Ingénierie des Systèmes d'Information
Vol. 31, No. 2, February, 2026, pp. 401-409
Journal homepage: http://iieta.org/journals/isi

Fake Reviews Detection in Tokopedia Using Fine-Tuned BERT-Indonesia

Rendra Soekarta*, Fajar Rahardika Bahari Putra, Suhardi Aras, Muhammad Rizki Setyawan, Reinhard Ruimassa, Ahmad Ilham

Informatics Engineering Study Program, Muhammadiyah University of Sorong, Southwest Papua 98416, Indonesia

Corresponding Author Email: rsoekarta@um-sorong.ac.id

https://doi.org/10.18280/isi.310208

Received: 4 September 2025
Revised: 25 November 2025
Accepted: 18 February 2026
Available online: 28 February 2026

Keywords:
fake review detection, Tokopedia, BERT-Indonesian, e-commerce, natural language processing

ABSTRACT

With the rapid growth of e-commerce platforms such as Tokopedia, the prevalence of fake reviews has become a significant concern, undermining consumer trust and decision-making. This study proposes a solution for detecting fake reviews on Tokopedia using a fine-tuned BERT-Indonesian model. The model, specifically trained for fake review detection, was applied to a dataset consisting of 1,787 reviews. Data preprocessing, including stopword removal, tokenization, and stemming, was performed to prepare the data for training. The model's performance was evaluated using standard metrics, including accuracy, precision, recall, and F1-score, with results showing an impressive accuracy of 97%, precision of 97% in each class, and an F1 score of 98%. The proposed system offers a robust solution for detecting fake reviews, which helps improve user safety and trust on e-commerce platforms. Moreover, this study highlights the effectiveness of fine-tuning BERT-Indonesian for fake review detection in the context of Indonesia.

1. INTRODUCTION

The development of information and communication technology has driven a major transformation in various sectors of life, including in the fields of trade and consumer services. One of the real impacts of this progress is the emergence of an e-marketplace platform that can allow the buying and selling process to be carried out online without the limitations of space and time. This convenience has made people increasingly accustomed to shopping online, especially through e-marketplaces that provide a variety of products from various sellers in one integrated platform [1-3], one of which is Tokopedia. Tokopedia is an e-marketplace that sells various products from household materials, beauty, cosmetics, tools, clothing, and others [4]. The high public interest in Tokopedia has led to an increasing number of people shopping without caution. This is because many sellers use a mode of asking third parties to write fake positive reviews to increase the popularity of their stores. Quoted from the Tirto Indonesia website with the news title "Beware of Fake Reviews in E-Commerce", that existing product reviews do not really match reality. A study by the research organization Mintel showed that approximately 57% of surveyed customers believe that companies or products only have positive reviews, with no negative ones. Furthermore, approximately 49% said that companies may incentivize those who post fake reviews online. Therefore, a system that can identify genuine and fake reviews is needed, which can then be implemented on a website interface so that users can use it to determine which products to purchase.

One of the technologies or approaches that can be used to create this system is natural language processing (NLP). The field of artificial intelligence known as NLP, focuses on teaching computers to comprehend, process, and produce language. Voice assistants, machine translation services, and search engines are all supported by this technology [5-7]. At the moment, NLP is extensively incorporated into daily life through virtual assistants like Google Home, Alexa, and Siri. NLP is also crucial for companies to obtain a competitive edge in the industrial sector. Applications of NLP can be useful in many aspects of life, particularly in the analysis and value extraction of unstructured data [8]. One of the most prominent models in NLP is BERT (Bidirectional Encoder Representations from Transformers). This model links every output token to all input tokens, with connection weights computed dynamically. Traditionally, language models process text in a single direction—either from left to right or from right to left—but not both at the same time. BERT differs from these approaches because it can analyze context from both directions simultaneously. Owing to this bidirectional capability, BERT is pre-trained for various NLP tasks, particularly masked language modeling and next sentence prediction [9]. Due to BERT's popularity, many people are trying to develop variants of this model for more specific tasks, such as BERT-Indonesian. BERT-Indonesian itself was created by Cahya Wirawan and can be accessed through the open-source website hugging face. This model is an extension of the BERT-Base model pre-trained using Indonesian Wikipedia data using the masked-language-model technique. In its description on hugging face it is stated that the model used is named BERT-Base-indonesian-522M [10].

In previous research conducted by Refaeli and Hajek, which focused on identifying and detecting fake reviews using Fine-Tuned BERT using two types of datasets, namely

crowdsourced datasets such as review datasets on Amazon, hotels, and restaurants and Yelp datasets, namely datasets provided by Yelp itself. Yelp is an online review platform that allows users to provide ratings on local businesses based in the United States. The results of this study showed the best accuracy results were obtained on the crowdsourced dataset with an accuracy of 91% and on the Yelp dataset reaching 73% [11]. Furthermore, in research conducted by Alamsyah et al. [12] focused on detecting fake reviews from Tokopedia using the support vector machine (SVM) and Naive Bayes algorithms based on analyzed sentence patterns. In this study, the dataset was scraped from Tokopedia with a dataset of around 887 reviews of slimming products. The results obtained can be seen in the performance of the SVM algorithm which achieved 94% accuracy by classifying 29 new unlabeled reviews into 8 original reviews and 21 fake reviews (Computer Generated) [12]. It is hoped that the results of this study can help people in choosing good and good products when shopping online and can also provide a Fine-Tuned BERT-Indonesian performance that delivers optimal outcomes with a high level of accuracy.

