



Identification of clickbait news articles using SBERT and correlation matrix

Supriya¹ · Jyoti Prakash Singh¹ · Gunjan Kumar¹

Received: 25 August 2023 / Revised: 9 October 2023 / Accepted: 30 October 2023 / Published online: 25 November 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Austria, part of Springer Nature 2023

Abstract

Clickbait refers to the practice of using attention-grabbing or misleading headlines to attract readers to click on particular headlines or pieces of content. This technique often involves exaggerating claims or adding false information to attract traffic. In general, clickbait headlines are unlike standard news posts or detailed articles where headlines are highly correlated with body content. In the proposed system, the dissimilarity between the headline and detailed articles is exploited to create features from headlines and its paragraph by using sentence bidirectional encoder representation from transformer (SBERT). Since the size of the paragraph is quite large compared to the headlines, only k where $k < l$ (total sentences in a paragraph) sentences from the paragraph are selected using a correlation matrix. The extracted features from the headlines, target title and selected sentences of the paragraph are concatenated and classified using machine learning (ML) classifiers. The proposed model was tested extensively on two real-world datasets, and the results showed that it performed better than the current state-of-the-art models. In experimental results, a support vector machine (SVM) classifier with the concatenated embedding of $k = 6$ dissimilar sentences from paragraph exhibited the best performance with an accuracy of 0.84, weighted precision of 0.83, weighted recall of 0.84, and weighted F_1 -score of 0.82 surpassing the state-of-the-art model by 8.8% in terms of F_1 -score.

Keywords Clickbait · SBERT · Natural language processing (NLP) · Machine learning · Correlation matrix

1 Introduction

In the modern era of digital spread, clickbait has become a common issue on social media. Currently, many articles on the internet, like news, blogs, and social media posts, often come up with sensational headlines and catchy thumbnail images to attract the reader's attention by creating curiosity (Loewenstein 1994) and encouraging them to click on the link. However, in general, those articles lack useful content for the reader. Clickbait refers to low-quality content posted on the internet to fetch a higher number of clicks for revenue.

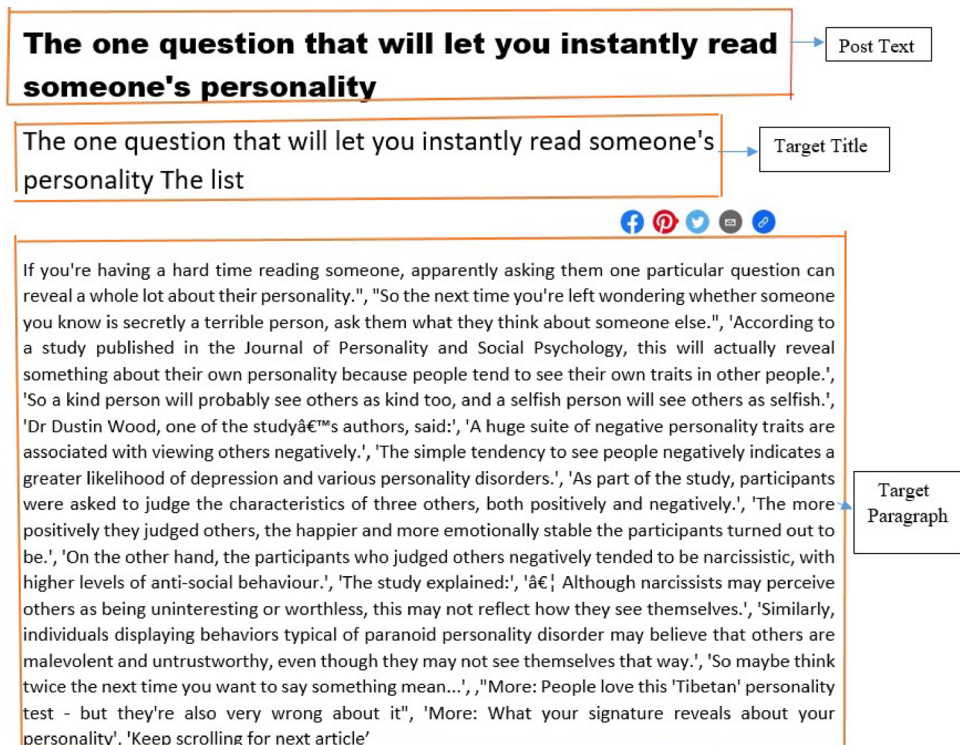
These clickbait news articles may contain text with videos, audio, and images. Generally, social media content creators often utilize various web content, such as news articles, blog posts, infographics, videos, interviews, etc., to attract attention and boost clicks with the goal of generating more revenue.

When social media users encounter this kind of post with high curiosity and find nothing after going through the articles, they end up feeling annoyed or dissatisfied with the content. It can have adverse effects on both the readers, the publishers, and the social media platforms in the long run. The click on these clickbait news articles plays with human emotions and is generally a waste of the reader's time, making them frustrated and angry. Social media website traffic would suffer if they become congested with low-quality false content, and the trust rank of the website also decreases. Some similar websites also use this concept to develop malicious links to get a click to steal personal information from readers' computers or mobiles. Consequently, it becomes crucial to develop a method for determining whether a web-posted article is clickbait or not by analyzing

✉ Jyoti Prakash Singh
jps@nitp.ac.in
Supriya
supriya.phd20.cs@nitp.ac.in
Gunjan Kumar
gunjank.phd20.cs@nitp.ac.in

¹ Computer Science and Engineering, National Institute of Technology Patna, Ashok Rajpath, Patna 800005, Bihar, India

Fig. 1 Example of clickbait news article consist of Post Text, Target Title, and Target Paragraph



relevant information such as titles, contents of posts, and other related factors.

One of the best ways to stop clickbait is to make users aware of those things and educate them to avoid clickbait: (1) The user may be informed to avoid clicking on links that lead to unfamiliar websites. (2) To avoid reading the content if the headline has a significant information gap or the image is not relevant to the headline. However, it is difficult for common people to understand and even identify clickbait headlines. The alternative and more effective solution is to develop a method to identify the clickbait headlines automatically by using the contents of the posts. Usually, a clickbait contains catchy headlines (Post Text) which, if clicked, takes the users to a page where the user will get another title (Target Title) and body content (Target Paragraph) as shown in Fig. 1. In the rest of the article, we will be referring to the catchy text shown to the user on the first screen as headlines and the target titles and body content on the page where the user lands after clicking will be referred to as title and body content, respectively.

Chakraborty et al. (2016) first developed a clickbait blocker browser extension named "Stop Clickbait" that detects and blocks clickbait from appearing while reading online news, only considering the headlines. Usually, clickbait titles include exaggerated language or have a unique style. Therefore, the most straightforward method to identify clickbait is to analyze and extract features from headlines. To achieve clickbait identification, numerous researchers (Chakraborty et al. 2016; Potthast et al. 2016) have manually

extracted linguistic features from headlines. In contrast, the only headline is insufficient to identify the clickbait because the clickbait has some huge text articles or videos that do not have relative information according to the title. So, considering the detrimental effects and unavailability of good quality datasets, a clickbait challenge was organized by the Webis clickbait challenge in 2017.¹ Zhou (2017) achieved the first rank in this challenge using a bidirectional gated recurrent unit (BiGRU) and self-attention mechanism.

Dong et al. (2019) considered the similarity between headlines and body content as the only predictive feature, as opposed to scientists in studies (Potthast et al. 2016; Zheng et al. 2021) who concentrated simply on the level of attractiveness in headlines. For example, Potthast et al. (2016) focused on the grammatical characteristics of headlines and divided them into three categories: misleading content, web page references, and data metadata details. Zheng et al. (2021) expanded on their technique by including attention mechanisms in order to learn the structures of headlines and body contents. They used the representations obtained from the headline to extract enticing features and then compared the similarity between the headline and body content of the article.

