applied sciences

MDPI

Article

AugFake-BERT: Handling Imbalance through Augmentation of Fake News Using BERT to Enhance the Performance of Fake News Classification

Ashfia Jannat Keya [1], Md. Anwar Hussen Wadud [1], M. F. Mridha [2,*], Mohammed Alatiyyah [3] and Md. Abdul Hamid [4]

[1] Department of Computer Science and Engineering, Bangladesh University of Business and Technology, Dhaka 1216, Bangladesh
[2] Department of Computer Science & Engineering, American International University of Bangladesh, Dhaka 1216, Bangladesh
[3] Department of Computer Science, College of Sciences and Humanities-Aflaj, Prince Sattam Bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia
[4] Department of Information Technology, Faculty of Computing and Information Technology, King AbdulAziz University, Jeddah 21589, Saudi Arabia
* Correspondence: firoz.mridha@aiub.edu

Abstract: Fake news detection techniques are a topic of interest due to the vast abundance of fake news data accessible via social media. The present fake news detection system performs satisfactorily on well-balanced data. However, when the dataset is biased, these models perform poorly. Additionally, manual labeling of fake news data is time-consuming, though we have enough fake news traversing the internet. Thus, we introduce a text augmentation technique with a Bidirectional Encoder Representation of Transformers (BERT) language model to generate an augmented dataset composed of synthetic fake data. The proposed approach overcomes the issue of minority class and performs the classification with the AugFake-BERT model (trained with an augmented dataset). The proposed strategy is evaluated with twelve different state-of-the-art models. The proposed model outperforms the existing models with an accuracy of 92.45%. Moreover, accuracy, precision, recall, and f1-score performance metrics are utilized to evaluate the proposed strategy and demonstrate that a balanced dataset significantly affects classification performance.

Keywords: text augmentation; transfer learning; Bidirectional Encoder Representation of Transformers; fake news classification

check for updates

Citation: Keya, A.J.; Wadud, M.A.H.; Mridha, M.F.; Alatiyyah, M.; Hamid, M.A. AugFake-BERT: Handling Imbalance through Augmentation of Fake News Using BERT to Enhance the Performance of Fake News Classification. *Appl. Sci.* **2022**, *12*, 8398. https://doi.org/10.3390/app12178398

Academic Editor: Yu-Dong Zhang

Received: 24 July 2022
Accepted: 18 August 2022
Published: 23 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A type of information known as "fake news" is either designed to delude readers or just a misguided viewpoint, and is distributed by printing in newspapers, online news portals, or social networks. Fake News is now a significant component of social media, raising questions about the information's authenticity, reliability, and honesty [1]. The effects of fake news are felt throughout society. For example, spreading false information on social media affects how organizations, governments, and individuals react to social and cultural events. The majority of fake news is directed at a specific demographic to instill strong beliefs and polarize society to promote a particular ideology. An enormous amount of fake news targets a specific demographic to inculcate strong beliefs and polarize society by promoting a particular agenda.

People are taking advantage of fake news even though fake news is unethical. For example, a newspaper called "The New York Sun" utilized a fake article to boost sales [2]. Such disinformation has the potential to have devastating and far-reaching consequences in many aspects of our lives, such as financial markets [3] and political events [4]. The dissemination of false information has an emotional impact on the public. It influences

people's decision-making [5]. Fake news is not a recent phenomenon, and yet it has grown increasingly troublesome in recent years due to its viral tendency on social media [6]. The number of people searching the internet for health-related news is continuously growing. Health-related fake news affects people's lives [7]. During the COVID-19 pandemic, false information and offensive text were rapidly propagated, which misled people [8,9].

False information spreads quickly on social media in the modern era, raising societal concerns about people's ability to differentiate between what is phony and what is genuine while browsing and actively using social media. As a result, one of the rising study topics is automatic false news identification. Numerous approaches for automatically detecting the authenticity of news have been developed. Initially, Natural Language Processing (NLP) issues were handled using traditional Machine Learning (ML) methods such as Logistic Regression and Support Vector Machine (SVM) with hand-crafted features [10,11]. These approaches inevitably produced high-dimensional interpretations of language processing, giving rise to the curse of dimensionality. Compared to traditional ML models, models based on neural networks achieved extraordinary success on many tasks involving natural language thanks to the use of word embeddings [12]. Word embeddings are low-dimensional with distributed feature representations suited for natural languages. Neural network-based approaches widely use Word2vec and GloVe word embedding models for context-free word embedding [13–15].