2. METHODOLOGY

2.1 Research flow

This research was conducted through several stages of a systematic research process. The initial stage began with problem identification to understand the issues raised, followed by a literature review to obtain a theoretical basis and related research. Next, data collection was carried out as the main material for the research, which was then used in the model development stage. The developed model was then implemented into a website-based system using the Extreme Programming method. The final stage of the research included drawing conclusions and providing suggestions for further research development. The entire research process is illustrated in Figure 1.

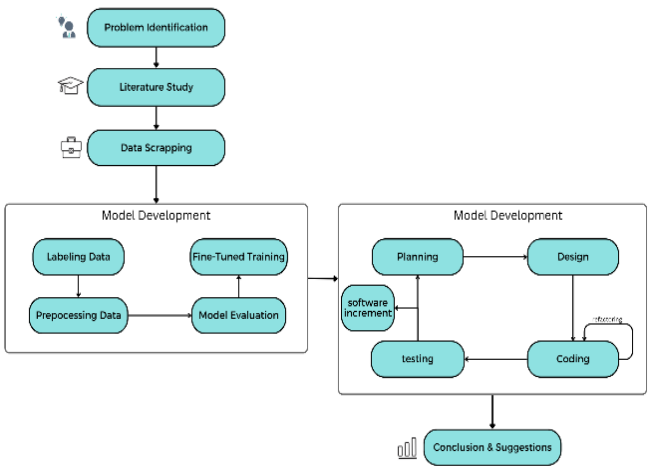

Figure 1. Research flow

The first step was to identify the problem to be addressed, namely the large number of sellers who use the method of asking third parties to write fake positive reviews to increase their store's popularity. This was done to become the main focus of the research. Next, the author searched for previous research such as articles, journals, and theses related to fake review classification, NLP, and BERT. After that, the author began collecting data in the form of review data from three stores on the website www.tokopedia.com and saved it in CSV format on Google Drive. This data will then be trained using a fine-tuned BERT-Indonesian model for a more specific task: fake review classification.

2.2 Requirement analysis

1. Hardware Requirements
   a. *Laptop Acer Aspire A314-36M*
   b. *Intel® Core ™ i5_N305*
   c. *RAM 8GB*
   d. *SSD Kingston M2 NVMe 500GB*
2. Software Requirements
   a. *Windows 11*
   b. *Visual Studio Code*
   c. *Chrome*
   d. *Streamlit*
   e. *Google Colab*
   f. Draw.io
3. Dataset Requirements
   a. 1,787 Unlabeled raw dataset in csv format

2.3 Model development

The model development form in this study focuses on the use of a fine-tuned Indonesian BERT model for a specific task, namely detecting fake reviews on the Tokopedia platform. The initial stage began with the collection of user review data, which was then annotated and labeled as genuine or fake reviews. This data includes various variations in language style, text length, and writing context to represent diverse conditions in the real world. The dataset that was successfully compiled contained 1,787 data points. Next, the collected data will be reprocessed to be labeled using Latent Dirichlet Allocation (LDA) to distinguish between genuine and fake reviews. Next, the data will be preprocessed using two stages, namely stopwords, tokenize, and stemming. After labeling, the results obtained were 913 genuine review datasets and 874 fake review datasets. The dataset was then divided into two parts, namely 80% training data and 20% validation data. After that, the data will be arranged using new parameters and then trained to become a model specifically for the task of detecting fake reviews. After the data is trained, the model will be evaluated using a confusion matrix to calculate accuracy, precision, recall, and F1-score. Finally, the model is saved with a pkl extension, ready to be implemented in a website-based system to detect fake reviews.

2.4 System development

Development of this system will follow the System Development Life Cycle (SDLC) using the Extreme Programming (XP) methodology, which is a key part of our process. This approach is divided into four main, iterative stages. The first stage, Planning, focuses on defining the project's foundation. Here, we identify core system requirements, including the purpose, target audience, development timeline, and estimated costs. The second stage, Design, translates these requirements into visual blueprints. We use flowcharts to map the program's logical flow, use case diagrams to visualize how users will interact with the system, and user interface (UI) mockups to create a preliminary visual

representation of the system's look and feel. These detailed plans guide the subsequent phases. After design, the Coding stage begins, where developers write and integrate code based on the approved architecture. The final stage, Testing, is a continuous process throughout development, not just at the end. We perform rigorous tests to ensure the system is robust, reliable, and meets all specified requirements, ensuring a high-quality final product that aligns with user needs.

In Figure 2, the system begins with the user entering a product URL from Tokopedia. The user then presses the "Start Detection" button, and the system processes the results. During the process, the system will retrieve reviews directly, and the model will detect the retrieved reviews. The system will then display the results in the form of a visualization and the percentage of detected reviews.

