the lists of numbers for words in each review, producing a final list of numbers. Additionally, the Word2Vec model was trained using the Gensim library in Python. Figures 2 and 3 show the CBOW and Skip-gram models, respectively.

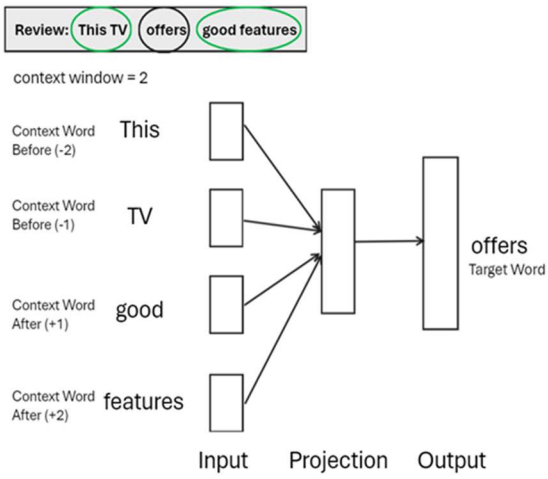

Fig. 2 CBoW Model

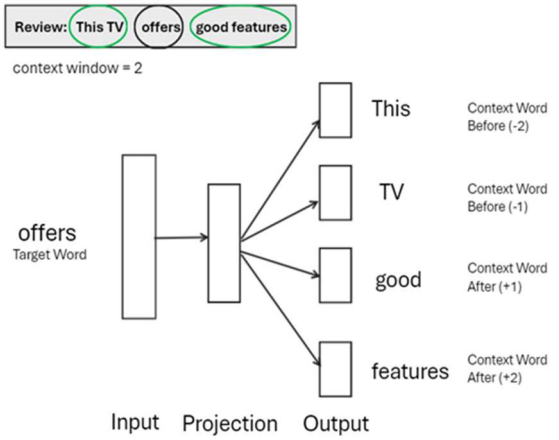

Fig. 3 Skip-Gram Model

4) GloVe: GloVe, an unsupervised learning method, is employed to capture word vector representations by establishing clear word contexts throughout the document corpus using statistics [14] [13]. The method relies on the co-occurrence matrix, denoted as C, which records the number of windows during the simultaneous occurrence of target word wc and context word wt. Consider an example with a window size of 1, where each unique word is assigned with a number based on its occurrence. For instance, in the sentences "He loves big Smart TV" and "He hates small Smart TV", if "Smart" is the target word, the window's content would be determined. Words like "big" and "small" occur once, while "TV" occurs twice within this window. The co-occurrence matrix, C, reflects these counts. For example, "big" and "small" receive a value of 1, and "TV" gets 2, reflecting their occurrences with "Smart" in the sentences. Utilizing this matrix, word relationships are established. GloVe offers the advantage of producing high-quality word embeddings for large datasets with minimal computational resources [13]. Figure 4 illustrates the Co-occurrence matrix. In this work, a pre-trained embedding vector from Stanford's GloVe, featuring 100 dimensions and 6 billion tokens, was utilised to generate word vectors for each review.

			Context						
		Word Number	big	hates	loves	Smart	small	He	TV
Target	big	1	0	0	1	1	0	0	0
	hates	2	0	0	0	0	1	1	0
	loves	3	1	0	0	0	0	1	0
	Smart	4	1	0	0	0	1	0	2
	small	5	0	1	0	1	0	0	0
	He	6	0	1	1	0	0	0	0
	TV	7	0	0	0	2	0	0	0

Fig. 4 Co-Occurrence Matrix

D. Machine Learning Architecture

1) Support Vector Machine (SVM): SVM is a supervised learning algorithm that learns from labelled data. Its prominent role is finding the best possible separating line or what is called the hyperplane between different groups in the training data [6]. SVM aims to find a hyperplane that maximizes the distance between different groups. In other words, the more significant the gap or margin among these groups, the better the classifier does its job, and fewer errors could be made when classifying new data [15]. Additionally, SVMs are usually useful when there are many features or high-dimensional areas and use only a small portion of the training data to make accurate decisions. This makes SVM memory efficient [16]. Therefore, in this work, SVM was chosen as one of the machine learning classifiers for the fake review detection model. Figure 5 illustrates the samples on the margin and training data points of two groups called support vectors.