Deep Learning (DL) methods such as Recurrent Neural Networks (RNN) and their variations along with Convolutional Neural Networks (CNN) have demonstrated outstanding performance in false news identification tasks [16–18]. Saleh et al. [19] proposed an optimized CNN model that outperformed the baseline models when they compared their model to RNN and ML classifiers. The ensemble approach with DL models is widely used in this domain [20]. Most studies to date have used ensemble models with CNN and RNN. CNN is used for extracting valuable features, and long-short-term memory (LSTM) is utilized for classification [13,21]. Opportunities remain to improve the performance of DL-based models. While CNNs and RNNs perform well with context-free features, textual content features with contextual information provide a better representation of words and yield better classification results. Recently, different language models have gained popularity in different NLP tasks [22–24]. A few studies have used embeddings from language model (ELMO), such as Bidirectional Encoder Representations from Transformers (BERT), that have outperformed several baseline methods in fake news detection [25,26]. BERT employs semi-supervised learning and a language representation model that only uses the transformer's encoder [27]. However, these DL-based approaches are language-dependent and mostly rely on text-based features of English news [7]. A study by Wei and Zou [28] proposed easy data augmentation (EDA) techniques for boosting performance on text classification tasks. EDA consists of four simple yet powerful operations: synonym replacement, random insertion, random swap, and random deletion. Another study by Rizos et al. [29] proposed three text augmentation methods: synonym replacement, warping the tokens of words, and language generation with RNN. Text generation is performed by Hiriyannaiah et al. [30] with sequence GAN. However, these methods only work well for short texts.

Bengali is the world's seventh most popular language. Two hundred and sixty-eight million people speak this language in Bangladesh and parts of India [31]. Thanks to the increase in the use of electronic devices, digital Bengali text content has increased dramatically in recent years on the internet and news portals. However, scientific investigation into Bengali Language Processing (BLP) remains in its early stages [32]. A study by Al-Zaman [33] conducted a study that examined 419 bogus news articles published in India, a country prone to fake news, to identify the primary themes, types of content, and sources of fake news on social media. Fake news circulating on social media focuses on six main topics: health, politics, crime, entertainment, and miscellaneous. Health-related fake news is more prevalent during health crises, such as when misleading online news about COVID-19

endangered the health of Indians [33]. In contrast, fake news about politics and religion is exceptionally prevalent and originates from online media [34].

Moreover, a lack of benchmark datasets of Bengali fake news is a significant issue in this research area. The imbalanced data on the accessible dataset is another significant problem in this field. A text augmentation strategy is required for this purpose, as manual labeling of news as fake or true is time-consuming. Because the BERT model was initially trained using the English language, it is unsatisfactory when utilized to classify Bengali news using transfer learning. Existing research on the Bengali language has mainly used pre-trained BERT for Bengali news classification, which has not yielded satisfactory results [35]. As a result, a system that augments the text and trains the BERT model from scratch may work well for Bengali. We determined that an effective fake news detection strategy with sufficient Bengali text is required. Therefore, we propose a classification mechanism with text augmentation that can classify Bengali news as fake or true. This model can be used by anyone willing to incorporate artificial intelligence into their systems to detect fake news. The system can help social media users differentiate between fake and true news. As a result, users can avoid being misled by the spread of fake news. The overall contribution of this paper includes:

- We propose AugFake-BERT, which integrates data augmentation with transfer learning to generate synthetic texts to supplement our fake news dataset. This study is the first to use a transfer learning approach to deal with imbalanced text data.
- We trained a miniature BERT model from scratch with our original and augmented dataset to extract contextual features and perform the classification with a softmax layer.
- We experimented with various BERT parameters to find the best-fit parameters for the Bengali fake news data.
- Different baseline machine learning (ML) and deep learning (DL) models were used to evaluate the proposed strategy.

The paper is structured as the follows: Section 2 highlights the related work in this domain; Section 3 illustrates the proposed methodology; and we discuss the results of our experiments in Section 4. We mention the limitations of this study and potential future work in Section 5. Finally, Section 6 brings the paper to a conclusion.

2. Literature Review

Disinformation is difficult to detect because it is purposefully written to misrepresent information. The major barrier to accurately identifying fake news is the scarcity of human news labeling. Numerous research studies already exist on false news detection utilizing various classification methods. It is critical to pick the proper feature reduction algorithm, as feature reduction has a significant impact on text classification performance [36]. For this reason, traditional machine learning-based methods perform poorly on high-dimensional features [37]. For the most part, neural network-based approaches have performed better than traditional machine learning-based methods, as they reduce the curse of dimensionality. Several studies have used CNN for extracting useful features, and long short-term memory (LSTM) has been utilized for classification [13,21]. A neural network's performance is significantly influenced by hyperparameter optimization. Saleh et al. [19] proposed an optimized CNN (OPCNN-FAKE) model to detect fake news. Their findings indicated that CNN with optimized parameters outperformed the baseline models when they compared their model to RNN and ML classifiers.

Attention techniques have recently been used to efficiently obtain information from news content associated with a mini query [38,39]. A study by Singhania et al. [38] utilized a three-level hierarchical attention network, 3HAN, which has three levels: one for words, one for sentences, and one for the headline. 3HAN can allocate weights to distinct sections of an article thanks to its multiple levels of attention. In comparison to other deep learning models, 3HAN produces more accurate results despite the fact that it only uses textual data. However, the studies mentioned earlier that employ neural networks mainly use

non-contextual embedding, which does not consider the contextual meaning of the text, and has thus yielded minimal success.

Furthermore, the majority of models fail to achieve satisfactory detection performance. To compensate for the shortcomings of previous works, Aloshban [40] suggested self-attention-based Automatic Fake News Classification (ACT). Their principle is derived from the fact that texts are typically short and thus inefficient for classification. Their proposed framework employs mutual interactions between a claim and numerous supporting replies. While the article input is fed into the LSTM, the final step of the LSTM output does not always entirely reflect the semantics of the article. By connecting all word vectors in the text, a massive vector dimension can be obtained. Thus, the internal correlation between the words in the articles can be overlooked.