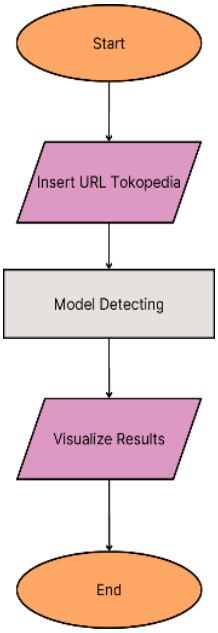

Figure 2. Flowchart

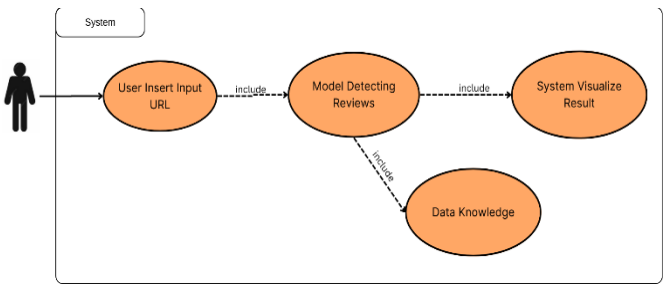

Figure 3. Use case diagram

In Figure 3, the user enters a URL, which is then processed by the system, and the model detects the retrieved reviews. Then, data knowledge is input in the form of a model trained for the review detection function. Finally, the system displays the review detection results.

The third stage is Coding. In this phase, this research implements application design according to the previously developed system design. The system interface was created using the Python programming language, specifically the Streamlit framework. The system was developed using Visual Studio Code as a text editor for writing the implementation code. Finally, the fourth stage is Testing. In this stage, the system created in the previous stage will be tested for feasibility before being launched to the public. This feasibility test uses blackbox testing to verify that features should function properly. Next, there is Usability Testing. This test is used to sample several people to use the system. This testing aims to determine the level of satisfaction of these participants.

If errors are found, evaluation and corrections are made to the relevant modules. The implementation of this system is expected to be a tool for effectively detecting fake reviews and increasing consumer confidence in online shopping.

3. RESULTS AND DISCUSSION

3.1 Model development results

3.1.1 Dataset prepare

The dataset scraping from the official Tokopedia website uses the Python programming language. The tools and libraries used include Beautiful Soap (BS4) and Chrome Driver. The data collected comprises 1,787 items with three columns: user name, review, and star rating, taken from three stores: a clothing store, a sandal store, and a beverage store. The following is an example of the collected dataset. An example of the collected dataset is presented in Table 1. This data will be stored in Google Drive and processed using Google Colab.

Table 1. Dataset

Username	Review	Stars Rating
eddy	Bahan halus tapi tidak terlalu tipis, jahitan OK. Seller ramah. Recommended	5
A***s	mantap jiwa pengirimannya cukup lama, statusnya sedang otw ke destinasi tapi 8 jam baru smpe rumah	5
Meda	Alhamdulillah barangnya sampai, worth it lah sesuai dengan harganya.terima kasih	5
M***s	ukirannya pas dikaki tp bahannya agak licin tipis mudahmudahan awet	4
Adi	barang sudah sampe terimakasih buat toko nya sayang nya tipis	4
C****i	Recommended sekali bahanya ##	5

3.1.2 Labeling data

At this stage, the process of assigning labels to data is carried out using the LDA algorithm, which is a topic modeling method widely used in text analysis [13, 14]. LDA works by identifying hidden patterns in a collection of documents and grouping them into a number of specific topics based on the distribution of words that appear. In the context of this research, LDA is used to group reviews obtained from Tokopedia into two main categories, namely real reviews and fake reviews. This process is carried out indirectly through probabilistic analysis, where each review is analyzed to determine its possible association with each topic. LDA calculates the probability distribution of each word in a review and associates the review with the topic that has the dominant probability. To ensure the validity of the dataset, a manual verification process was conducted on a sample of reviews from each generated topic. Based on this manual review, it was confirmed that Topic 0 is dominated by linguistic characteristics typical of genuine reviews (e.g., specific context, natural sentence structure). Consequently, Topic 0 is mapped to the “Real” label (encoded as 1). Conversely, Topic

1 primarily contains linguistic patterns indicating fake reviews (e.g., repetitive templates, overly general or excessive praise), and is mapped to the “Fake” label (encoded as 0). This mapping logic is implemented in the system by evaluating the dominant_topic condition. The use of LDA at this stage not only accelerates the semi-automated data labeling process but also helps identify linguistic patterns commonly found in both fake and genuine reviews, thereby providing a stronger foundation for training the classification model in the subsequent stage.

```
vectorizer = TfidfVectorizer(stop_words='english', lowercase=True)
X_tfidf = vectorizer.fit_transform(df['Ulasan'])
corpus = gensim.matutils.Sparse2Corpus(X_tfidf, documents_columns=False)
dictionary = gensim.corpora.Dictionary.from_corpus(corpus, id2word=dict(enumerate(vectorizer.get_feature_names_out())))
lda_model = gensim.models.LdaModel(corpus=corpus,
                                   id2word=dictionary,
                                   num_topics=2,
                                   random_state=42,
                                   passes=10)
print("\nTopik yang ditemukan:")
for idx, topic in lda_model.print_topics(num_words=5):
    print(f"Topik {idx+1}: {topic}")
def get_dominant_topic(bow):
    topics = lda_model.get_document_topics(bow)
    dominant_topic = max(topics, key=lambda x: x[1])[0]
    if dominant_topic == 0:
        return "Real", 1
    else:
        return "Fake", 0
df[['topik', 'label']] = pd.DataFrame([get_dominant_topic(bow) for bow in corpus], index=df.index)
df.to_csv('/content/drive/MyDrive/skripsi/test-1/ulasan_dengan_label_topik.csv', index=False)
```

Figure 4. Labeled data

Based on the implementation of the LDA algorithm shown in Figure 4, the following is the result of the dataset labeled using LDA, as shown in Table 2.