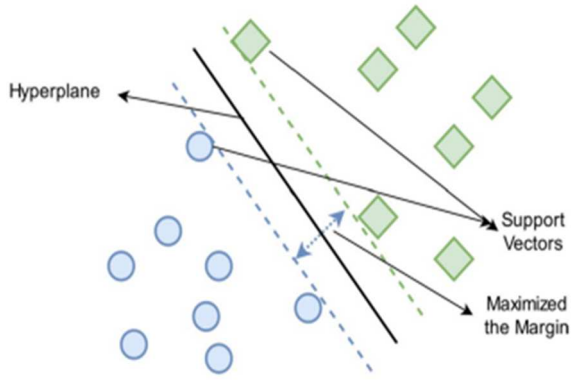

Fig. 5 Sample on Margin and Support Vector for Classifying Data [16]

2) Random Forest (RF): A machine learning algorithm designed to address overfitting issues encountered by Decision Trees. It constructs a collection of decision trees using different datasets, introducing variation by creating each tree with slight differences in its dataset [6]. Each tree in the RF is built with a limited number of features, ensuring diversity. The accuracy of the RF, also known as an ensemble model, relies on the performance of individual trees and their correlation with each other [15]. This ensemble model is adept at handling outliers and noise and finds application in various domains, including text processing. RF offers several advantages, such as its effectiveness in dealing with many features, preventing overfitting by creating classification rules with a small amount of data, and its fast operation speed, coupled with excellent classification performance [15]. Hence, RF was chosen as one of the models to train the fake review detection model in this project. During the training process, smaller groups of training sets are selected using Bootstrap Sampling. These subsets are then used to build decision trees, and the results from each tree are combined through majority voting to make predictions [17]. Figure 6 illustrates the operation of Random Forest.

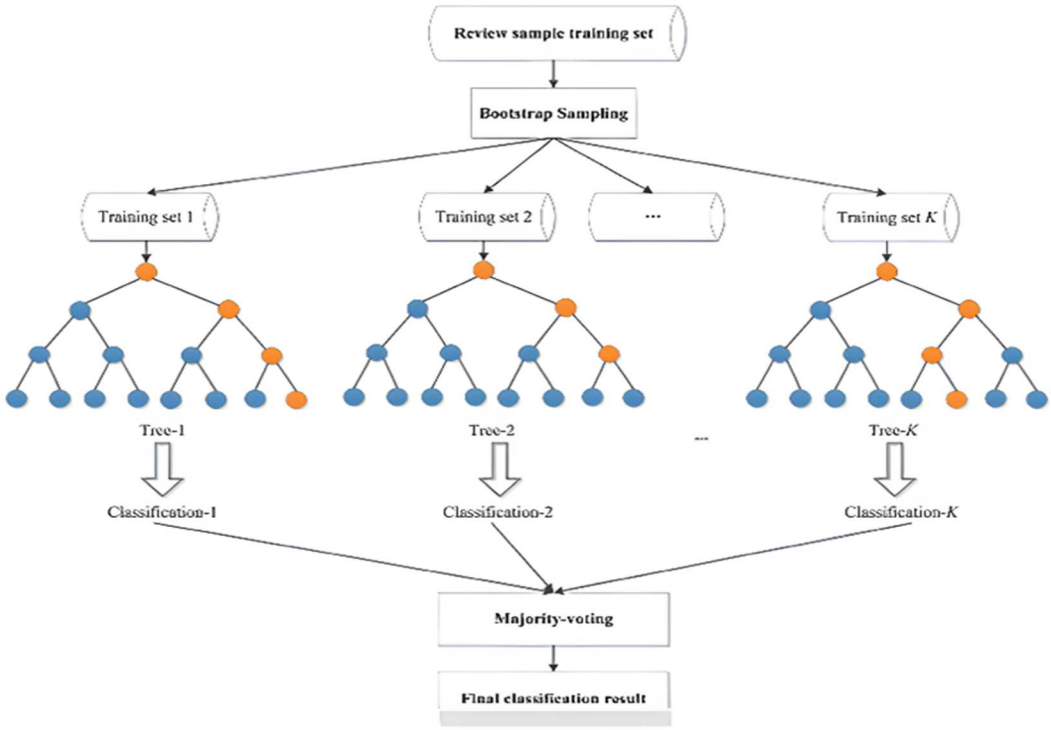

Fig. 6 Random Forest operations [17]