As a consequence of using the self-attention process with the LSTM model, essential components of the article are extracted using several feature vectors. Their approach is highly dependent on self-attention and a representation matrix of articles. Graph-aware Co-Attention Networks (GCAN) is a relatively new method to detect fake news [41]. GCAN can decide whether a short-text Twitter post is fraudulent based on the historical record of its retweeters. This model, however, is not appropriate for long texts.

Another notable advancement in the NLP field is language models, which produce contextual embeddings. Bidirectional Encoder Representation (BERT) is mainly used in fake news detection tasks. Jwa et al. [26] included news data in BERT's pre-training phase to improve fake news detection performance. ExBAKE (BERT with additional unlabeled news text data) outperformed the state-of-the-art model stackLSTM by a 0.137 F1 score. Zhang et al. [42] demonstrated a BERT-based Domain-Adaption Neural Network (BDANN) for multimodal fake news detection. The significant parts of BDANN are a multimodal feature extractor, a false news detector, and a domain classifier. The pre-trained BERT method was utilized to extract text features, while the pre-trained VGG-19 (a 19-layer deep CNN) model was used to extract visual features. The extracted features are merged and sent to the detector, which uses them to distinguish between fake and real news. Furthermore, the presence of corrupted image data in the Weibo dataset influenced the results of BDANN. Kaliyar et al. [43] suggested a fake news detection mechanism with BERT based on a deep convolutional approach (fakeBERT). This fakeBERT is made up of various parallel blocks of a deep CNN, which is a one-dimensional (1D-CNN) with varying kernel sizes and filters, as well as the BERT. Various filters can extract valuable data from the training sample. BERT combined with 1D-CNN can handle both large-scale structured and unstructured text. As a result, the combination is advantageous in dealing with ambiguity. Wu et al. [44] proposed a multi-modal framework that combines RNN and Graph Convolutional Network (GCN) for encoding fake news propagation in social media. The limited data on fake news is a significant issue in the NLP field. Thus, a few studies have tried to generate synthetic data to solve this problem. A study by Hiriyannaiah et al. [30] presented Sequence Generative Adversarial Network (SeqGAN), which is a GAN structure that solves the gradient descent problem in GANs for discrete outputs using a reinforcement learning (RL)-oriented approach and Monte Carlo search. The authors fed actual news content into the GAN, then a classifier based on Google's BERT model was trained to distinguish actual samples from produced samples. However, seqGAN while can produce short text, it is not a better solution for generating entire news content with seqGAN, and may result in non-contextual news. The authors used DistilBERT to encode the representation of textual data such as fake news, tweets, and user self-descriptions. Furthermore, the authors improved their performance by employing an attention mechanism.

Limited data on fake news is a significant issue in the NLP field. Thus, a few studies have tried to generate synthetic data to solve this problem. A study by Hiriyannaiah et al. [30] presented seqGAN, which can only produce short text; it is not a better solution for generating entire news content with seqGAN, and may result in non-contextual news. Most research efforts have focused on English data. Other low-resource language-related fake news detection

mechanisms are scarce in this domain. Hossain et al. [35] published a Bangla dataset consisting of fake news and true news and performed classification with CNN, LSTM, and BERT. The authors classified news documents using the pre-trained multilingual BERT model. The multilingual BERT model is trained in 104 different languages. However, the model is not explicitly trained to have shared representations across languages. Thus, we propose a technique where we utilize the pre-trained multilingual BERT to generate synthetic news. Furthermore, we trained a smaller BERT from scratch for embedding and classification, and experimented with using a frozen BERT with our training samples.

3. Proposed Methodology

The fake news detection system's general architecture comprises two parts, embedding and classification. The embedding technique uses unlabeled texts to generate an embedding model, while the classification method uses labeled texts to derive a classification model. Figure 1 depicts the overall process of our proposed strategy. The most recent research papers focus on data augmentation and embedding systems. The data augmentation model is trained based on transfer learning. The embedding model is trained from scratch with the augmented data. The dataset description that we used in this study is demonstrated in Section 3.1. Section 3.2 describes the basic framework of the augmentation method built using multilingual BERT. Section 3.3 explains the basic framework of our embedding model.