Table 2. Labeled dataset

Review	Stars Rating	Topic	Label
Bahan halus tapi tidak terlalu tipis, jahitan OK. Seller ramah. Recommended	5	Real	1
mantap jiwa pengirimannya cukup lama, statusnya sedang otw ke destinasi tapi 8 jam baru smpe rumah	5	Real	1
Alhamdulillah barangnya sampai, worth it lah sesuai dengan harganya.terima kasih	5	Real	1
ukirannya pas dikaki tp bahannya agak licin tipis mudahmudahan awet	4	Fake	0
barang sudah sampe terimakasih buat toko nya sayang nya tipis	4	Fake	0
Recommended sekali bahanya ##	5	Fake	0

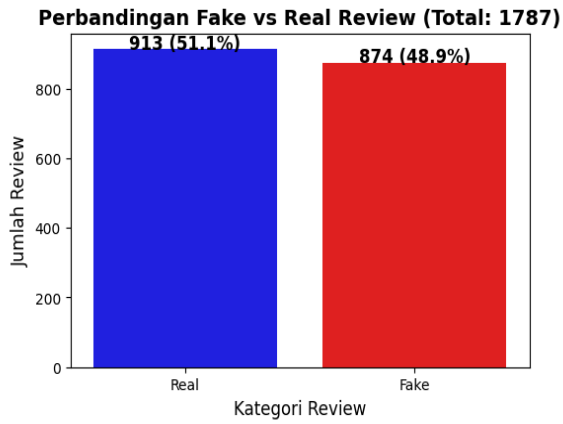

Figure 5. Comparison labeled dataset

Based on Figure 5, "Comparison Fake vs Real Review (Total: 1,787)," this graph presents a comparison between the number of real and fake reviews out of a total of 1,787 analyzed reviews. The real reviews are represented by the blue bar, while the fake reviews are shown by the red bar. Out of the total, 913 reviews, or 51.1%, were real, whereas 874 reviews, or 48.9%, were identified as fake. Overall, this comparison indicates that the number of real reviews is slightly greater than the number of fake reviews in the available data.

3.1.3 Prepocessing data

In this stage, the author uses three data preprocessing functions: stopwords, tokenization, and stemming, which will be applied to previously labeled data. Stopwords are used to remove meaningless data. Tokenization is used to break down sentences into phrases or words. Finally, stemming is used to remove affixes or suffixes from the broken down phrases. In this stage, the author applies three main techniques in the data preprocessing process: stopword removal, tokenization, and stemming, all of which are performed after the data labeling process is complete. These preprocessing stages aim to clean and prepare the text data to make it more structured and relevant before being used in the model training process. First, stopwords are removed, which are common words in Indonesian that frequently appear but have no significant meaning in the context of the analysis, such as "yang," "dan," "atau," and so on. Removing these words is important to reduce noise in the data and help the model focus on more meaningful words [15, 16]. Can see Figure 6 shows an example of the stopwords removal process.

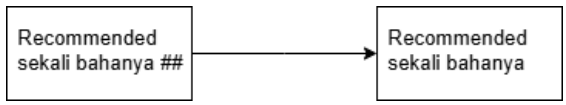

Figure 6. Stopwords removal

For code implementation of stopwords removal can be seen in Figure 7.

```
stop_words = StopWordRemoverFactory().get_stop_words()
new_array = ArrayDictionary(stop_words)
stop_words_remover_new = StopWordRemover(new_array)

def stopword(str_text):
  str_text = str(str_text)
  str_text = stop_words_remover_new.remove(str_text)
  return str_text

data['Ulasan'] = data['Ulasan'].apply(lambda x: stopword(x))
data.head()
```

	Ulasan	Rating Bintang	topik	label
0	bahan halus tidak terlalu tipis jahitan seller...	5	Real	1
1	mantap jiwa pengirimannya cukup lama statusnya...	5	Real	1
2	alhamdulillah barangnya worth it lah sesuai ha...	5	Real	1
3	bahan halus ga terlalu tebal enak dipake recom...	5	Real	1
4	barang diterima baik sesuai iklan packing eapi...	5	Real	1

Figure 7. Implementation code stopwords

Figure 8. Tokenize

Next, tokenization is performed, which is the process of breaking down sentences into smaller units, namely words or

phrases [17, 18]. By breaking sentences into tokens, the system can analyze language structures and patterns more accurately. Figure 8 shows an example of the tokenize process.

For code implementation of tokenize can be seen in Figure 9.

```
tokenized = data['Ulasan'].apply(lambda x:x.split())
tokenized
```

	Ulasan
0	[bahan, halus, tidak, terlalu, tipis, jahitan,...
1	[mantap, jiwa, pengirimannya, cukup, lama, sta...
2	[alhamdulillah, barangnya, worth, it, lah, ses...
3	[bahan, halus, ga, terlalu, tebal, enak, dipak...
4	[barang, diterima, baik, sesuai, iklan, packin...
...	...
1782	[seller, fast, respond, kemasan, sangat, aman,...
1783	[produk, dibuat, bahan, berkualitas, rasa, enak]
1784	[packingnya, mantap, produk, mendarat, selamat...
1785	[bagus, bahan, bahannya, asli, terasa, bawang,...
1786	[beberapa, hari, cobain, lom, keliatan, manfaa...