3) Logistic Regression (LR): A type of supervised learning in machine learning that operates on labeled data to determine the optimal separating line or hyperplane that classifies training data into different classes [6]. It employs functions like the logistic function or log function to create the hyperplane between distinct data types or classes [18]. Logistic Regression is particularly suited for scenarios where the dependent variable is binary, with two possible outcomes (e.g., 1 or 0), while the independent variables can be categorical or numerical [16]. Logistic Regression resembles Linear Regression but is tailored for classification tasks, unlike Linear Regression, which is used for regression problems such as predicting continuous values [19]. Given that the work involves classifying reviews as fake or genuine, Logistic Regression was chosen as one of the classification models. Figure 7 illustrates the logistic function or sigmoid function of Logistic Regression, which maps real number values between 0 and 1, producing an S-shaped curve.

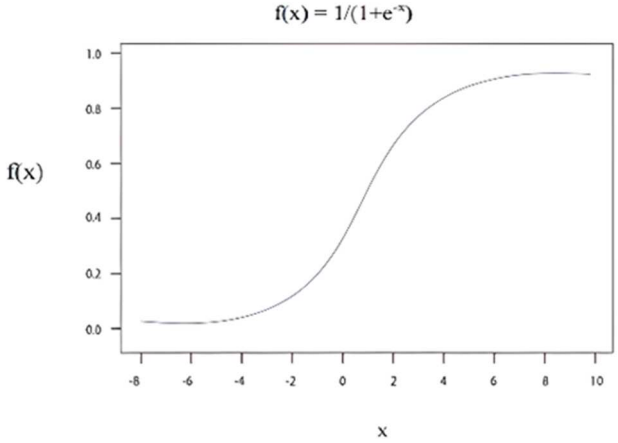

Fig. 7 Logistic Function or Sigmoid Function of Logistic Regression [19]

E. Transformer-Based Architecture

DistilBERT: A simplified version of BERT, aims to overcome some of BERT's limitations, such as computational heaviness, fixed input sizes, and issues with word piece embeddings [13]. While retaining the overall structure of BERT, DistilBERT reduces the number of layers and eliminates specific components like token type embeddings and the pooler, making it lighter than BERT [13]. Despite its reduced size, DistilBERT has demonstrated effectiveness in tasks like text classification, maintaining 97% of BERT's performance on the General Language Understanding Evaluation (GLUE) benchmark tasks [4]. Consequently, in this work, DistilBERT was employed as a transformer model for detecting fake reviews and to evaluate its effectiveness compared to machine learning models integrating external word embeddings such as Word2Vec, Count Vectorizer, TF-IDF, and GloVe.

III. Result and Discussion

This section presents the results of the fake review detection model and some exploratory data analysis and discusses accordingly.

F. Exploratory Data Analysis (EDA)

Data Analysis was performed using the Amazon Labelled Dataset to detect fake reviews. The insights obtained from the data analysis will be discussed in the following section.

1) Dataset Characteristics: The dataset utilized in this work, as shown in Table III, does not contain any missing values. Therefore, handling missing values was unnecessary during the preprocessing stage. The attributes in the dataset encompass a variety of data types, including three object data types and one float data type. The "rating" attribute, with

values ranging from 1 to 5, is of float data type, representing the scores given in the reviews. The remaining object data types include "category," which denotes the product category; "label," serving as the identifier for fake or original reviews; and "text_," representing the text of the product reviews provided by customers.

TABLE III
DATASET CHARACTERISTICS

Column	Data type	Missing values
category	object	None
rating	float	None
label	object	None
Text_	object	None

2) Duplicated Values: The dataset does not contain duplicate values, so there is no need to handle duplicate values.

3) Lowercasing Reviews: Table IV shows the review text before and after performing lowercasing. This is important as it ensures that the same word in different cases is interpreted as the same term when lowercasing is performed.

TABLE IV
LOWERCASING REVIEWS BEFORE AND AFTER

Before Lowercasing	After Lowercasing
Love this! Well made, sturdy, and very comfortable. I love it!Very pretty	love this! well made, sturdy, and very comfortable. i love it!very pretty

4) Punctuation Removal: Table V depicts the review texts with and without punctuation. Review text with punctuation like "Love this!" will not be helpful when performing word counts or searching for a word; therefore, punctuation should be removed to make it easier.