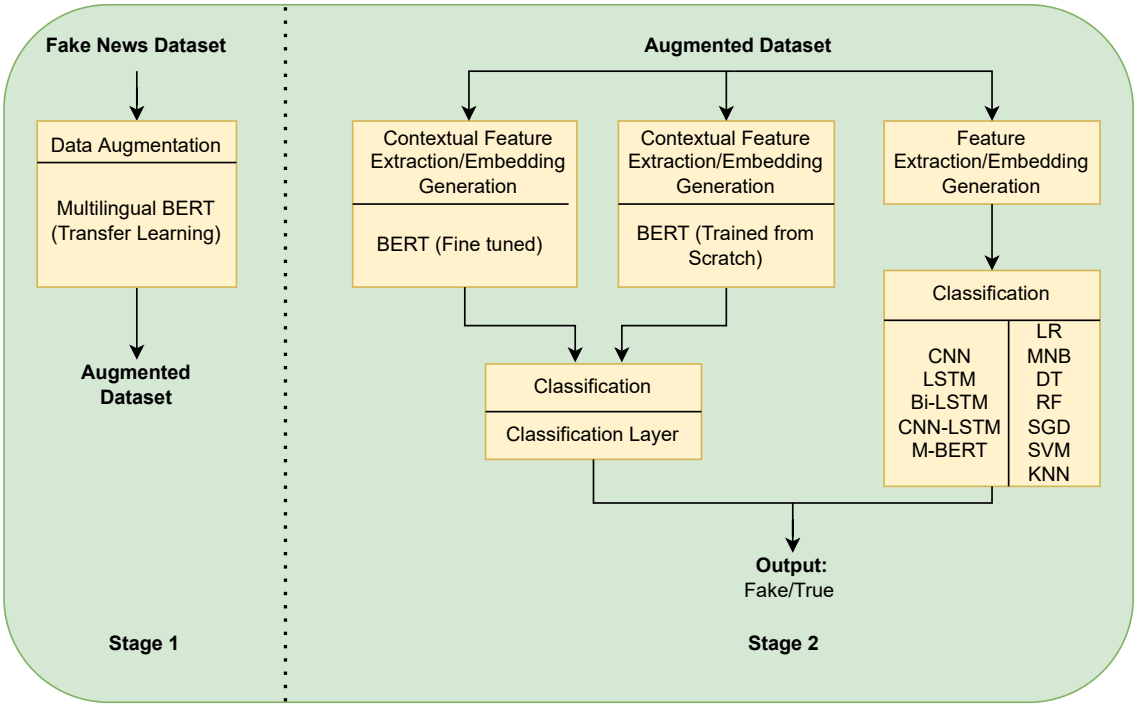
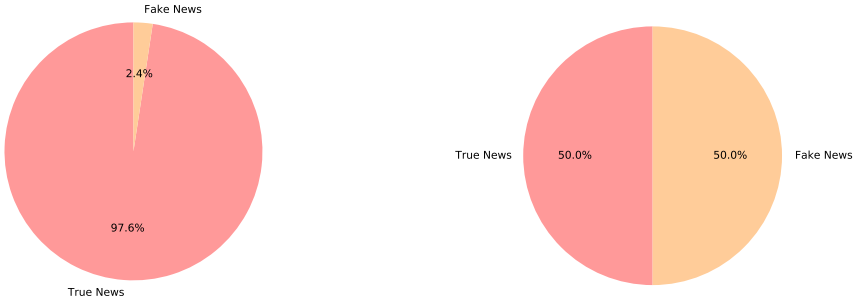


Figure 1. An abstract view of our fake news detection mechanism.

3.1. Dataset Description

We utilized the "BanFakeNews" dataset, (https://www.kaggle.com/cryptexcode/banfakenews) (accessed on 20 February 2022), which is available publicly on Kaggle. The "BanFakeNews" dataset [35] contains approximately 50,000 Bengali news items. It is presented as a CSV file. The dataset contains seven attributes in total: articleID, domain, date, category, headline, content, and label. For this investigation, we considered headline and content-based features. The dataset is highly imbalanced, as the fake news class data are less common than the true news class data. As previously stated, due to the lack of a reliable, huge, and up-to-date dataset in Bangla, we performed a data augmentation technique to balance the dataset. Figure 2 presents the ratio of fake news and true news data.



(**a**) (**b**)

Figure 2. The amount of fake and true news in both the balanced (augmented) and imbalanced (original) datasets. (**a**) Imbalanced dataset; (**b**) Balanced dataset.

3.2. Data Augmentation through Transfer Learning

Data augmentation techniques have been used in image processing, visual identification, and text classification projects because they make it easy to collect and generate new data using simple and quick transformations. The goal of augmentation is to increase the quantity of training data samples to reduce the model's overfitting. The text augmentation NLP sector is challenging because of the immense complexity of language. Several techniques are available, including Easy Data Augmentation (EDA), replacement with a synonym, random swap, random insertion, and random deletion. Certain words, however, such as a, an, and the, cannot be filled by others. In addition, often a term does not have a synonym. Even if someone changes a single word, the context changes dramatically. SeqGAN [45], a sequence generation method, is being explored for synthetic text generation, although it is only suitable for generating short text. As we needed news content consisting of a large amount of text to create fake news, this model was unsuitable for our data augmentation technique.

We had 1300 fake news data instances collected from the "BanFakeNews" dataset. To increase the amount of fake news data, we adopted a process that would consume less time and effort than manually collecting the fake news data. We employed transfer learning to facilitate data collection more efficiently. The idea behind transfer learning is that a method trained for one endeavor is re-purposed for a second related task. We used the BERT multilingual uncased model, which is self-supervised and trained on a massive corpus of multilingual data. This means it was trained solely on raw texts, with no human labeling (hence its ability to use a large amount of publicly available data), and then used an automatic strategy with these texts to produce input data and labels. It was explicitly configured with two goals described below in mind.