Till now, very few works have explored the similarity/dissimilarity in the information presented in different components of clickbait posts. However, the headlines, titles, and

¹ <https://webis.de/events/clickbait-challenge/shared-task.html>.

body content are the components read by a user and hence are the better options to focus on for clickbait identification. However, using all sentences in the body content is not a good option as the body content is usually large and consists of several sentences. Processing a large number of sentences may introduce noise as some of the body sentences will be similar and others will be dissimilar to the headline, impacting the performance of clickbait identification. It will also make the model computationally inefficient. The problem motivated us to select k sentences from the body content keeping the various themes of the body. These k sentences from the paragraph are combined with headlines and titles to identify clickbait in the proposed model. Our major contributions of this research work are as follows:

- A *Dissimilarity Based Clickbait Detection (DBCD)* model is proposed by integrating valuable information from both the titles and body contents. Generally, news article content usually consists of an enormous number of sentences; some of these sentences may be redundant or unconnected and may not be necessary. Only $k < l$ (total sentences in a paragraph) sentences from a paragraph that are dissimilar to each other while still covering all the information from the target content are selected to minimize the computational overhead.
- An evaluation of the proper way of combining features extracted from *headline* (Post Text), *titles* (Target Title), and *body content* (Target Paragraph) is done using two approaches: (i) Concatenated sentence embedding and (ii) Average sentence embedding, to get feature vector representation of the entire clickbait content and evaluate the effectiveness of the ML classifiers.
- The extensive experiments on two widely used datasets demonstrated that the proposed model outperforms existing state-of-the-art models.

The remaining sections of the paper are structured as follows: In Sect. 2, previous work done in the domain of clickbait and false news is presented. Section 3 defines the problem statement. Section 4 presents the model developed to detect clickbait. Section 5 presents the experimental settings, performance metrics and results. In Sect. 6, the results are discussed and compared with different state-of-the-art models. Lastly, the paper is concluded in Sect. 7 pointing to some future directions.

2 Related work

Recently, clickbait has emerged as a significant concern in online content consumption and has attracted the attention of numerous research workers. However, most attempts used

heavy manual feature engineering and deep learning (DL) approaches to identify clickbait over news headlines.

The feature-engineering-based detection strategy manually extracts features and identifies the clickbait using traditional ML techniques. In the previous studies, researchers (Cao et al. 2017; Potthast et al. 2016; Sisodia 2019) employed this technique to solve the issue of clickbait. One of the earliest techniques based on ML was proposed by Potthast et al. (2016), who created a dataset of 2992 tweets, of which 767 were clickbait. They trained an RF classifier using 215 features derived from teaser messages, linked web pages, and meta-information to achieve the best performance measures with an area under the curve (AUC) value of 0.79, precision of 0.76, and recall of 0.76. Biyani et al. (2016) distributed the clickbait into eight categories: (i) teasing, (ii) ambiguous, (iii) exaggerated, (iv) inflammatory, (v) bait-and-switch, (vi) formatting, (vii) wrong, and (viii) graphic. They classified the dataset using a gradient-boosted decision tree (GBDT) with URL, title, similarity between headline and text, use of references, and informality as features. They found that the model performed best with informality and forward reference features to achieve an F_1 -score of 74.9%. Cao et al. (2017) selected the best 60 features using the Fisher score, trained on RF regression and reported an accuracy of 0.819, mean squared error (MSE) of 0.035, and F_1 -score of 0.61. Papadopoulou et al. (2017) empirically evaluated the fitness of individual features and their combinations which were derived from the post text and the target title.

Sisodia (2019) created a dataset for clickbait identification by collecting 60,000 clickbait headlines and 40,000 non-clickbait headlines from the New York Times. They extracted 19 features from article headlines and achieved an accuracy of 0.911 and F_1 -score of 0.91 with an RF classifier. Coste and Bufnea (2021) used the Webis-Clickbait-17 dataset² to extract grammatical, orthographic, text similarity measures, and linguistic features that are independent of any particular language. Using the SVM classifier, they obtained an accuracy of 0.74, precision of 0.75, recall of 0.71, and F_1 -score of 0.73. Indurthi and Oota (2017) also used the same clickbait challenge dataset to identify the clickbait. They concatenated seven handcrafted features with the 300-GloVe embedding as a feature vector from the headline to get a total of 307 features for training logistic regression (LR) and obtained an F_1 -score of 64.98%. Geckil et al. (2018) collected 1000 clickbait posts and 1000 non-clickbait posts from news websites of media companies and Twitter. They investigated the suitability of term frequency-inverse document frequency (TF-IDF) by comparing the frequencies of being clickbait and also using the reliability index to

² <https://zenodo.org/record/3346491>.

categorize news headlines. This approach had an accuracy of 86.5%, precision of 0.899, recall of 0.941, and F_1 -measure of 0.920.

Recently, many researchers have focused on extracting features using DL models. Deep neural networks (DNN) have gained significant success due to their ability to extract semantic, syntactic and context-based features. Regarding the clickbait detection approach, the related research might be split into three specific categories: title-based, context-based strategies, and semantic similarity-based.

Title-based detection strategies concentrated on extracting the characteristics of news titles. Zhou (2017) used the word embedding of post text to train BiGRU and a self-attention mechanism to detect the clickbait score. Their model outperformed the baseline approaches in terms of accuracy of 0.856, F_1 -score of 0.683, and MSE of 0.033. Chawda et al. (2019) collected a total of 7500 headlines from various websites. They discovered that the recurrent convolutional neural network (RCNN)+ gated recurrent unit (GRU) model using pre-trained word2vec-based contextual information worked more effectively than SVM and achieved an accuracy of 0.9776. Bronakowski et al. (2023) used a clickbait headline dataset.³ They extracted 30 semantic features from this corpus and found that SVM and GBDT achieved the highest accuracy of 98%.

Similarly, Agrawal (2016), Glenski et al. (2017), Manjesh et al. (2017), Pandey and Kaur (2018) and Patil et al. (2021) have utilized DNN to uncover the underlying characteristics of clickbait. Agrawal (2016) collected 814 clickbait headlines and 1574 non-clickbait headlines from social media. Textual headlines are transformed into word embeddings, which are subsequently employed as input by a convolutional neural network (CNN)-based model to identify the clickbait and obtain an accuracy of 0.90, precision of 0.85, and recall of 0.88. Glenski et al. (2017) employed CNN and long short-term memory (LSTM) models for clickbait classification and obtained an F_1 -score of 0.69. Manjesh et al. (2017) collected 62,000 headlines from clickbait websites. They used LSTM to analyze the clickbait headlines and achieved an average precision of 99%, recall of 98%, and F_1 -score of 98%. Pandey and Kaur (2018) collected 39,400 articles from various news sources, comprising both clickbait and non-clickbait content. They developed a bidirectional long short-term memory (BiLSTM) model with GloVe embeddings to predict clickbait headlines, achieving an impressive accuracy of 98.78%. Additionally, they utilized a genetic algorithm for hyperparameter optimization with an accuracy of 94.36% for LSTM and 95.61% for BiLSTM. Klairith and Tanachutiwat (2018) contributed a clickbait corpus of 30,000 headlines from well-known newspapers using crowdsourcing. In this

work, a clickbait detection model was created employing two types of features embedded at the word and character levels in the embedding layer and three distinct networks in the hidden layer. They found that BiLSTM with word-level embedding performed very well, with an accuracy of 0.98 and F_1 -score of 0.98. Patil et al. (2021) collected 814 clickbait samples and 1574 non-clickbait samples and used pre-trained word2vec embedding from headlines as input features to BiLSTM, followed by attention mechanisms. Naeem et al. (2020) created a dataset with 16,000 clickbait and non-clickbait headlines and developed an LSTM-based model with 300 dimensional word2vec embedding to achieve an accuracy of 0.97. They also developed a model with part of speech analysis module (POSAM) to get an accuracy of 0.88. Razaque et al. (2022) proposed a deep recurrent neural network (RNN) using source rating analysis that examined 1800 legal websites with an accuracy of 0.9983. Ma et al. (2022) created a dataset of 12,000 posts with 6,000 as clickbait and another 6000 as non-clickbait posts from Chinese news websites. They developed a CNN-LSTM-based model with title embedding, content embedding, and 18 manual features to obtain an accuracy of 98.42%.

Context-based features focused on the feedback from individuals reading online content. Al-Sarem et al. (2021) extracted both user- and content-based features and observed that SVM with the top 10 selected features using selection by analysis of variance (ANOVA) F-test performed better with an accuracy of 92.16%. Varshney and Vishwakarma (2021) used the fake video corpus [FVC-2018]⁴ and misleading video dataset (MVD). They extracted user profiling, human consensus, and video-content-based features from these datasets. They reported that the J-48 classifier performed best with an accuracy of 98.89% on the MVD dataset. The transfer learning approach is also used to identify clickbait. Rajapaksha et al. (2021) used transfer learning models BERT, Roberta, and XLNet and fine-tuned them to get an accuracy of 0.858 and F_1 -score of 0.69 on the Clickbait Challenge-2017 dataset with Roberta model.