TABLE V
REVIEWS WITH AND WITHOUT PUNCTUATION

With Punctuation	Without Punctuation
Love this! Well made, sturdy, and very comfortable. I love it!Very pretty	love this well made sturdy and very comfortable i love itvery pretty

5) Tokenization: Table VI shows the review text with and without tokenization. Tokenization helps continue the text cleaning process, such as stop words removal and word lemmatization, by applying them to each token separately. Tokenization also helps in word embeddings, which allow fake review detection models to work with text data by converting them to numerical values.

TABLE VI
REVIEWS WITH AND WITHOUT TOKENIZATION

Without Tokenization	With Tokenization
Love this! Well made, sturdy, and very comfortable. I love it!Very pretty	['love', 'this', 'well', 'made', 'sturdy', 'and', 'very', 'comfortable', 'i', 'love', 'itvery', 'pretty']

6) Stop Words Removal: As observed in Table VII, commonly used words like "and," "I," and "this" were removed from the original text without stop words removal. Although these words are useful in understanding the flow of a sentence, they do not convey the main context of the text, which would be more beneficial for fake review detection.

TABLE VII
REVIEWS WITH AND WITHOUT STOP WORDS REMOVAL

Without Stop Words Removal	With Stop Words Removal
Love this! Well made, sturdy, and very comfortable. I love it!Very pretty	love well made sturdy comfortable love itvery pretty

7) Word Lemmatization: Lemmatization is a technique that reduces words to its root form. Table VIII shows how the words are reduced to its root form. For instance, words like "cloths" become "cloth" and "towels" become "towel". Lemmatization is particularly useful in making the vocabulary throughout the reviews to be standardized which could help in identifying patterns in fake reviews.

TABLE VIII
REVIEWS WITH AND WITHOUT WORD LEMMATIZATION

Without Lemmatization	With Lemmatization
Super rough, not soft wash cloths, more like bar towels	super rough soft wash cloth like bar towel

8) Checking Review Length Before and After Cleaning: Checking the review length before cleaning will offer insight into how much the data cleaning process influenced the reviews. For example, if the text length is reduced after cleaning steps are performed, the data cleaning process has removed unnecessary content from the text. Figure 8 shows that the length of text reviews was decreased after the cleaning process was done. This suggests that the cleaning process effectively removed unnecessary and not useful words. Reducing the length of reviews can improve the trained model's performance as they would deal with less noise in the review text.

Review_Text	Length_Before	Cleaned_Review	Length_After
Love this! Well made, sturdy, and very comfor...	12	love well made sturdy comfortable love pretty	7
love it, a great upgrade from the original. I...	16	love great upgrade original mine couple year	7
This pillow saved my back. I love the look and...	14	pillow saved back love look feel pillow	7
Missing information on how to use it, but it i...	17	missing information use great product price	6
Very nice set. Good quality. We have had the s...	18	nice set good quality set two month	7

Fig. 8 Review Length after Cleaning

9) Changing Labels to Numeric Form: Mapping of categorical labels to numerical labels where Computer Generated Labels (CG) are mapped to 1 and Original Reviews Labels (OR) are mapped to 0. This was done in this work mainly because many machine learning models require numerical input data and do not work with categorical data.

10) Label Distribution of the Fake Reviews: Figure 9 shows the distribution of different labels in the dataset. The Label consists of two categories Computer Generated (CG) or Original Review (OR). From Figure 9, the dataset is balanced with an equal representation of CG and OR categories. This means that there are no problems like class imbalances to handle. This is useful in improving the accuracy of detecting fake reviews as the models can learn from all the classes and avoid being biased towards a majority class.

11) Labels Distribution over Categories: Figure 10 displays the count plot for label distributions across different product categories. The count plot shows a balanced distribution of Computer Generated (CG) and Original Reviews (OR) across various product categories. An

imbalanced distribution can affect the performance of the classification models trained for fake review detection.

12) Rating Distribution over Categories: The count plot in Figure 11 shows how many rating stars were given from 1 to 5 for different product categories. From this plot, there are many 5-star ratings surrounding every category. This can be interpreted as customers often leaving more positive reviews; however, it can also indicate that not all the reviews may be genuine. On the other hand, fewer low-rated stars could also be observed, such as not many 1-star and 2-star ratings being given. This could suggest that the negative ratings are genuinely rated or being removed by the sellers to maintain a good reputation.