Masked Language Modeling (MLM): given a sentence, the method masks 15% of the words in the input and then runs the whole masked text through the model, which must estimate the masked words. This is in contrast to traditional RNNs, which typically see words sequentially, or auto-regressive models such as Generative Pre-trained Transformer (GPT), that internally mask prospective tokens. It enables the model to learn a two-way representation of the sentence. Next Sentence Prediction (NSP): During pre-training, the model merges two masked texts as inputs. They sometimes, though not always, correspond to sentences which were next to one another in the original text. The model must then predict whether or not the two sentences followed each other.

We performed next word prediction instead of next sentence prediction, which allowed us to perform insertion by prediction, whereas previous word embedding models pick one word randomly for insertion. The same word in sentences is represented by a static vector in traditional word embedding models, which might not work in certain circumstances. To address this issue, contextualized word embedding was introduced to take into account surrounding words when generating a vector in a different context. Thus, for substitution, we considered surrounding words in order to predict the target word. Figure 3 presents the data insertion and substitution performed on a training sample.

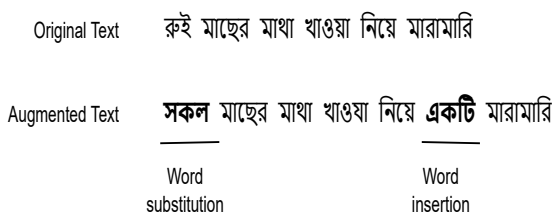


Figure 3. The original text from the training data and the augmented data using insertion and substitution of words.

3.3. Fine-Tuning BERT for Embedding and Classification

The deep learning-based method BERT has demonstrated cutting-edge performance in a wide range of natural language processing tasks. BERT integrates language representations trained by Google. BERT is an advanced pre-trained word-embedding model based on a transformer-encoded architecture [27]. BERT is trained with a large corpus of texts, which allows the model to better grasp the vocabulary and learn variability in patterns of data as well as to generalize across a variety of NLP tasks. Because it is bidirectional, BERT understands information from both the left and right sides of a token's context during training. The BERT method is notable for its ability to recognize and capture contextual meaning from a text [46]. The main limitation of traditional language models is that these models are unidirectional, limiting the architectures that can be used during pre-training. Existing research employs pre-trained language models previously trained on data other than the relevant data. which is ineffective in resolving the problem at hand [35]. However, such models have the potential to perform well if trained with additional relevant problem data. Thus, we trained BERT with our additional news corpora in order to obtain better results.

Figure 4 illustrates the architecture of our proposed model. By employing a mask language model, the BERT model eradicates unidirectional constraints (MLM). As we have already mentioned, BERT uses the next sentence prediction (NSP) task in conjunction with the masked language model to concurrently pre-train text-pair representations. The encoder in the base model comprises 12 layers. Token embedding, position embedding, and segment embedding are used to represent input sequences to BERT. Token embedding involves tokenizing and embedding the input data in a method known as WordPiece embedding. BERT can take one or two sentences as input and distinguish them using the special token (SEP). The (CLS) token, which is distinctive to classification tasks, always appears at the beginning of the text. In English, WordPiece is more than just separating words by space; it can be viewed as a specialized tool for sentence tokenization and embedding. WordPiece embedding has a vocabulary size of 30K, so the form of the embedding size is $V * H$, where H symbolizes the hidden size and V is the vocabulary size (typically, we selected an embedding size equal to the hidden size). Segment embedding illustrates the connection between two sentences. Figure 5 presents the word embedding of our BERT model. After performing contextual embedding generation with BERT, a softmax classification layer was used to classify the training data into fake or true news. A detailed training description is elaborated in Section 4.