Semantic sentence similarity has been used in many research areas such as summarization (Gupta et al. 2023a, b), hate speech detection, clickbait detection (Zheng et al. 2021; Kumar et al. 2018), and spam email classification (Srinivasarao and Sharaff 2023a, b). In clickbait, semantic similarity is used to find the similarity/dissimilarity between headlines and the body contents. Most of the time, the body contents of clickbait news cannot satisfy the reader's expectations generated by the headlines.

Dong et al. (2019) split their work into three distinct parts: obtaining hidden representations through learning, matching similarities, and recognizing clickbait by

³ <https://www.kaggle.com/datasets/amananandrai/clickbait-dataset>.

⁴ <https://zenodo.org/record/2535479>.

combining the features with the similarities. They used two different datasets: the clickbait challenges dataset and the Fake News Challenge (FNC) dataset. They trained four models across these datasets, including CNN, RNN, and bi-AttGRU, an attention-based bi-directional GRU. Notably, bi-AttGRU outperformed other models in learning latent representations, including similarity insights and an attention mechanism for the final prediction. The clickbait challenge dataset had an accuracy of 86% and F_1 -score of 71%, whereas the FNC dataset had an accuracy of 89.4% and F_1 -score of 92.8%. Kumar et al. (2018) performed experiments using a test corpus of 19,538 social media posts. They extracted visual features by employing a visual geometry group (VGG)-19 pre-trained model and used a Siamese network to measure the similarity between source and target data. They reported an F_1 -score of 65.37% on a BiLSTM with an attention mechanism. Zheng et al. (2021) created the Chinese clickbait dataset, which includes almost 5,000 posts from news websites. They computed the luring score of headlines using the BiLSTM layer, followed by self-attention, and also employed stack transformer encoder blocks to find the similarity between post headlines and content. They achieved an accuracy of 88% and an F_1 -score of 74%. Meng et al. (2022) used a multilayer gated convolutional network to understand the connections and dependencies between words at both the local and long-distance levels and then developed a deep relevance matching network called attention-fused deep relevance matching network (ARMN) that efficiently identifies the similarity between headline and body contents. They made use of the Clickbait Challenge 17 dataset in addition to the TouTiao text classification for news titles (TNEWS) dataset.⁵ In terms of F_1 -score, the model's performance on the TNEWS dataset was 88.28%, and on the Webis clickbait challenges dataset, the authors were able to achieve an F_1 -score of 59.14% with a precision of 62.18%. Bronakowski et al. (2023) identifies clickbait headlines using semantic analysis and ML techniques. They used 30 unique semantic features and fed them to six different ML classification algorithms individually and in ensemble forms. They obtained the highest accuracy of 98% in classifying clickbait headlines. They serve their work as a template for developing practical applications to detect clickbait headlines automatically. However, the limitation of this work is that this classification is based on clickbait headlines only.

Based on the above literature analysis, the following research gap in existing clickbait detection techniques has been observed. Traditional ML algorithm-based models are based on manual feature engineering, which may not properly capture clickbait content's complex and ever-changing

properties. Traditional methods may not scale effectively to deal with the massive amount of content posted in real-time on social media networks. Most of the work done by the researcher does not include the body content of the clickbait post which is a major limitation and gap in the existing works. Incorporating user comments and preferences into the clickbait detection process to improve model performance and adjust to user expectations is a challenge. In order to effectively detect clickbait, existing studies focused on headline representations to extract lure-based features and determine how catchy and attractive they were and how closely they matched with the titles. However, these approaches do not capture the context of the body content and its dissimilarity with the corresponding headline and title. Further exploration and investigation of body content analysis techniques can contribute to advancements in clickbait detection methodologies. It is clear that there is a noticeable difference between headlines, which typically consist of a single sentence, and body content, which often comprise multiple sentences. The research work mentioned is the first attempt to identify clickbait by selecting the minimum number of dissimilar sentences from the body content.

3 Problem formulation

Let $N = N_1, N_2, \dots, N_i, \dots, N_n$ be a set of news, where n is the number of clickbait news articles in the dataset. $N_i = \{H_i, T_i, B_i\}$ is the i th clickbait news, where H_i refers to the headlines (Post Text) of the clickbait news, T_i refers to the title (Target Title) of clickbait news N_i , and B_i means the body content (Target Paragraph) of clickbait news N_i . Every headline H_i consists of a sequence of words, represented as the $H_i = \{h_{i1}, h_{i2}, \dots, h_{ip}\}$, where p is the number of words in the headlines of the clickbait news article in the dataset. Each title T_i consists of a sequence of words, which is denoted as $T_i = \{t_{i1}, t_{i2}, \dots, t_{im}\}$, where m is the sequence of the words in a title. Each content $B_i = \{s_{i1}, s_{i2}, \dots, s_{il}\}$ contains l sentences, and s_{il} refers to the l th sentence of the body content B_i .

The clickbait detection task is to assign a probability score of it being a clickbait to N_i . Mathematically, this can be represented as:

$$P(y = c | (N_i)) = P(y = 1 | (H_i, T_i, B_i)) \quad (1)$$

where N_i refers to a clickbait news article that contains headlines (H_i), titles (T_i), and body content (B_i). $c \in \{0, 1\}$ where $c = 1$ means that the news is clickbait news, and $c = 0$ means that the news is not clickbait news.

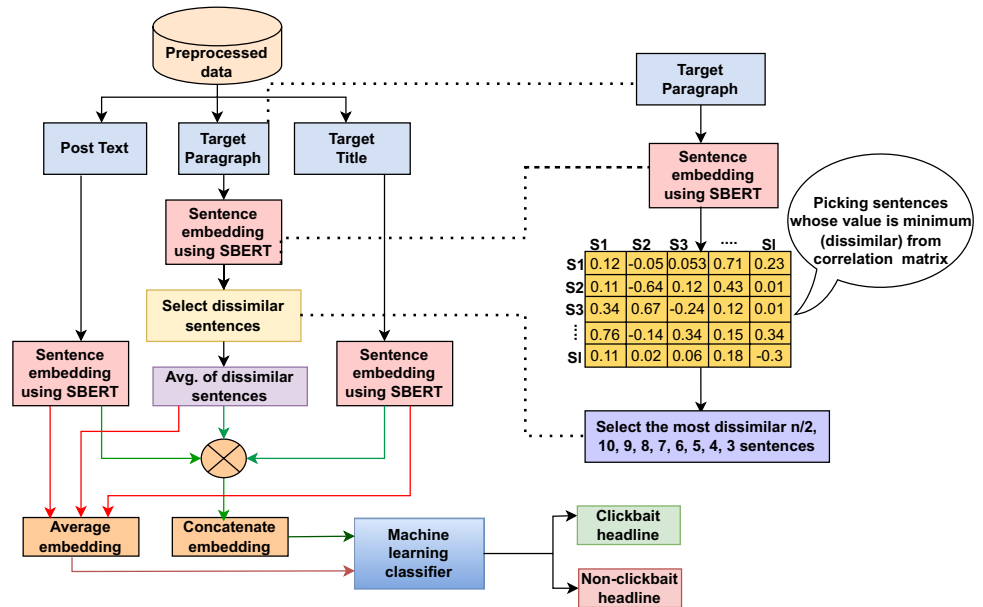
⁵ [https://metatext.io/datasets/toutiao-text-classification-for-news-titles-\(tnews\)-\(clue-benchmark\)](https://metatext.io/datasets/toutiao-text-classification-for-news-titles-(tnews)-(clue-benchmark)).

Table 1 The details of dataset 1 and dataset 2

Datasets	Total samples	Clickbait samples	Non-clickbait samples
Dataset 1	22,033	5523	16,477
Dataset 2	32,000	15,999	16,001

Dataset 2 The second dataset utilized in the research was compiled by Chakraborty et al. (2016). It consists of 32,000 headlines collected from different web domains that publish clickbait news articles, such as ‘Scoopwhoop’, ‘ViralNova’, ‘UpWorthy’, ‘ViralStories’, ‘Wikinews’, ‘New York Times’, ‘The Hindu,’ and ‘The Guardian’. The dataset is a balanced dataset with 15,999 labeled as clickbait headlines, whereas 16,001 were labeled non-clickbait headlines. Complete sta-

Fig. 2 Flow diagram of proposed model



3.1 Data description

In this study, two datasets are utilized to validate the experiment:

Dataset 1 The first dataset is taken from the Weibis Clickbait Corpus 2017 dataset⁶ released in 2017. The training and validation datasets, which included 19,538 and 2459 tweets, respectively, are combined to make a large training dataset of 22,033 labeled data samples. Each post includes “IDs”, “labels”, “post text”, “target title”, “target keyword”, “post timestamp”, “media”, “caption attached to the image”, and “target paragraph”. This work focuses only on “post text (headlines)”, “target title (title)”, “target paragraph (body content)”, and “label”. These data samples were then divided into an 80:20 ratio, with 80% of the data used as a training sample and the remaining 20% used for testing. The purpose of this division was to train the models and evaluate their performance by comparing them with benchmark models that have been implemented using state-of-the-art techniques.

tistics of the datasets are presented in Table 1.