1787 rows × 1 columns

Figure 9. Implementation code Tokenize

The final step is stemming, which is the process of converting a tokenized word into its root form by removing any affixes attached to the word, such as the prefixes "ber-", "me-", "di-", or the suffixes "-kan", "-an" [19, 20]. Stemming is important so that words with the same meaning but different forms can be recognized as a single entity by the model. Figure 8 shows an example of the stemming process. Figure 10 shows an example of stemming process.

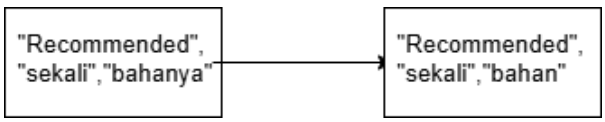

Figure 10. Stemming

For code implementation of stemming can be seen in Figure 11.

```
def stemming(Ulasan):
  factory = StemmerFactory()
  stemmer = factory.create_stemmer()
  do = []
  for w in Ulasan:
    dt = stemmer.stem(w)
    do.append(dt)
  d_clean = []
  d_clean = " ".join(do)
  print(d_clean)
  return d_clean

tokenized = tokenized.apply(stemming)

tokenized.to_csv('/content/drive/MyDrive/skripsi/test-1/label_bersih', index=False)
data_clean = pd.read_csv('/content/drive/MyDrive/skripsi/test-1/label_bersih')
data_clean.head()

bahan halus tidak terlalu tipis jahit seller ramah recommended
mantap jiwa kirim cukup lama status sedang otw destinasi 8 jam baru smpe rumah
alhamdulillah barang worth it lah sesuai harga terima kasih
bahan halus ga terlalu tebal enak dipake recommended
barang terima baik sesuai iklan packing eapi aman seller aktif chat cepat proses kirim kaos kaki pas kaki yang ukur sepatu 44 bahan 1
warna sesuai request bahan material nyaman pakai kaki ukur oke kualitas soft bagus moga awet kaos kaki thanks
sesuai deskripsi walaupun bahan tebal nyaman pakai moga tidak mudah kendor
nyaman tipis gak gerah gatau nih tahan lama ga kaos kaki belum tipis2 jebol bagi tumit
tipis pendekbahan halusnyaman dikajitidak panas
barang bagus sesuai iklan halus nyaman pakai kirim cepat
model sesuai gambar bahan ringan nyaman moga awet cocok kaki ukur 42
bahan nyaman terlalu tebal terlalu tipis warna hitam abu tua hampir tdk lihat beda
```

Figure 11. Implementation code stemming

3.1.4 Training model

At this stage, the data that has been preprocessed will be split into a ratio of 8:2 for training and validation, with parameters such as Max Length 128, Batch Size 16, AdamW optimizer, Learning Rate 2e-5, and 3 epochs. The configuration and training process are illustrated in Figure 12 and Figure 13.

```
tokenizer = BertTokenizer.from_pretrained("cahya/bert-base-indonesian-522M")
model = BertForSequenceClassification.from_pretrained("cahya/bert-base-indonesian-522M", num_labels=2)

/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
tokenizer_config.json: 100%
vocab.txt: 100%
special_tokens_map.json: 100%
config.json: 100%
pytorch_model.bin: 100%
model.safetensors: 100%
Some weights of BertForSequenceClassification were not initialized from the model checkpoint at cahya/bert-base-indonesian-522M and are newly initialized: ['classifier.bias', 'classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

Figure 12. Set and config model

```
MAX_LEN = 128
BATCH_SIZE = 32

dataset = CustomDataset(result, tokenizer, MAX_LEN)
train_size = 0.8
train_dataset, val_dataset = torch.utils.data.random_split(dataset, [int(train_size*len(dataset)), len(dataset) - int(train_size*len(dataset))])

train_dataloader = DataLoader(train_dataset, batch_size=BATCH_SIZE, shuffle=True)
val_dataloader = DataLoader(val_dataset, batch_size=BATCH_SIZE)