13) Labels Distribution over Ratings: Figure 12 shows the count plot of the CG and OR distribution across different rating stars ranging from 1 to 5. This count plot shows that as the rating star increases from 1 to 5, the number of computer-generated reviews increases too. This suggests that customers should be more aware and look more closely at the positive reviews posted on the e-commerce platform, as sellers might manipulate the ratings by posting a fake review to boost their reputation and gain customers' trust.

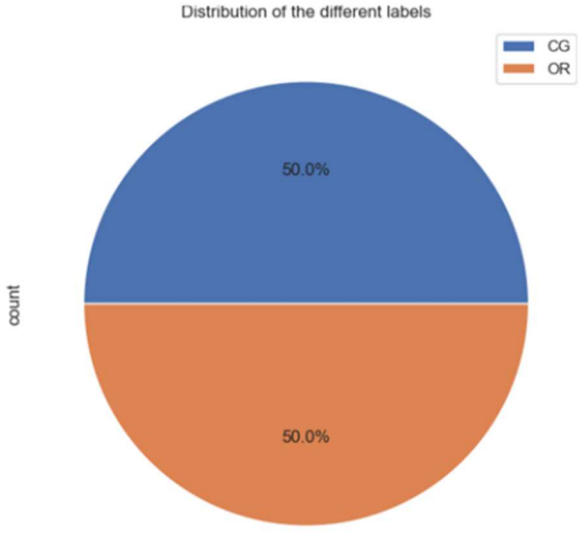

Fig. 9 Pie Chart Label Distribution of the Fake Reviews

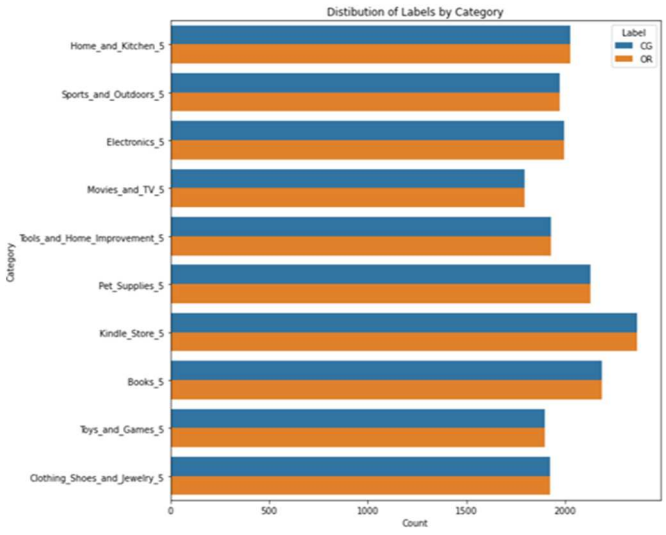

Fig. 10 Count Plot for Label Distribution over Categories

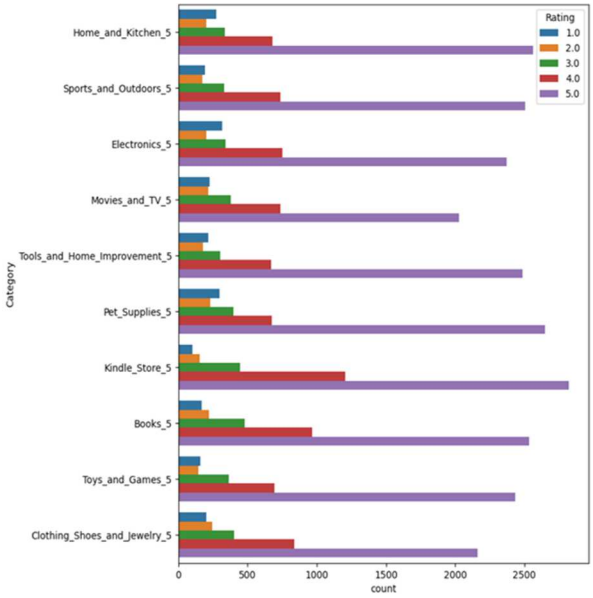

Fig. 11 Count Plot for Rating Distribution over Categories

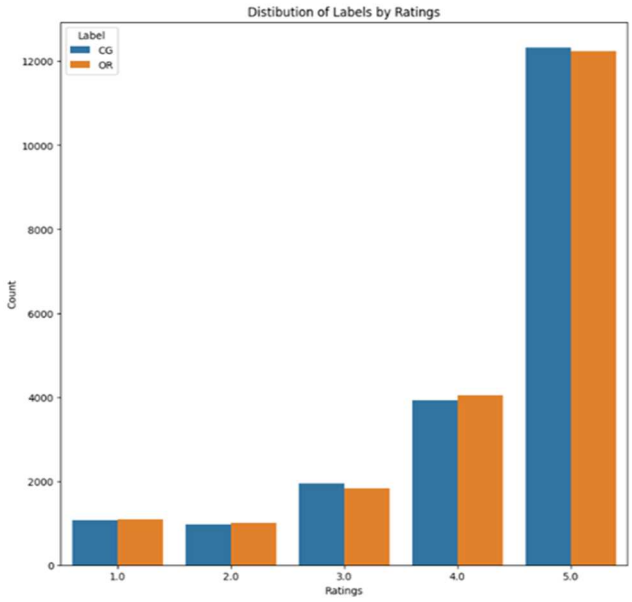

Fig. 12 Count Plot for Label Distribution over Ratings

G. Comparison Results of Transformer Model and Machine Learning Models

According to the comparison table in Table IX, the transformer model, namely the DistilBERT model, was the most effective in detecting fake reviews with an accuracy of 98%. It has outperformed all the other machine learning models used, regardless of the word embeddings like Count Vectorizer, TF-IDF, Word2Vec, and GloVe. While traditional machine learning models like Random Forest, Support Vector Machine (SVM), and Logistic Regression showed competitive accuracies with various word embedding techniques, but they did not match the performance of the DistilBERT model. The Support Vector Machine (SVM) with Word2Vec was the only model that came closest to the DistilBERT model with 92% accuracy, followed by Logistic Regression (LR) and Random Forest (RF) with 90% accuracy, respectively. This proves that DistilBERT can understand the context of the review text better with its

contextualized embedding when compared to the word embedding used for machine learning models. Hence, the best framework uses DistilBERT as a fake review detection model.

TABLE IX
COMPARISON TABLE BETWEEN MODELS USED IN THIS AND PREVIOUS STUDY

Word Embeddings	Accuracy (%) Proposed Framework			
	SVM	RF	LR	DistilBERT
Count Vectorizer	85%	86%	88%	98%
TF-IDF	89%	85%	87%	
Word2Vec	92%	90%	90%	
GloVe	81%	78%	78%	

H. Comparison of Proposed Framework with Previous Work