4 Proposed model

The proposed model for detecting clickbait post using headlines, titles, and body content is depicted in Fig. 2. In the current work, all the major components of a clickbait post, such as headlines, titles and body content, are considered to extract features from them. The body content of clickbait news articles often contains a huge number of sentences, some of which are unrelated or misleading, so considering every sentence of body content can be computationally inefficient due to its extensive length. One of the primary tasks was to minimize the number of sentences from the body content, but it should cover complete information about the whole body content. To achieve this, an algorithm was developed to select the most dissimilar sentences from the body content of the clickbait news articles. This was accomplished by creating a correlation matrix using the SBERT-encoded vector representations (Reimers and Gurevych 2019) of a paragraph. The correlation matrix is generated with the number of sentences in the body content

⁶ <https://zenodo.org/record/3346491>.

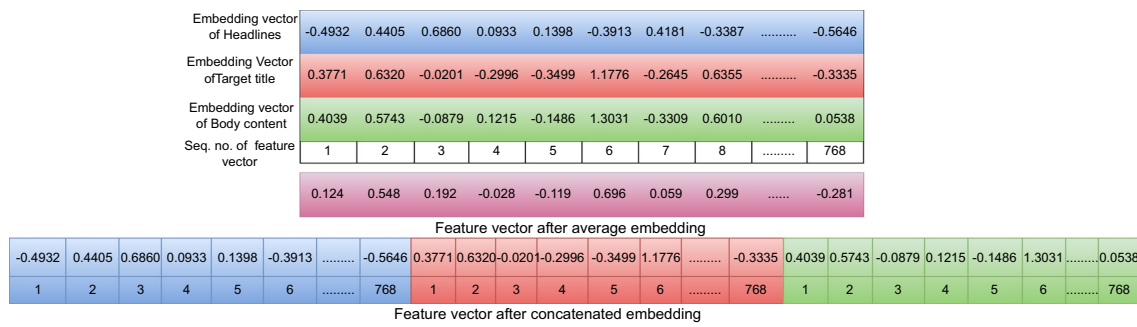


Fig. 3 Diagram of embedding feature vector after feature after concatenation and averaging

of the clickbait posts. The correlation matrix has an equal number of rows and columns, which is equal to the number of sentences in the body content. The resulting correlation matrix allows one to determine the relationship between the sentences within the paragraph. The most k dissimilar sentences having lowest correlation values are picked from the matrix. The data flow diagram consists of the following phases: (i) Data pre-processing, (ii) Feature representation, and (iv) Classification by ML classifiers.

4.1 Data pre-processing

Data pre-processing is necessary for each work to clean and prepare the data to extract the important features. The first step in pre-processing is to discard the data sample whose body content (target paragraph) has four sentences or fewer to keep enough text for training. All the emojis and Unicode are removed from the document during this step because this research only concentrates on textual content. All the sentences are converted to their root form using lemmatization. The sentences are further converted to lowercase. Label encoding is used to encode the label value (clickbait and non-clickbait) of the data sample into binary values 1 and 0, representing clickbait as 1 and non-clickbait as 0. After pre-processing, we had only 5075 clickbait samples and 14,901 non-clickbait samples in the clickbait dataset 1. No pre-processing was done on dataset 2, as it contained clean headlines only.

4.2 Feature representation

In the case of clickbait posts, most of the time, the headlines are presented to the user for clicking, and the headlines do not match with the body content associated with that headline. Hence, it becomes an important part of clickbait identification to focus on the headlines (Post Text), the title (Target Title), and the body content (Target Paragraphs). The proposed approach, known as *Dissimilarity Based Clickbait Detection (DBCD)*, considers all the above-mentioned

components of the news articles to identify as clickbait. The method used after feature extraction is shown in Fig. 3.

Each headline is converted to 768-sized vectors using the SBERT pre-trained model. Similarly, the target titles were also embedded as a vector of 768 using the SBERT pre-train model. Extracting representative sentences from the body content of the posts served two objectives: (i) minimizing the number of selected sentences (k) from the body, and (ii) ensuring that the chosen sentences comprehensively cover the information contained in the entire body content. The first objective was determined empirically by experimenting with different numbers of sentences. The empirically determined value is explained in the result section. k -most dissimilar sentences from the body content are selected to achieve the second objective.

For selecting the k most dissimilar sentences, each sentence of body content is embedded into an embedded vector of size 768 using the pre-trained SBERT embedding technique as shown in Fig. 2. The body content $B_i = \{s_{i1}, s_{i2}, \dots, s_{il}\}$ contains l sentences where s_{il} refers to the l th sentence of the body content B_i . These sentences were represented as vectors in space X , representing $X \in R^{l \times d}$, where l represents the number of sentences and d represents a 768-dimensional vector for each sentence. Each sentence of the body content is converted to an embedded vector using SBERT, yielding a d -dimensional vector for each of the l sentences.

Each sentence of the paragraph is iterated to calculate its similarity score with all the sentences already in it. These methods involve comparing the encoded representations or embeddings of the sentences to determine their similarity. Subsequently, they generate the correlation matrix with dimensions of $l \times l$ using cosine similarity between sentence-embedding vectors. Cosine similarity ranges from -1 to 1 , where 1 indicates strong similarity, and -1 indicates dissimilarity. We pick both the corresponding sentences from the correlation matrix having the lowest correlation value. The dissimilarity between the two sentences is computed using the cosine similarity by the equation given below.

Algorithm 1 The pseudocode for finding dissimilar sentences

Data: list of l number of sentences of body content B_i
Result: A list $diss_sentences$ containing k lowest similar sentences

```

1 Initialize an empty list embeddinglist of sentences;
2 for sentence in  $B_i$  do
3   Embedding ← Encode(sentence);
4   embeddinglist.append encoded sentences;
5 end
6 Main
7   Initialize an empty list diss_sentences;
8   for  $i \leftarrow 1$  to  $len(embeddinglist)$  do
9     Initialize an empty dictionary score_dict;
10    for  $j \leftarrow 1$  to  $len(embeddinglist)$  do
11      if  $i \neq j$  then
12        score ← cosine similarity(embeddinglist[i],
13        embeddinglist[j]);
14        score_dict[j] ← score;
15      end
16    end
17    sorted_scores ← sort score_dict by values in ascending order;
18    top_k_indices ← first  $k$  indices from sorted_scores;
19    for  $idx$  in top_k_indices do
20      diss_sentences.append( $B_i[idx]$ );
21    end
22 end
23 return diss_sentences;
```

$$\cos(s_1, s_2) = \frac{s_1 \cdot s_2}{\|s_1\| \cdot \|s_2\|} \quad (2)$$

The pseudocode for finding dissimilar sentences is given in Algorithm 1.

The algorithm returns the list of k dissimilar sentences whose mean values are taken as an encoded embedding vector of the body content B_i used for further processing.

4.3 Feature vector

The extracted embedding vectors from the headline, body content, and title are combined in two ways: (i) Concatenated vector and (ii) Average vector. In the first technique, the embedded vectors of headlines, the title, and the body content are placed one after the another to create a feature vector size of length 2304 (768×3), shown in the lower part of Fig. 3. The second way of creating a feature vector was averaging the value of embedded vectors of the headlines, the title, and the body content. The resulting feature vector size is only 768 as it is the average of all the different components, shown in the upper part of Fig. 3.

4.4 Machine learning classifier

The concatenated sentence embedding and average sentence embedding of headlines, the title, and the body content are used as features to train and test the ML classifiers. The

performance of the model was evaluated on eight different ML classification models: (i) decision tree (DT), (ii) LR, (iii) naive Bayes (NB), (iv) SVM, (v) K-nearest neighbor (KNN), (vi) decision tree (DT), (vii) ada-boost (AB) classifier, and (viii) RF classifier. The dataset was randomly split into an 80:20 ratio. Various train-test split ratios (70:30 and 85:15) were also evaluated, but no significant changes were observed in the performance of the models.