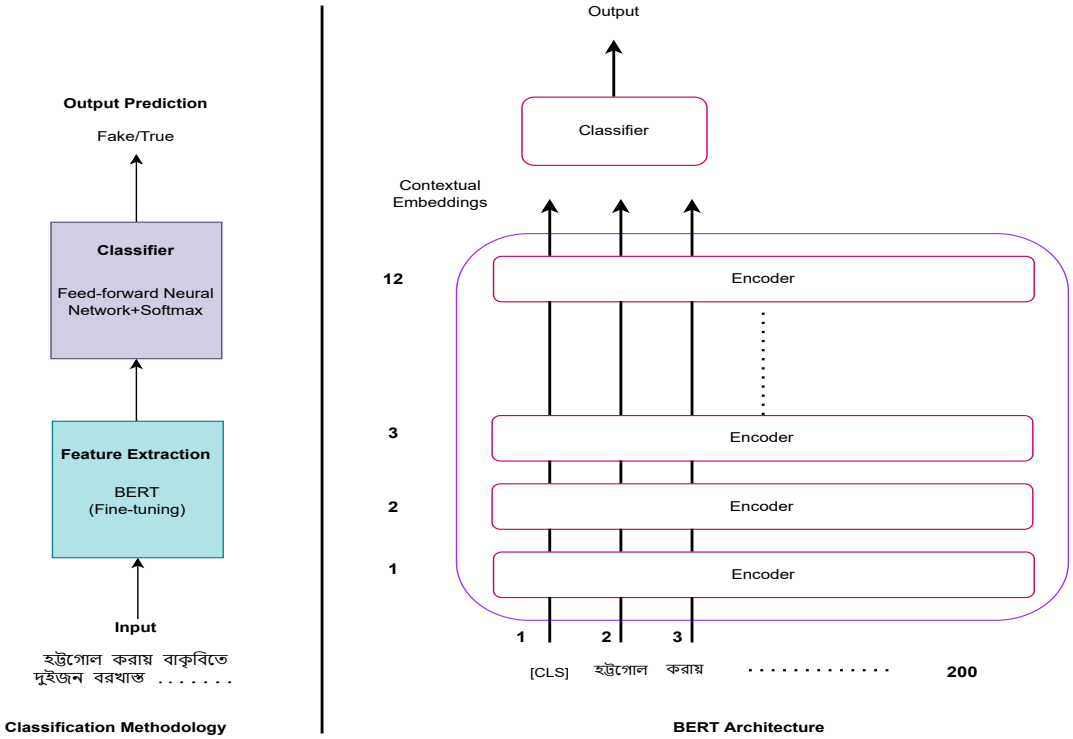


Figure 4. The proposed model architecture.

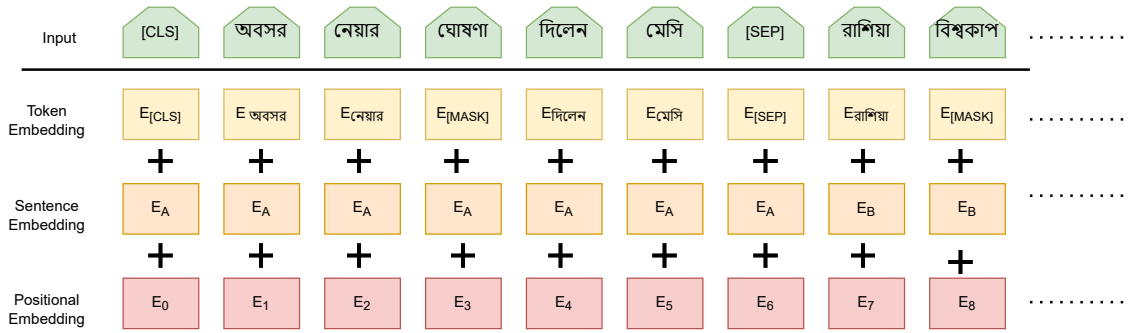


Figure 5. The word embedding of our input corpus with BERT .

4. Experiment and Result

We performed several experiments to achieve a better outcome. The experiments are illustrated below:

4.1. Augmentation

We utilized a transfer learning-based approach for text augmentation. We used a pre-trained multilingual BERT base uncased for generating synthetic data through insertion and substitution of words. The model was originallhy developed for generating contextual word embeddings. We modified the BERT model for performing text augmentation. The pre-trained model's weights were frozen for performing the task. The multilingual BERT incorporates masked language modeling and next sentence prediction tasks for generating embeddings. For text augmentation, we performed next word prediction instead of next sentence prediction. We masked the word that needed to be replaced with another word, and our trained model then predicted the masked word utilizing the vectors of surrounding words, which resulted in the generation of augmented texts. We fed the model one sample at a time containing the news content as a list of tokens. For the substitution of words, the BERT model masked 15% of the words of the news content, then these masked words

were replaced with the predicted words. For the insertion of words, the model inserted the predicted word before the masked word. After performing these processes, the result was synthetic news content which was produced by our model. By repeating this process for 1299 fake news content items from the original dataset, we generated 2700 fake news content items, for a total number of 4000 fake news data items. We undersampled the authentic news data to produce a balanced augmented dataset comprising an equal amount of data in both classes. Table 1 describes the original dataset and augmented dataset. Twelve different classification algorithms were used on the created dataset to examine how well they worked for fake news classification.

Table 1. Description of the datasets.

Dataset	True News	Fake News	Total
BanFakeNews	48,678	1299	49,977
Augmented Dataset	4000	4000	8000

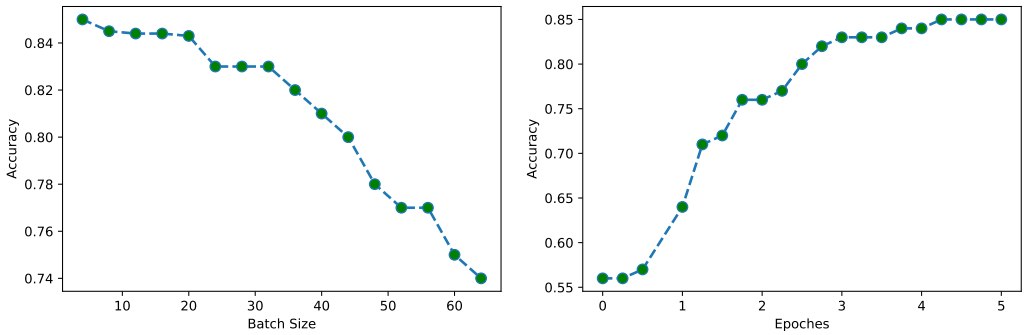
4.2. Embedding and Classification

We imported the BERT tokenizer and BERT base uncased from Huggingface's transformers library. The original and augmented datasets were then prepared with a BERT tokenizer in the format required by the BERT model. The encoder in the base model comprises 12 layers and 110 million parameters. We used the BERT base uncased model by frozen and unfrozen weights for training with our news corpora. We named our proposed model AugFake-BERT. The suggested AugFake-BERT classifier model was assigned a number of parameter values, which are provided in Table 2, to achieve optimized hyperparameters.