Based on Table X, the comparison was made with the work done by [20], which utilizes transformer models like BERT, XLNet, and DeBERTa for fake review detection using the same dataset used in this work, the OSF fake reviews dataset. This study fully utilized the dataset for all models, including DistilBERT. Despite the lighter architecture of DistilBERT, consisting of 6 layers and sixty-six million parameters, it achieved an accuracy of 98%, which is competitive with BERT, XLNet, and DeBERTa, which consists of twelve layers and a more significant number of parameters. This demonstrates that DistilBERT can achieve high performance while maintaining a more efficient and streamlined architecture.

TABLE X
COMPARISON TABLE BETWEEN MODELS USED IN PREVIOUS WORKS AND THIS WORK

Models	dataset size (%)	Accuracy (%)	Key properties
BERT [20]	100%	97%	BERT base version contains 12 layers and 110 million parameters
XLNet [20]	100%	97%	XLNet base version contains 12 layers and 110 million parameters
DeBERTa [20]	100%	98%	DeBERTa base version contains 12 layers with 86 million parameters
DistilBERT (this work)	100%	98%	DistiLBERT base uncased version contains six layers with 66 million parameters

IV. CONCLUSION

In conclusion, it is essential to identify fake reviews in the e-commerce platform as they may provide buyers with discomfort or unpleasant experiences. In this work, machine learning models were experimented with different word embeddings and a transformer model was used to determine its effectiveness in detecting fake reviews. The machine learning models used were Support Vector Machine (SVM), Logistic Regression, and Random Forest, as they produced high accuracy in previous works. Similarly, Distil BERT was chosen for the same reason as transformer models. One of these research aims is to compare the effectiveness of machine learning and transformer models in detecting fake reviews. Therefore, upon comparing the results of machine learning and transformer models, it can be inferred that DistilBERT outperformed in detecting fake reviews SVM, Logistic Regression, and Random Forest with an accuracy of 98%. On the other hand, this work also found that overall, Distil BERT can perform well and produce comparable results even when it has a lighter architecture. Future works include the development of a robust fake review detection model capable of accurately classifying fake or original reviews on real-time scrapped Amazon data. Additionally, the integration of sentiment analysis to effectively identify fraudulent sellers based on sentiment polarity scores of reviews (Ziming et al., 2020). Besides, deep learning models can also be tested to observe their accuracy in detecting fake reviews. Working with deep neural network models with various architectures is also possible.