optimizer = AdamW(model.parameters(), lr=2e-5)
num_epochs = 3
num_training_steps = num_epochs * len(train_dataloader)
lr_scheduler = get_scheduler(
    "linear",
    optimizer=optimizer,
    num_warmup_steps=0,
    num_training_steps=num_training_steps
)
```

Figure 13. Training model

3.1.5 Model evaluation

Once the testing phase is conducted using evaluation data, the next step is to evaluate the model's performance. This evaluation is done by utilizing a confusion matrix, as shown in Figure 14. A confusion matrix is a tabular representation that shows the comparison between the model's predictions and the actual labels. From the results obtained, the model was able to correctly identify 304 real data as real and 44 fake data as fake. However, there was 1 real data point that was incorrectly predicted as fake and 9 fake data points that were incorrectly predicted as real. The misclassification of these 10 reviews may be attributed to ambiguous language usage, excessively short reviews lacking sufficient context, or the use of informal words that disrupted the model's semantic interpretation.

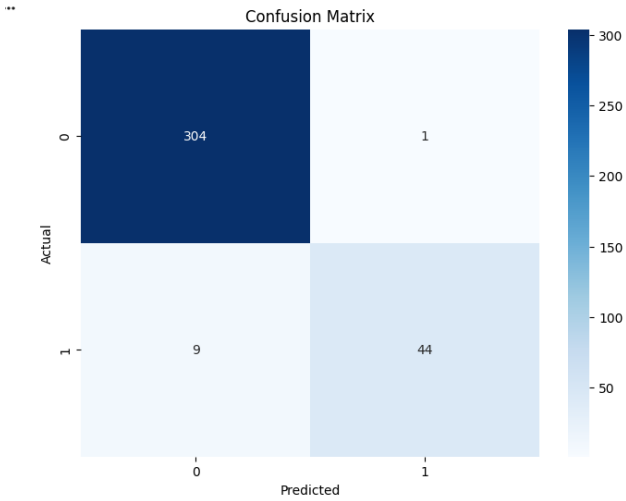

Figure 14. Results of confusion matrix

Next, the model evaluation results are continued using the classification report shown in Figure 15. This report presents key metrics such as precision, recall, and f1-score for each class, as well as overall accuracy. Based on the obtained results, the real class (0) achieved a precision of 0.99, a recall

of 0.97, and an F1-score of 0.98 from 175 data samples. In comparison, the fake class (1) recorded a precision of 0.97, a recall of 0.99, and an F1-score of 0.98 across 183 data samples. Overall, the model reached an accuracy of 0.98, with both the macro average and weighted average also scoring 0.98. This shows that the model has excellent and balanced performance in detecting both classes, real and fake. This evaluation is based on the amount of evaluation data that has been divided into 358 data points.

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.97	0.98	175
1	0.97	0.99	0.98	183
accuracy			0.98	358
macro avg	0.98	0.98	0.98	358
weighted avg	0.98	0.98	0.98	358

Figure 15. Classification reports

3.1.6 Save model

At this stage, the model will be saved with a pkl extension, which will later be used on the web. This file will serve as a reference for detecting genuine and fake reviews on the web-based system. For more details, see Figure 16.

Figure 16. Saved model

3.2 Implementation result

The system implementation phase was conducted using the Python programming language and leveraged the Streamlit framework to build an interactive and easy-to-use web-based user interface. The entire system development process was conducted locally using Visual Studio Code (VSCode) software, which facilitates efficient code writing and live testing. The system implementation consists of two main code files: app.py, the core application component, and one or more supporting files that handle specific functions such as model processing and data management. During system development, several critical functions were developed to ensure seamless integration between the user interface and the fake review detection model. These functions include: a model load function, which initializes the trained BERT model; a function for directly retrieving review data, either through manual input or external sources; and a classification function to determine whether a review is genuine or fake. Furthermore, the system is equipped with a results visualization function, which presents the model's prediction output in an easy-to-understand graphical display; and a page navigation function to allow users to navigate between application sections, such as the main page, user guide, and detection results. The system interface is shown in Figures 17 and 18.

Figure 17. Detect view

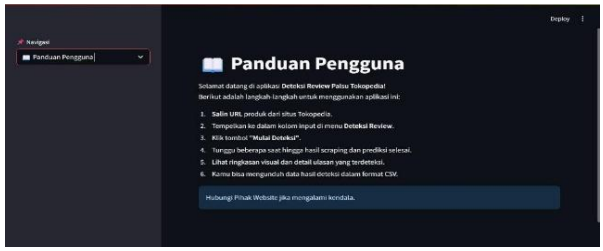

Figure 18. User guide

3.2.1 Blackbox testing

At this stage, the system is tested to ensure all the functions created work as intended. The test Table 3 shows the system.

Table 3. Blackbox testing

Test Case	Scenarios Tested	Expected Results	Test Results
Access the home page url	User accesses url and displays home page for detection	Success with the system can display the initial detection page	The system successfully displayed the main detection page consistently.
Trying the Detection Feature	Users enter product URL links from Tokopedia	Success in getting reviews from the Tokopedia page directly and displaying prediction results along with visualizations of the results obtained	The system successfully displays predictions and visualizations of results.
Trying Out the User Guide Feature	Users can log in and the system will display a guide on how to use the website	Success with the user can successfully enter the user guide page and the system displays the user guide	The system successfully displayed the user guide page.
Trying Out the Page Navigation Feature	Users can move between pages and the system will display the desired page directly	Success with users can successfully move pages directly and the system also displays the page directly	The system successfully navigates between pages smoothly.

3.2.2 Usability testing