5 Results

In this section, the experiment results on two real-world datasets to identify news articles as clickbait and non-clickbait are presented. The following definitions are given with respect to the clickbait class. A similar definition also applies for the non-clickbait class.

The model performance was evaluated using four well-known evaluation metrics: accuracy, precision, recall, and F_1 -score.

True Positive (TP) The model has correctly predicted the clickbait class as clickbait class.

False Positive (FP) The model incorrectly predicted the non-clickbait class as the clickbait class.

True Negative (TN) The model has correctly predicted the non-clickbait class as a non-clickbait class.

False Negative (FN) The model incorrectly predicted the clickbait class as a non-clickbait class.

Table 2 The results of the ML classifiers using $l/2$ sentences on dataset 1 with concatenated and average embedding

ML Classifier	Class	Concatenate				Average			
		P	R	F_1	Acc.	P	R	F_1	Acc.
SVM	Clickbait	0.79	0.49	0.60	0.84	0.78	0.48	0.59	0.83
	Non-clickbait	0.85	0.96	0.90		0.84	0.95	0.90	
	weighted avg	0.83	0.84	0.82		0.83	0.83	0.82	
RFC	Clickbait	0.77	0.32	0.45	0.80	0.79	0.30	0.44	0.80
	Non-clickbait	0.81	0.97	0.88		0.81	0.97	0.88	
	Weighted avg	0.80	0.80	0.77		0.80	0.80	0.77	
LR	Clickbait	0.71	0.58	0.64	0.83	0.68	0.57	0.62	0.82
	Non-clickbait	0.87	0.92	0.89		0.86	0.91	0.88	
	Weighted avg	0.83	0.83	0.83		0.82	0.82	0.82	
GB	Clickbait	0.74	0.44	0.56	0.82	0.74	0.42	0.54	0.82
	Non-clickbait	0.84	0.95	0.89		0.83	0.95	0.89	
	Weighted avg	0.81	0.82	0.80		0.81	0.82	0.80	
DT	Clickbait	0.43	0.47	0.45	0.71	0.43	0.46	0.44	0.71
	Non-clickbait	0.82	0.79	0.80		0.81	0.79	0.80	
	Weighted avg	0.72	0.71	0.71		0.72	0.71	0.71	
AB	Clickbait	0.67	0.46	0.55	0.81	0.66	0.46	0.54	0.80
	Non-clickbait	0.84	0.92	0.88		0.84	0.92	0.88	
	Weighted avg	0.80	0.81	0.80		0.79	0.80	0.79	
NB	Clickbait	0.49	0.69	0.57	0.74	0.47	0.67	0.55	0.73
	Non-clickbait	0.88	0.75	0.81		0.87	0.75	0.80	
	Weighted avg	0.78	0.74	0.75		0.77	0.73	0.74	
KNN	Clickbait	0.67	0.38	0.49	0.80	0.68	0.39	0.50	0.80
	Non-clickbait	0.82	0.94	0.87		0.82	0.94	0.88	
	Weighted avg	0.78	0.80	0.78		0.79	0.80	0.78	

Accuracy Accuracy is commonly defined as the ratio of correctly predicted instances divided by the total number of instances in the dataset for all classes.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

Precision Precision for clickbait class is defined as the number of predictions correctly identified as clickbait divided by all predictions as clickbait.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

Recall The recall for clickbait class is calculated as the ratio between the number of clickbait instances correctly predicted to the total number of clickbait instances in the dataset.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

F₁-score The F_1 -score, which is the harmonic mean of precision and recall.

$$F_1\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

The first experiment was conducted to decide which feature representation method, concatenating or averaging the extracted feature vectors from the title, headline, and body content, is better for identifying the clickbait. Both methods of feature representations are fed into the ML classifier to classify the clickbait, and the obtained performances are shown in Table 2. For this experiment, the value of k is $l/2$, where l is the number of sentences in the body content. All the classifiers performed better on concatenated feature vectors than the average feature vectors. The SVM performs best among all the ML classifiers to achieve the highest accuracy of 0.84, weighted precision of 0.83, weighted recall of 0.84, and weighted F_1 -score of 0.82, as shown in bold in Table 2. Hence, the SVM classifier with the concatenated embedding feature was considered for further experiments.

The second experiment has been done to select the number of dissimilar sentences from the body content (B_i), which can cover the whole body content information. The

Table 3 The results of different sets of sentences using SVM for dataset 1

Sentences	Class	Concatenate			
		P	R	F_1	Acc.
2 sentences	Clickbait	0.76	0.47	0.58	0.83
	Non-clickbait	0.84	0.95	0.89	
	Weighted avg	0.82	0.83	0.81	
3 sentences	Clickbait	0.78	0.48	0.60	0.83
	Non-clickbait	0.84	0.95	0.89	
	Weighted avg	0.82	0.83	0.81	
4 sentences	Clickbait	0.79	0.48	0.60	0.83
	Non-clickbait	0.84	0.95	0.89	
	Weighted avg	0.82	0.83	0.81	
5 sentences	Clickbait	0.77	0.49	0.60	0.83
	Non-clickbait	0.84	0.95	0.89	
	Weighted avg	0.82	0.83	0.81	
6 sentences	Clickbait	0.77	0.49	0.60	0.84
	Non-clickbait	0.85	0.95	0.90	
	Weighted avg	0.83	0.84	0.82	
7 sentences	Clickbait	0.74	0.49	0.59	0.83
	Non-clickbait	0.85	0.94	0.89	
	Weighted avg	0.82	0.83	0.82	
8 sentences	Clickbait	0.75	0.50	0.60	0.83
	Non-clickbait	0.85	0.95	0.89	
	Weighted avg	0.82	0.83	0.82	
9 sentences	Clickbait	0.75	0.45	0.56	0.80
	Non-clickbait	0.84	0.95	0.89	
	Weighted avg	0.82	0.82	0.81	
10 sentences	Clickbait	0.74	0.49	0.59	0.83
	Non-clickbait	0.84	0.94	0.89	
	Weighted avg	0.82	0.83	0.81	

experimentation was done by varying the number of sentences from 2 to 10 and $l/2$. The precision, recall, accuracy, and F_1 -score using concatenated features vector over dissimilar sentences from 2 to 10 on dataset 1 are listed in Table 3.

The selected six dissimilar sentences from the body content performed better than all the other sets of dissimilar sentences, as shown in bold in Table 3, which is also similar to $l/2$ dissimilar sentences' performance. The performance of both sets of sentences obtained an accuracy of 0.84, weighted precision of 0.83, weighted recall of 0.84, and weighted F_1 -score of 0.82, as shown in Tables 2 and 3. The effect of increasing and decreasing the value of dissimilar sentences is shown in Fig. 4. This Fig. 4 shows that the weighted recall and accuracy of the six dissimilar sentences are higher than the other sets of dissimilar sentences. When decreasing or increasing the number of dissimilar sentences, the precision, recall, F_1 -score, and accuracy of the model are degraded. From now on, all the experiments are done with the six dissimilar sentences.

While observing the performance of the proposed model, it was found that for the non-clickbait class, the model performed well, but for the clickbait class, it performed poorly, achieving a recall of 0.49 only. The confusion matrix for the classification of the unbalanced dataset shown in Fig. 5 also confirms that the recall is only 49% while 51% of the clickbait class is misclassified as non-clickbait. The third experiment was done to improve the performance of the clickbait class; it was done by balancing dataset 1. The dataset 1 is balanced using the under-sampling method and the performance of the ML model was measured with eight different sets of data samples. The clickbait data samples in dataset 1 are 5075, and the non-clickbait data sample comprises 16,477, as mentioned in Table 1. To balance dataset 1, all

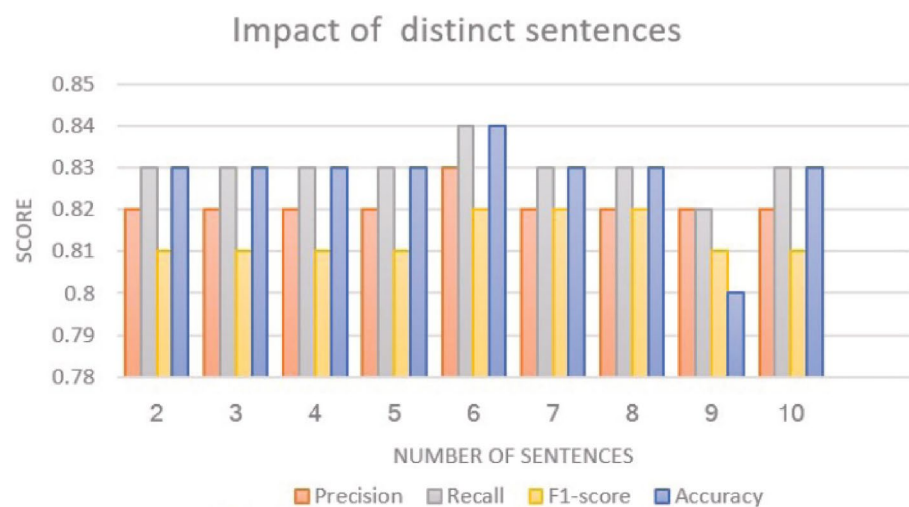
Fig. 4 Impact of different number of sentences on **a** Precision, **b** Recall, **c** F_1 -score and **d** Accuracy for SVM