ACKNOWLEDGMENT

We extend our heartfelt thanks to CITIC2024 for their invaluable contributions and steadfast support. Your efforts have greatly influenced our initiatives, and we are deeply grateful for everything you have accomplished.

REFERENCES

[1] Z. Li *et al.*, "What happens behind the scene? Towards fraud community detection in e-commerce from online to offline," in *Companion Proc. Web Conf.*, 2021, doi: 10.1145/3442442.3451147.

[2] H. Qayyum, F. Ali, M. Nawaz, and T. Nazir, "FRD-LSTM: A novel technique for fake reviews detection using DCWR with the Bi-LSTM method," *Multimed. Tools Appl.*, pp. 1–15, 2023, doi: 10.1007/s11042-023-15098-2.

[3] L. Chen and L. Qiang, "How does fake review influence e-commerce platform revenue?," in *Proc. 23rd Asia-Pacific Netw. Oper. Manag. Symp. (APNOMS)*, 2022, doi: 10.23919/apnoms56106.2022.9919902.

[4] S. Panda and S. Levitan, "Deception detection within and across domains: Identifying and understanding the performance gap," *ACM J. Data Inf. Qual.*, vol. 15, no. 1, pp. 1–27, 2022, doi:10.1145/3561413.

[5] J. Salminen, C. Kandpal, A. M. Kamel, S. G. Jung, and B. J. Jansen, "Creating and detecting fake reviews of online products," *J. Retail. Consum. Serv.*, vol. 64, p. 102771, 2022, doi:10.1016/j.jretconser.2021.102771.

[6] P. M. Kumar, S. S. Harrsha, K. Abhiram, M. Kavitha, and M. Kalyani, "Role of machine learning in fake review detection," in *Proc. 6th Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, 2022, doi:10.1109/iceca55336.2022.10009174.

[7] U. R. I. and E. P. Naseem, "A survey of pre-processing techniques to improve short-text quality: A case study on hate speech detection on Twitter," *Multimed. Tools Appl.*, vol. 80, pp. 35239–35266, Nov. 2021, doi: 10.1007/s11042-020-10082-6.

[8] J. Hauschild, "Examining the effect of word embeddings and preprocessing methods on fake news detection," Ph.D. dissertation, Univ. of Nebraska, 2023. [Online]. Available: https://digitalcommons.unl.edu/dissertations/AAI30487005/

[9] J. Bhopale, R. Bhise, A. Mane, and K. Talele, "A review-and-reviewer based approach for fake review detection," in *Proc. 4th Int. Conf. Electr., Comput. Commun. Technol. (ICECCT)*, 2021, doi:10.1109/icecct52121.2021.9616697.

[10] C. Sasikala *et al.*, "Fake review detection and classification using improved convolutional neural network on Amazon dataset," in *Proc. 3rd Int. Conf. Pervasive Comput. Social Netw. (ICPCSN)*, 2023, doi:10.1109/icpcsn58827.2023.00071.

[11] D. Baishya, J. J. Deka, G. Dey, and P. K. Singh, "SAFER: Sentiment analysis-based fake review detection in e-commerce using deep learning," *SN Comput. Sci.*, vol. 2, no. 6, p. 479, Nov. 2021, doi:10.1007/s42979-021-00918-9.

[12] A. B. H. Krishnan, "Unmasking falsehoods in reviews: An exploration of NLP techniques," *arXiv preprint*, arXiv:2307.10617, Jul. 2023. [Online]. Available: https://arxiv.org/abs/2307.10617.

[13] R. Mohawesh, M. Hawawreh, and M. Alqudah, "Factitious or fact? Learning textual representations for fake online review detection," *Cluster Comput.*, pp. 1–16, Sep. 2023, doi: 10.1007/s10586-023-04148-x.

[14] P. Shetgaonkar *et al.*, "Fake review detection using sentiment analysis and deep learning," in *Proc. Int. Conf. Technol. Advancements Innov. (ICTAI)*, 2021, doi: 10.1109/ictai53825.2021.9673375.