Table 2. Initial hyperparameters adapted for AugFake-BERT.

Parameters	Parameters Value
Maximum Sequence Length	128
Epochs	2
Batch Size	32
Activation	ReLU, SoftMax
Learning Rate	1×10^{-3}
Optimizer	AdamW

For AugFAKE-BERT, we adapted a set of values for diverse parameters. The maximum sequence length was from 128 to 50, the learning rate from 1×10^{-3} to 2×10^{-3}, and the batch size from 5 to 64. The batch size describes how many samples are sent to the algorithm for extraction at a specific iteration. The classifier performance seen in Figure 6a is significantly impacted by the batch-wise approach, which boosts GPU performance. For the frozen model, the pre-trained BERT base uncased model's weights were frozen in all the layers except the output layer, which was fine-tuned for news classification. For the unfrozen model, the model was trained from scratch with our news corpora. The data were classified using a feed-forward neural network layer utilizing a softmax activation function. As our optimizer, we chose AdamW, a more effective variation of the Adam optimizer. As more epochs results in overfitting issues and requires a longer training time for the algorithm, the number of epochs was fixed to four.



(**a**) (**b**)

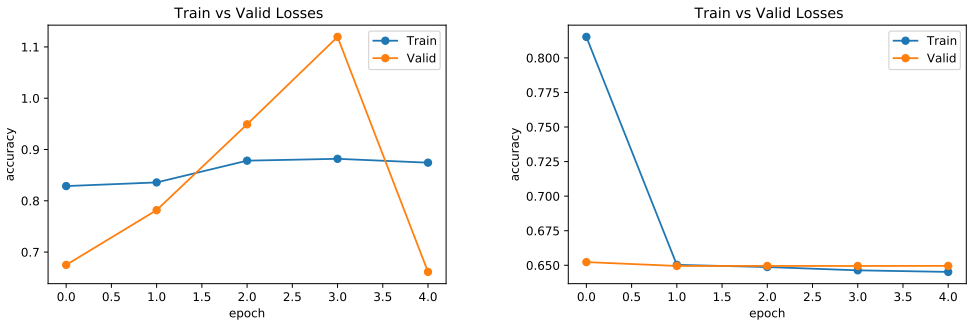
Figure 6. The impact on accuracy of the number of batches and epochs: (**a**) The impact of batch size on accuracy and (**b**) The impact of the number of epochs on accuracy.

Large batch sizes slow down training by consuming more GPU memory. In other words, GPU memory plays a significant role in the processing of diverse batch sizes. Because of the reasonable accuracy observed, the batch size was set at fifteen for all assessments. A small batch size enhances the outcomes, while a large batch size uses more memory, takes longer to analyze data, and reduces overall accuracy. Dropout is a simple and direct mathematical operation used to protect the classification algorithm from noisy features and overfitting. Because of the small dataset used for training, the classifier may be prone to overfitting. This issue can be fixed by increasing the dataset quantity or reducing the number of hidden layers. This issue was resolved by our enhanced dataset. Dropout removes or disables detached units in the hidden neuron that do not participate in subsequent iterations of the computation. The learning rate determines the best termination time for classification models. A high learning rate causes the model to be terminated quickly, whereas a low learning rate causes the model to be terminated slowly. At a learning rate of 1×10^{-3}, the accuracy goes up to 85.25%, which is the maximum. The optimized parameters that achieved maximum accuracy are illustrated in Table 3.

Table 3. Optimized hyperparameters adapted for AugFake-BERT.

Parameters	Parameters Value
Maximum Sequence Length	80
Epochs	5
Batch Size	15
Activation	ReLU, SoftMax
Learning Rate	2×10^{-3}
Optimizer	AdamW

To compute the loss of the AugFake-BERT model, the cross-entropy loss function was used. The loss shows the total number of errors in our model to evaluate how well (or poorly) the model performs. The training and validation loss of the frozen and unfrozen models are shown in Figure 7. Figure 7 shows that the AugFake-BERT model has a high loss value when the weights are frozen, indicating that it is unable to correctly predict the class in new test data in the original dataset due to imbalanced data. Thus, we further trained the BERT model from scratch with our augmented dataset. The results on the augmented dataset are illustrated in Section 4.4.



(**a**) (**b**)

Figure 7. The loss (during training and validation) of both frozen and unfrozen AugFake-BERT models on the original dataset. (**a**) Loss of BERT (Frozen) Model; (**b**) Loss of BERT (Unfrozen) Model.