Table 4 The results of the ML classifiers after balancing (under-sampling) the dataset 1

Sample Dataset	Seed-value	Class	Concatenate			
			P	R	F_1	Acc.
BD ₁	Seed = 1	Clickbait	0.78	0.75	0.77	0.78
		Non-clickbait	0.77	0.801	0.79	
		Weighted avg.	0.78	0.78	0.78	
BD ₂	Seed = 2	Clickbait	0.79	0.75	0.77	0.78
		Non-clickbait	0.77	0.81	0.79	
		Weighted avg.	0.78	0.78	0.78	
BD ₃	Seed = 3	Clickbait	0.79	0.76	0.77	0.78
		Non-clickbait	0.78	0.81	0.79	
		Weighted avg.	0.78	0.78	0.78	
BD ₄	Seed = 4	Clickbait	0.78	0.76	0.77	0.781
		Non-clickbait	0.78	0.80	0.79	
		Weighted avg.	0.78	0.78	0.78	
BD ₅	Seed = 5	Clickbait	0.78	0.84	0.81	0.79
		Non-clickbait	0.81	0.75	0.78	
		Weighted avg.	0.80	0.79	0.79	
BD ₆	Seed =v7	Clickbait	0.78	0.75	0.76	0.77
		Non-clickbait	0.77	0.80	0.78	
		Weighted avg.	0.77	0.77	0.77	
BD ₇	Seed = 8	Clickbait	0.78	0.75	0.76	0.77
		Non-clickbait	0.77	0.80	0.78	
		Weighted avg.	0.77	0.77	0.77	
BD ₈	Seed = 11	Clickbait	0.78	0.75	0.77	0.78
		Non-clickbait	0.77	0.80	0.781	
		Weighted avg.	0.78	0.78	0.78	
Mean	-	Clickbait	<u>0.78 ± 0.005</u>	<u>0.76 ± 0.02</u>	<u>0.77 ± 0.015</u>	<u>0.77 ± 0.010</u>
		Non-clickbait	<u>0.77 ± 0.015</u>	<u>0.79 ± 0.019</u>	<u>0.77 ± 0.015</u>	
		Weighted avg.	<u>0.78 ± 0.008</u>	<u>0.77 ± 0.010</u>	<u>0.78 ± 0.007</u>	

the clickbait data samples are mixed with 5216 randomly selected data samples from the non-clickbait class.

The first set of non-clickbait samples was selected using the random state seed-value = 1, while splitting the non-clickbait samples using the *train – test – split* function of *sklearn*. This was merged with the clickbait samples and shuffled to make a balanced dataset, naming them to balance dataset-1 (BD₁). A similar process is used to make the eight sets of different datasets using different seed values 1, 2, 3, 4, 5, 7, 8, and 11. The embedding vector of the sample datasets is saved into separate comma-separated values (CSV) files, named BD₁, BD₂, BD₃, BD₄, BD₅, BD₆, BD₇, and BD₈. The SVM model is trained and tested with these datasets and the performances are listed in Table 4.

The DB₅ data samples that performed better among the other sets of data samples shown in bold in Table 4 and achieved the highest precision, recall, and F_1 -score of the clickbait class were 0.78, 0.84 and 0.81, respectively. The confusion matrix for the classification of DB5 data samples is shown in Fig. 6, which indicates that the recall of the clickbait class has increased to 75%, compared to only 49%

for the unbalanced dataset. The last row of Table 4 shows the mean value of all the results on the sampled dataset. The mean value is very similar to the individual results of the sample dataset. The standard deviation is also shown with the mean precision, recall, and F_1 -score value. The value may be increased or decreased up to the standard deviation value after changing the seed value in the sample dataset. The precision, recall, and F_1 -score for the non-clickbait class are 0.81, 0.75, and 0.78, respectively. The accuracy of the proposed model is decreased after balancing the dataset, but the precision, recall, and F_1 -score of the clickbait class increases. The mean of all the performance metrics for clickbait and non-clickbait classes is calculated to find the deviation on the different sample datasets. The mean precision, recall and F_1 -score of the clickbait class are found to be 0.78, 0.76, and 0.77, respectively, while the mean precision, recall and F_1 -score for the non-clickbait class are found to be 0.77, 0.79, and 0.77, respectively, as shown by the underlined values in the last row of Table 4.

The performance of ML classifiers over dataset 2 is shown in Table 5. For dataset 2, SVM performed better

Table 5 Result of dataset 2 on ML classifiers using Sentence-BERT embedding (S-BERT)

ML classifier	Class	P	R	F_1	Acc.
SVM	Clickbait	0.98	0.98	0.98	0.98
	Non-clickbait	0.98	0.98	0.98	
	Weighted avg	0.98	0.98	0.98	
RFC	Clickbait	0.94	0.93	0.93	0.93
	Non-clickbait	0.93	0.94	0.93	
	Weighted avg	0.93	0.93	0.93	
LR	Clickbait	0.97	0.98	0.97	0.97
	Non-clickbait	0.98	0.97	0.97	
	Weighted avg	0.97	0.97	0.97	
GB	Clickbait	0.94	0.95	0.95	0.95
	Non-clickbait	0.95	0.95	0.95	
	Weighted avg	0.95	0.95	0.95	
DT	Clickbait	0.83	0.83	0.83	0.83
	Non-clickbait	0.83	0.83	0.83	
	Weighted avg	0.83	0.83	0.83	
AB	Clickbait	0.92	0.92	0.92	0.92
	Non-clickbait	0.92	0.92	0.92	
	Weighted avg	0.92	0.92	0.92	
NB	Clickbait	0.88	0.86	0.87	0.87
	Non-clickbait	0.86	0.88	0.87	
	Weighted avg	0.87	0.87	0.87	
KNN	Clickbait	0.95	0.93	0.94	0.94
	Non-clickbait	0.93	0.95	0.94	
	Weighted avg	0.94	0.94	0.94	

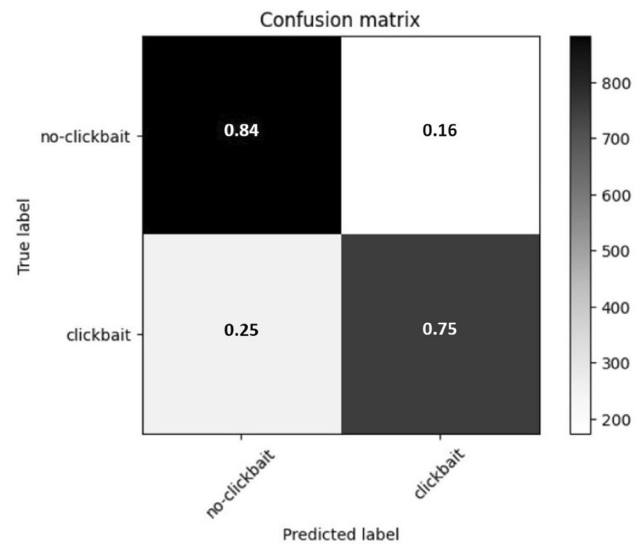


Fig. 6 Confusion matrix of the proposed model on balance dataset 1

-score for both clickbait and non-clickbait classes of 98%, as shown in bold in Table 5.

5.1 Statistical test for consistency

To prove that the results are consistent across several runs, the experiment was repeated for each dataset and their performances are noted. The mean value for weighted precision, weighted recall, weighted F_1 -score and accuracy is presented in the last row of Table 4. The mean weighted precision, weighted recall, weighted F_1 -score, and accuracy are 0.78, 0.76, 0.77, and 0.77, respectively. The following hypotheses were formulated to establish that the results obtained from the proposed model are statistically consistent across various runs:

- Null hypothesis H_0 : There is no significant difference in the results across different runs.
- Alternative hypothesis H_1 : The test results vary significantly across different runs.