[15] G. C. R. and W. Z. Budhi, "Resampling imbalanced data to detect fake reviews using machine learning classifiers and textual-based features," *Multimed. Tools Appl.*, vol. 80, pp. 13079–13097, Apr. 2021, doi:10.1007/s11042-020-10299-5.

[16] S. K. Maurya, D. Singh, and A. K. Maurya, "Deceptive opinion spam detection using feature reduction techniques," *Int. J. Syst. Assur. Eng. Manag.*, vol. 15, no. 3, pp. 1210–1230, Mar. 2024, doi:10.1007/s13198-023-02208-4.

[17] N. Wang, J. Yang, X. Kong, and Y. Gao, "A fake review identification framework considering the suspicion degree of reviews with time burst characteristics," *Expert Syst. Appl.*, vol. 190, p. 116207, 2022, doi:10.1016/j.eswa.2021.116207.

[18] A. K. F. and C. M. Mir, "Online fake review detection using supervised machine learning and BERT model," *arXiv preprint*, arXiv:2301.03225, Jan. 2023. [Online]. Available: https://arxiv.org/abs/2301.03225.

[19] V. C. S. Rao, P. Radhika, N. Polala, and S. Kiran, "Logistic regression versus XGBoost: Machine learning for counterfeit news detection," in *Proc. 2nd Int. Conf. Smart Technol. Comput., Electr. Electron. (ICSTCEE)*, 2021, doi: 10.1109/icstcee54422.2021.9708587.

[20] I. Choudhary, N. Tyagi, P. Taneja, and R. Bhatia, "Enhancing review authenticity using transformers: Web extension for detecting AI-generated fake reviews vs human-written feedback," in *Proc. 3rd Asian Conf. Innov. Technol. (ASIANCON)*, 2023, doi:10.1109/asiancon58793.2023.10270213.

[21] R. Catelli, H. Fujita, G. De Pietro, and M. Esposito, "Deceptive reviews and sentiment polarity: Effective link by exploiting BERT," *Expert Syst. Appl.*, 2022, doi: 10.1016/j.eswa.2022.118290.

[22] P. Gupta, S. Gandhi, and B. R. Chakravarthi, "Leveraging transfer learning techniques—BERT, RoBERTa, ALBERT and DistilBERT for fake review detection," in *Proc. 13th Annu. Meet. Forum Inf. Retrieval Eval.*, 2021, doi: 10.1145/3503162.3503169.

[23] D. Refaeli and H. Hajek, "Detecting fake online reviews using fine-tuned BERT," in *Proc. 5th Int. Conf. E-Bus. Internet*, 2021, doi:10.1145/3497701.3497714.

[24] T. C. B. and P. Lin, "A study on identification of important features for efficient detection of fake reviews," in *Proc. Int. Conf. Data Analytics for Business and Industry (ICDABI)*, 2021, doi:10.1109/icdabi53623.2021.9655845.

[25] S. M. Anas and S. Kumari, "Opinion mining based fake product review monitoring and removal system," in *Proc. 6th Int. Conf. Inventive Comput. Technol. (ICICT)*, 2021, doi:10.1109/icict50816.2021.9358716.

[26] D. K. S. and G. Y. Jain, "Fake reviews filtering system using supervised machine learning," in *Proc. IEEE Int. Conf. Data Sci. Inf. Syst. (ICDSIS)*, 2022, doi: 10.1109/icdsis55133.2022.9915878.

[27] Z. Zeng, Z. Zhou, and X. Mu, "User review helpfulness assessment based on sentiment analysis," *Electron. Library*, 2020, doi:10.1108/el-08-2019-0200.

[28] R. Shahid *et al.*, "Predicting customer sentiment in social media interactions: Analyzing Amazon help Twitter conversations using machine learning," *Int. J. Adv. Sci. Comput. Eng.*, vol. 6, no. 2, pp. 52–56, Jul. 2024, doi: 10.62527/ijasce.6.2.211.

[29] Y. Lim, K.-W. Ng, P. Naveen, and S.-C. Haw, "Emotion recognition by facial expression and voice: Review and analysis," *J. Informatics Web Eng.*, vol. 1, no. 2, pp. 45–54, Sep. 2022, doi:10.33093/jiwe.2022.1.2.4.

[30] W.-H. Khong, L.-K. Soon, and H.-N. Goh, "A comparative study of statistical and natural language processing techniques for sentiment analysis," *J. Teknol.*, vol. 77, no. 18, Nov. 2015, doi:10.11113/jt.v77.6502.