To assess user satisfaction with the developed service or product, researchers developed a simple survey using the Google Forms platform. The survey instrument consisted of five questions designed to directly elicit user views and responses regarding the system used. This method was chosen because it efficiently collected feedback from a large number of respondents and presented the data in a structured and easily analyzed format. Based on the processed data from the survey responses, an interpretive score of over 87.4% was obtained, categorized as "Strongly Agree." These results were obtained from a questionnaire of 25 respondents, consisting of 11 women and 14 men. Data were collected using a Google Form, with the following percentages obtained.

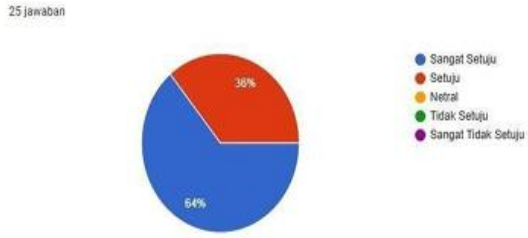

Figure 19. Diagram question 1

Based on Figure 19, the statement "I feel that product reviews on Tokopedia greatly influence my decision to buy a product." received 25 responses, including 64% or 16 people who chose Strongly Agree, and 36% or 9 people who chose Agree.

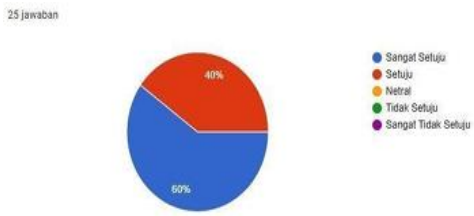

Figure 20. Diagram question 2

Based on Figure 20, the statement "Fake reviews on Tokopedia can affect the quality of my shopping experience." received 25 responses, including 60% or 15 people who chose Strongly Agree, and 40% or 10 people who chose Agree.

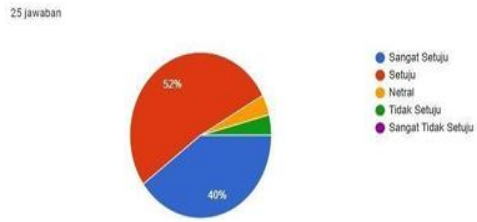

Figure 21. Diagram question 3

Based on Figure 21, the statement "I will feel more confident buying a product if the product reviews have been checked using a fake review detection system" received 25 responses, including 40% or 10 people who chose Strongly Agree, 52% or 13 people who chose Agree, 4% or 1 person who chose Neutral, and 4% or 1 person who answered Disagree.

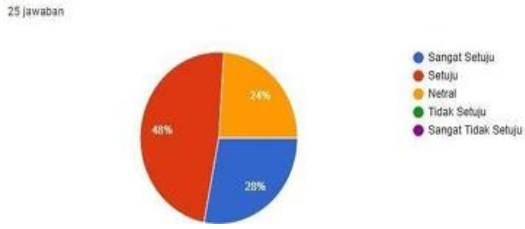

Figure 22. Diagram question 4

Based on Figure 22 above, the statement "I would buy products on Tokopedia more often if I knew that product reviews had been filtered to remove fake reviews," received 25 responses, including 28% (7 people) who strongly agreed, 48% (12 people) who agreed, and 24% (6 people) who answered neutral.

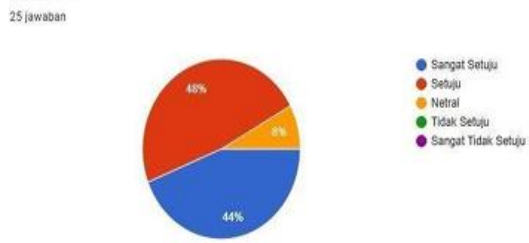

Figure 23. Diagram question 5

Figure 23 above, regarding the statement "I agree that technology like Fine-Tuned BERT-Indonesia can identify inauthentic reviews on e-marketplace platforms like Tokopedia," received 25 responses, of which 44% (11 people) strongly agreed, 48% (12 people) agreed, and 8% (2 people) answered neutral. Based on the questionnaire results above, the results were then calculated using a Likert scale (1-5) to measure respondents. The Likert scale score table can be seen in Table 4.

Table 4. Scala Likert

Answer Scale	Description	Score
SS	Strongly Agree	5
S	Agree	4
N	Neutral	3
TS	Don't Agree	2
STS	Strongly Disagree	1

Based on the questionnaire score data in Table 4, the percentage of each answer can be calculated using the following formula.

$$Y = \frac{x}{skor\ ideal} X \quad (1)$$

Based on the formula above, we can calculate each question on the Google form as follows. The first question, I feel that product reviews on Tokopedia really influence my decision to buy a product, getting a percentage of 92%, the second question, fake reviews on Tokopedia can affect the quality of my shopping experience, I got a percentage of 92%, the third question, I feel more confident buying a product if the product reviews have been checked using a fake review detection

system, got a percentage of 86%, the fourth question, I would buy products on Tokopedia more often if I knew the reason why products have been filtered to eliminate fake reviews getting a percentage of 80%, and last question, I agree that technology like fine-tuned BERT Indonesian can identify inauthentic reviews from e-commerce platforms like Tokopedia getting a percentage of 87%. if calculated it will produce a percentage result of 87.4%.

$$Y = \frac{92+92+86+80+87}{5} = \frac{437}{5} = 87.4 \quad (2)$$

This indicates that the majority of users were satisfied and responded positively to the fake review detection system. Therefore, it can be inferred that the developed system has met user expectations.

4. CONCLUSION

Based on the results of research conducted to detect fake reviews for products on Tokopedia using a fine-tuned BERT-Indonesian model, the following conclusions can be drawn:

1. Based on the results of this study, the fine-tuned BERT-Indonesian model, retrained to update weights such as maximum length, batch size, 8:2 data split, optimizers, and learning rate, and the new dataset labeled using LDA techniques for the fake review classification task, achieved 98% overall accuracy across a total of 1,787 datasets. This demonstrates the model's ability to effectively classify the tasks in this study.

2. Based on the results of this study, a website-based system that can predict whether a review on Tokopedia is genuine or fake was successfully designed and developed using the pre-trained fine-tuned BERT-Indonesian model. This system is designed to be user-friendly and deliver effective results.