We conducted a sample t -test at a significance level of 0.05, in line with previous studies (Kumar et al. 2019). The calculated t -statistical values for accuracy, precision, recall, and F_1 -score are 1.428, 1.78, 1.428, and 1.428, respectively, and are less than the t -critical value (1.895) at a significance level (α) of 0.05, indicating that we can reject the alternative hypothesis. Statistically, it can be inferred that there is no difference in the results across different runs.

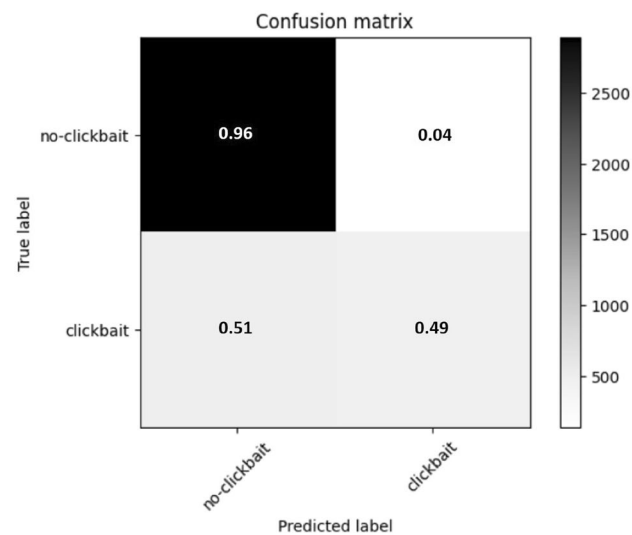


Fig. 5 Confusion matrix of the proposed model on unbalanced dataset 1

and obtained the highest accuracy, precision, recall, and F_1

Table 6 Performance comparison with different state-of-the-art models on Webis Clickbait Corpus 2017 (WCC) and Chakraborty et al. (2016) datasets

Dataset	Author	Model	Types	F_1 -score	Acc.
Webis Clickbait Corpus 2017	Zhou (2017)	BiGRU-ATT	Text	0.683	0.856
	Kumar et al. (2018)	Siamese Network	Text, image	0.653	0.855
	Dong et al. (2019)	Deep Semantic Similarity Model (DSSM)	Text	0.71	<u>0.86</u>
	Rajapaksha et al. (2021)	Transfer Learning	Text	0.69	0.858
	Coste and Bufnea (2021)	SVM, RFC	Text	<u>0.732</u>	0.739
	Liu et al. (2022)	Multiple Features for WeChat Clickbait Detection using BERT (MFWCD-BERT)	Text	0.71	0.81
	Meng et al. (2022)	Multilayer Gated Convolution Network	text	0.59	–
	Elyashar et al. (2022)	XGBoost	Text	–	0.81
	DBCBC (ours)	SVM with concatenated S-BERT embedding	Text	0.82	0.84
	Chakraborty et al. (2016)	SVM	Text	0.93	0.93
Chakraborty et al. (2016) Dataset	Chawda et al. (2019)	Recurrent CNN+GRU with Word2Vec	Text	–	0.97
	Kaur et al. (2020)	CNN-LSTM with glove embedding	Text	0.88	0.95
	DBCBC (ours)	SVM with S-BERT embedding	Text	0.98	0.98

6 Discussion

The first finding of the current research is that the feature vector obtained by concatenating the feature vectors of title, headline and body content performs better than the average of those feature vectors. The averaging of the feature vectors of the headline, title, and body content of the news article may lose some unique features from specific parts and hence perform less effectively while classifying the clickbait. The concatenated feature vector does not overlap with other features; it increases the size of the feature vector, which better represents the features from title, headline and body. Observations revealed that the model trained with the feature vector obtained from averaging the embedding vector demonstrated lower accuracy in predicting clickbait. It may be caused by model failure to fully capture the headlines, title and body content information in a low-dimensional feature vector.

The second finding is that when incorporating the top two, three, four, five, seven, eight, nine, and ten sentences, there were no notable changes in the performance of the SVM classifier, as shown in Fig. 4. However, with the top six dissimilar sentences from the body text the model performed considerably better. The SVM classifier's performance is similar for identifying the clickbait with 1/2 or six dissimilar sentences and achieved the highest accuracy using concatenated embedding. The SVM performs best with the concatenated embedding vector because it can handle high-dimensional data. Consideration is given to six distinct sentences to represent the body content of the

clickbait news articles. It reduces the space complexity and execution time of the model.

The third finding is that the balanced dataset improves the class-wise performances of the model. A sufficiently high number of data samples for training and testing the ML models allowed under-sampling of the data to balance them, thereby increasing the clickbait class's performance. After balancing the dataset, the accuracy of the model was degraded from 84 to 79%, but the performances of the clickbait class were increased. Previously the precision, recall, and F_1 -score of the model were 0.77, 0.49, and 0.60; after balancing, it became 0.78, 0.84, and 0.81. There is a drastic increment in the recall value of the clickbait class, which means correctly identifying the maximum number of clickbait samples.

The proposed model is compared with several baseline models based on F_1 -score and accuracy, as indicated in Table 6. Based on the observations, it has been determined that the proposed approach surpasses both conventional manually designed feature engineering techniques and DL approaches used by Coste and Bufnea (2021), Dong et al. (2019), Kumar et al. (2018), Rajapaksha et al. (2021), Meng et al. (2022) and Zhou (2017). It was observed that in transfer learning (Rajapaksha et al. 2021), Gated-CNN (Meng et al. 2022), Siamese network (Kumar et al. 2018), and deep semantic similarity model (DSSM) (Dong et al. 2019) does not perform well and shows lower values for the F_1 -score, as shown in Table 6. The model by Dong et al. (2019) achieved the highest accuracy among all the reported work with a value of 0.86 (underlined value in the Acc. column of Table 6), but the model was only able to achieve an F_1

-score of 0.71 indicating that either the precision or recall value of their model was poor. The earlier best F1 score was 0.732 by Coste and Bufnea (2021), as shown by the underlined value in the F1 score column of Table 6. Our proposed model achieved an F_1 -score of 0.82 better than all the existing models and accuracy of 0.84 comparable to the existing models, as shown in bold in the ninth row of Table 6. It also shows that when dealing with imbalanced classes, the F_1 -score becomes very useful because it handles instances where either precision or recall could greatly impact the final decision.

To understand how the proposed model performs with the other dataset, the model was tested on another dataset referred to as *dataset 2*, which is maintained by Chakraborty et al. (2016). Dataset 2 has only the headlines, so the text embedding is generated for the headlines using SBERT. The embedding is fed into ML classifiers to identify whether the headlines are clickbait or non-clickbait. A comparative result of the proposed model with the state-of-the-art models for dataset 2 is given in the lower part of Table 6. Chakraborty et al. (2016) used 14 handcrafted features and reported an accuracy of 0.93 using SVM. Chawda et al. (2019) used pre-trained word2vec effectively to capture the contextual information and achieved an accuracy of 0.97, as shown in Table 6. Compared to the state-of-the-art models for detecting clickbait, the proposed model achieves the highest accuracy of 0.98, a weighted precision of 0.98, a weighted recall of 0.98, and a weighted F_1 -score of 0.98 using SVM.

The limitation of the proposed work is that the present model is validated with only an English language dataset. Since clickbaits are becoming common in other languages also, it will be interesting to see how the proposed model performs with other languages. The proposed model has also ignored the cases of code-mix language, which is currently a major challenge for NLP researchers. The current work is also limited by modalities as other than text, no other modalities such as images (post media), image captions, URLs, and target descriptions are considered for model development. The current work did not perform any Part Of Speech (POS) tagging or any other feature engineering, and it will be interesting to see some manual features along with the DL models.

7 Conclusion

This paper addresses clickbait detection by capturing essential information from clickbait news articles. A correlation matrix was used to examine the relationships between the sentences within the body content of the news. The existing conventional approach relies on feature engineering and similarity perspectives, which often fail to represent the information between titles and body content adequately. So,

the proposed model allowed us to determine only the minimum number of sentences from the body content of the news article required to detect clickbait. In this work, we used two methods (concatenation and averaging) for representing the embedding vector to make the feature vector for the clickbait classification. The model's performance increases to a certain extent by concatenating feature vectors. The model uses a correlation matrix to find the top six dissimilar sentences from the body content. The impact of distinct and dissimilar sentences on the classification was also observed. The experimental result demonstrates that our proposed model surpasses the current state-of-the-art methods, leading to a notable F_1 -score improvement of +8.8%. The proposed model uses only the English language dataset to train and test the model. In the future, our objective is to use multilingual and codemix datasets to train and test the model. The proposed model may be extended for visual-centric social media platforms like YouTube and Instagram, where clickbaits are becoming very common.

Acknowledgements We thank the data provider for helping us.

Author contributions All authors have contributed equally to the manuscript.

Funding Not applicable.

Data availability The dataset used in this study is available: (1) <https://zenodo.org/record/3346491>, and (2) <https://www.kaggle.com/datasets/amananandrai/clickbait-dataset>.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical approval Not applicable.

References

- Agrawal A (2016) Clickbait detection using deep learning. In: 2nd International conference on next generation computing technologies (NGCT), pp 268–272
- Al-Sarem M, Saeed F, Al-Mekhlafi ZG, Mohammed BA, Hadwan M, Al-Hadhrami T et al (2021) An improved multiple features and machine learning-based approach for detecting clickbait news on social networks. *Appl Sci* 11(20):9487
- Biyani P, Tsioutsoulouklis K, Blackmer J (2016) “8 Amazing secrets for getting more clicks”: detecting Clickbaits in news streams using article informality. In: AAAI Conference on artificial intelligence, vol 30, no 1, pp 94–100
- Bronakowski M, Al-khassaweneh M, Al-Bataineh A (2023) Automatic detection of clickbait headlines using semantic analysis and machine learning techniques. *Appl Sci* 13(4):2456
- Cao X, Le T, Zhang JY (2017) Machine learning based detection of clickbait posts in social media. [arXiv:1710.01977](https://arxiv.org/abs/1710.01977)
- Chakraborty A, Paranjape B, Kakarla S, Ganguly N (2016) Stop clickbait: detecting and preventing clickbaits in online news media. In:

- 2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), pp 9–16
- Chawda S, Patil A, Singh A, Save AM. (2019) A novel approach for clickbait detection. In: 2019 3rd International conference on trends in electronics and informatics (ICOEI), pp. 1318–1321
- Coste CI, Bufnea D (2021) Advances in clickbait and fake news detection using new language-independent strategies. *J Commun Softw Syst* 17(3):270–280
- Dong M, Yao L, Wang X, Benatallah B, Huang C (2019) Similarity-aware deep attentive model for clickbait detection. In: Pacific-Asia conference on knowledge discovery and data mining, vol 11440. Springer, Cham. https://doi.org/10.1007/978-3-030-16145-3_5
- Elyashar A, Bendahan J, Puzis R. (2022) Detecting clickbait in online social media: you won't believe how we did it. In: International symposium on cyber security, cryptology, and machine learning, vol 13301. Springer, Cham, pp 377–387. https://doi.org/10.1007/978-3-031-07689-3_28
- Geckil A, Mungen AA, Gündogan E, Kaya M (2018) A clickbait detection method on news sites. In: 2018 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), pp 932–937
- Glenski M, Ayton E, Arendt DL, Volkova S. (2017) Fishing for clickbaits in social images and texts with linguistically-infused neural network models. [arXiv:1710.06390](https://arxiv.org/abs/1710.06390)
- Gupta S, Sharaff A, Nagwani NK (2023a) Control stochastic selection-based biomedical text summarization using Sim-TLBO. *Arab J Sci Eng* 1–15
- Gupta S, Sharaff A, Nagwani NK (2023b) Frequent item-set mining and clustering based ranked biomedical text summarization. *J Supercomput* 79–1:139–159
- Indurthi V, Oota SR (2017) Clickbait detection using word embeddings. [arXiv preprint arXiv:1710.02861](https://arxiv.org/abs/1710.02861)
- Kaur S, Kumar P, Kumaraguru P (2020) Detecting clickbaits using two-phase hybrid CNN-LSTM biterm model. *Expert Syst Appl* 151:113350
- Klairith P, Tanachutiwat S (2018) Thai clickbait detection algorithms using natural language processing with machine learning techniques. In: 2018 International conference on engineering, applied sciences, and technology (ICEAST), pp 1–4
- Kumar V, Khattar D, Gairola S, Lal YK, Varma V. (2018) Identifying clickbait: a multi-strategy approach using neural networks. In: The 41st international ACM SIGIR conference on research & development in information retrieval, pp 1225–1228
- Kumar AP, Nayak A, Shenoy M (2019) A statistical approach to evaluate the efficiency and effectiveness of the machine learning algorithms analyzing sentiments. In: 2019 IEEE international conference on distributed computing, vlsi, electrical circuits and robotics (DISCOVER) pp 1–6
- Liu T, Yu K, Wang L, Zhang X, Zhou H, Wu X (2022) Clickbait detection on Wechat: a deep model integrating semantic and syntactic information. *Knowl Based Syst* 245:108605
- Loewenstein G (1994) The psychology of curiosity: a review and reinterpretation. *Psychol Bull* 116(1):75–98
- Ma YW, Chen JL, Chen LD, Huang YM (2022) Intelligent clickbait news detection system based on artificial intelligence and feature engineering. *IEEE Trans Eng Manag* 1–10
- Manjesh S, Kanakagiri T, Vaishak P, Chettiar V, Shobha GH (2017) Clickbait pattern detection and classification of news headlines using natural language processing. In: 2017 2nd International conference on computational systems and information technology for sustainable solution (CSITSS), pp 1–5
- Meng Q, Liu B, Sun X, Yan H, Liang C, Cao J et al (2022) Attention-fused deep relevancy matching network for clickbait detection. *IEEE Trans Comput Soc Syst* 1–12
- Naeem B, Khan AA, Beg MO, Mujtaba H (2020) A deep learning framework for clickbait detection on social area network using natural language cues. *J Comput Soc Sci* 3:231–243
- Papadopoulou O, Zampoglou M, Papadopoulos S, Kompatsiaris I (2017) A two-level classification approach for detecting clickbait posts using text-based features. [arXiv preprint arXiv:1710.08528](https://arxiv.org/abs/1710.08528)
- Pandey S, Kaur G (2018) Curious to click it?—Identifying clickbait using deep learning and evolutionary algorithm. In: 2018 International conference on advances in computing, communications and informatics (ICACCI), pp 1481–1487
- Patil S, Koul M, Chauhan H, Patil P (2021) Detecting and categorization of click baits. *Int J Eng Res Technol (IJERT)* 09(3):449–454
- Potthast M, Köpsel S, Stein B, Hagen M (2016) Clickbait detection. In: European conference on information retrieval. ECIR 2016, vol 9626. Springer, Berlin. https://doi.org/10.1007/978-3-319-30671-1_72
- Rajapaksha P, Farahbakhsh R, Crespi N (2021) BERT, XLNet or RoBERTa: the best transfer learning model to detect clickbaits. *IEEE Access*. 9:154704–154716. <https://doi.org/10.1109/ACCESS.2021.3128742>
- Razaque A, Alotaibi B, Alotaibi M, Hussain S, Alotaibi A, Jotsov VS (2022) Clickbait detection using deep recurrent neural network. *Appl Sci* 12(1):504. <https://doi.org/10.3390/app12010504>
- Reimers N, Gurevych I (2019) Sentence-BERT: Sentence Embeddings using Siamese BERT-networks. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp 3982–3992. <https://doi.org/10.18653/v1/D19-1410>
- Sisodia DS (2019) Ensemble learning approach for clickbait detection using article headline features. *Informing Sci International J Emerg Transdiscipline* 22:31–44. <https://doi.org/10.28945/4279>
- Srinivasarao U, Sharaff A (2023) Machine intelligence based hybrid classifier for spam detection and sentiment analysis of SMS messages. *Multimed Tools Appl* 82(20):1–31. <https://doi.org/10.1007/s11042-023-14641-5>
- Srinivasarao U, Sharaff A (2023) Spam email classification and sentiment analysis based on semantic similarity methods. *Int J Comput Sci Eng* 26(1):65–77
- Varshney D, Vishwakarma DK (2021) A unified approach for detection of Clickbait videos on YouTube using cognitive evidences. *Appl Intell (Dordrecht, Netherlands)* 51:4214–4235
- Zheng J, Yu K, Wu X (2021) A deep model based on lure and similarity for adaptive clickbait detection. *Knowl Based Syst* 214:106714
- Zhou Y (2017) Clickbait detection in tweets using self-attentive network. [arXiv:1710.05364](https://arxiv.org/abs/1710.05364)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.