Furthermore, the results of this study demonstrate a significant improvement in performance compared to previous research. In the study conducted by Refaeli and Hajek [11], the fine-tuned BERT model achieved an accuracy of 91% on crowdsourced datasets. Meanwhile, research by Alamsyah et al. [12] using the SVM algorithm reported a best accuracy of 94%. In contrast, this study employs a fine-tuned BERT-Indonesian model with optimized training parameters and an enhanced dataset labeled using the LDA method, resulting in a higher accuracy of 98%. This improvement highlights the importance of utilizing language-specific models, as well as effective data labeling and parameter optimization, in enhancing the performance of fake review detection systems. Therefore, this study can be considered to provide superior results compared to previous approaches, particularly in terms of accuracy.

REFERENCES

[1] Reinartz, W., Wiegand, N., Imschloss, M. (2019). The impact of digital transformation on the retailing value chain. International Journal of Research in Marketing, 36(3): 350-366. https://doi.org/10.1016/j.ijresmar.2018.12.002

[2] Zou, T., Cheshmehzangi, A. (2022). ICT adoption and booming e-commerce usage in the COVID-19 era. Frontiers in Psychology, 13: 916843. https://doi.org/10.3389/fpsyg.2022.916843

[3] Figueiredo, N., Ferreira, B.M., Abrantes, J.L., Martinez, L.F. (2025). The role of digital marketing in online shopping: A bibliometric analysis for decoding consumer behavior. Journal of Theoretical and Applied Electronic Commerce Research, 20(1): 25. https://doi.org/10.3390/jtaer20010025

[4] Liana, C., Haniza, N. (2022). The effect of Tokopedia marketplace promotion on consumer shopping behavior in the COVID-19 Pandemic: Case study on Tokopedia marketplace users at Pesona Depok estate housing RT 002 RW 022. International Journal of Social Sciences, 5(4): 256-261. https://doi.org/10.21744/ijss.v5n4.1978

[5] Raiaan, M.A.K., Mukta, M.S.H., Fatema, K., Fahad, N.M., Sakib, S., Mim, M.M.J., Ahmad, J., Ali, M.E., Azam, S. (2024). A review on large language models: Architectures, applications, taxonomies, open issues and challenges. IEEE Access, 12: 26839-26874. https://doi.org/10.1109/ACCESS.2024.3365742

[6] Onan, A., Toçoğlu, M.A. (2021). A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification. IEEE Access, 9: 7701-7722. https://doi.org/10.1109/ACCESS.2021.3049734

[7] Ofer, D., Brandes, N., Linial, M. (2021). The language of proteins: NLP, machine learning & protein sequences. Computational and Structural Biotechnology Journal, 19: 1750-1758. https://doi.org/10.1016/j.csbj.2021.03.022

[8] Rumaisa, F., Puspitarani, Y., Rosita, A., Zakiah, A., Violina, S. (2021). Penerapan Natural Language Processing (NLP) di bidang pendidikan. Jurnal Inovasi Masyarakat, 1(3): 232-235. https://doi.org/10.33197/jim.vol1.iss3.2021.799

[9] Cahyawijaya, S., Winata, G.I., Wilie, B., Vincentio, K., Li, X., Kuncoro, A., Ruder, S., Lim, Z.Y., Bahar, S., Khodra, M., Purwarianti, A., Fung, P. (2021). IndoNLG: Benchmark and resources for evaluating Indonesian natural language generation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Dominican Republic, pp. 8875-8898. https://doi.org/10.18653/v1/2021.emnlp-main.699

[10] Imron, S., Setiawan, E.I., Santoso, J. (2023). Deteksi aspek review e-commerce menggunakan IndoBERT embedding dan CNN. INSYST: Journal of Intelligent System and Computation, 5(1): 10-16. https://doi.org/10.52985/insyst.v5i1.267

[11] Refaeli, D., Hajek, P. (2021). Detecting fake online reviews using fine-tuned BERT. In Proceedings of the 2021 5th International Conference on E-Business and Internet, Singapore, pp. 76-80. https://doi.org/10.1145/3497701.3497714

[12] Alamsyah, H., Cahyana, Y., Pratama, A.R. (2023). Deteksi fake review menggunakan metode support vector machine dan naïve bayes di tokopedia. Jutisi: Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi, 12(2): 585-598. http://doi.org/10.35889/jutisi.v12i2.1222

[13] Egger, R., Yu, J. (2022). A topic modeling comparison between LDA, NMF, top2vec, and bertopic to demystify twitter posts. Frontiers in Sociology, 7: 886498. https://doi.org/10.3389/fsoc.2022.886498

[14] Chauhan, U., Shah, A. (2021). Topic modeling using latent Dirichlet allocation: A survey. ACM Computing Surveys (CSUR), 54(7): 1-35. https://doi.org/10.1145/3462478

[15] Sarica, S., Luo, J. (2021). Stopwords in technical language processing. Plos One, 16(8): e0254937. https://doi.org/10.1371/journal.pone.0254937
[16] Rajwal, S. (2024). LiHiSTO: A comprehensive list of Hindi stopwords. Multimedia Tools and Applications, 83(17): 50047-50059. https://doi.org/10.1007/s11042-023-17205-9
[17] Mehta, H. (2022). Social media hate speech detection using explainable AI (Doctoral dissertation).
[18] Zuo, Y. (2022). Tokenizing renewable energy certificates (recs)—A blockchain approach for rec issuance and trading. IEEE Access, 10: 134477-134490. https://doi.org/10.1109/ACCESS.2022.3230937
[19] Pradipta, N.Y., Soetanto, H. (2024). Sentiment classification of general election 2024 news titles on detik. com online media website using multinominal naive bayes method. Journal of Applied Science, Engineering, Technology, and Education, 6(1): 43-55. https://doi.org/10.35877/454RI.asci2754
[20] Lestandy, M., Abdurrahim, A., Syafaah, L. (2021). Analisis sentimen tweet vaksin COVID-19 menggunakan recurrent neural network dan naïve bayes. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 5(4): 802-808. https://doi.org/10.29207/resti.v5i4.3308