4.3. Evaluation Metrics

We considered a number of performance metrics to assess our model. We used Fasttext, Word2Vec, and different N-grams (Unigram (U), Bigram (B), Trigram (T), UBT (U+B+T), character 3-gram (C3Gram), character 4-gram (C4Gram), and character 5-gram (C5Gram)) with TF-IDF word embedding methods on the benchmark dataset to find the different features and compare the proposed model with different baseline models. To evaluate the baseline model, we randomly selected 80% of samples for training and 20% of samples for testing purposes. In addition, we used ten-fold cross-validation and chose the best performance results. The confusion matrix depicts a summary of the classification results on the testing dataset based on known true values. It provides a summary of the model's success as well as effective results for true positive, false positive, true negative, and false negative. The proposed model is evaluated using accuracy (A), precision (P), recall (R), and F1-score (F).

Accuracy: The classification accuracy rating, known as the accuracy score, is calculated as the percentage of accurate predictions to the total number of predictions. Here, we define accuracy (A) using the formula in Equation (1):

$$\mathrm{A} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total Number of Predictions}} \tag{1}$$

Precision: Precision is defined as the number of true positive outcomes divided by the total positive outcomes, including those that are misidentified (P). Equation (2) is used to calculate precision:

$$\mathrm{P} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{2}$$

Recall: The recall is measured as the proportion of true positive results to the actual positive samples that should have been detected. The recall is computed using Equation (3):

$$\mathrm{R} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{3}$$

F1-score: The F1-score determines the model's accuracy in each class. The F1-score metric is usually used when the dataset is imbalanced. Here, we use the F1-score as an evaluation metric to demonstrate the adequacy of our proposed method. The F1-score is computed using Equation (4):

$$\mathrm{F1} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

4.4. Evaluation Baseline

The proposed model was compared with different baseline deep learning-based models, Convolutional Neural Network (CNN), Long Short Term Memory Network (LSTM) Bidirectional LSTM (Bi-LSTM), ensemble model CNN-LSTM, pre-trained Multilingual BERT (M-BERT) Model, and machine-learning classifiers such as Decision Tree (DT), Logistic regression (LR), Random Forest (RF), Multinomial Naive Bayes (MNB), Stochastic Gradient Descent (SGD), Support Vector Machine (SVM), and K-Nearest Neighbor (KNN), all of which were trained on the augmented dataset of Bengali news. The following sections provide an overview of these models' performance.

4.4.1. Machine Learning Classifiers

The results of the cross-validation and testing performance of different ML classifiers on our augmented dataset are illustrated in Table 4–10. In cross-validation, the LR and MNB model achieved the best accuracy of 86.44% and 86.16% for the Word2Vec feature extraction method. In the LR model, Character 5-gram (C5Gram) feature has the best precision (86.09%), and the best recall is 96.42% for character 3-gram (C3Gram). In the MNB classifier, C5Gram has the best precision (95.79%), while Unigram has the best recall (96.65%). The best F1-Score for both classifiers, LR and MNB, are 89.94% and 87.89% for the combination of Unigram (U), Bigram (B) and Trigram (T) (UBT) with the IF-IDF feature extraction model.

Table 4. Performance of logistic regression (LR) machine learning classifier.

	Cross-Validation Performance				Test Performance			
Features	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**
Unigram	85.27	81.63	94.31	87.52	84.10	79.11	94.58	86.16
Bigram	84.16	81.10	92.19	86.29	77.82	72.42	93.45	81.60
Trigram	82.27	80.00	90.80	85.06	76.71	71.43	93.39	80.95
UBT	88.66	85.00	95.50	89.94	86.21	81.62	94.62	87.64
C3Gram	88.55	84.21	96.42	89.90	81.21	75.63	93.62	83.67
C4Gram	85.77	85.70	88.83	87.24	68.32	62.28	95.00	76.75
C5Gram	85.71	86.09	87.77	86.92	74.60	67.45	95.70	80.56
Fasttext	85.94	84.36	91.70	87.88	81.49	74.11	96.06	85.13
Word2Vec	86.44	85.95	89.01	87.45	82.82	75.04	96.59	85.74

Table 5. Performance of Multinomial Naive Bayes (MNB) machine learning classifier.

	Cross-Validation Performance				Test Performance			
Features	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**
Unigram	81.99	76.59	96.65	85.46	79.82	73.57	95.86	83.25
Bigram	51.81	93.44	11.72	20.82	50.86	54.90	37.28	44.40
Trigram	78.82	75.02	92.80	82.97	74.87	68.91	95.80	80.16
UBT	85.88	80.66	96.54	87.89	79.49	73.65	93.86	82.54
C3Gram	85.66	85.54	87.70	86.60	81.66	80.39	85.08	82.67
C4Gram	80.66	78.16	89.75	83.55	79.49	72.66	97.45	83.25
C5Gram	84.60	95.79	74.82	84.02	74.87	67.69	99.56	80.73
Fasttext	82.99	92.21	75.80	83.21	79.32	71.92	97.76	83.67
Word2Vec	86.16	83.49	92.15	87.61	82.05	74.44	99.35	85.11

Table 6. Performance of Decision Tree (DT) machine learning classifier.

	Cross-Validation Performance				Test Performance			
Features	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**
Unigram	82.38	78.50	93.40	85.30	79.77	74.81	92.45	82.70
Bigram	81.05	77.38	91.78	83.97	75.93	70.36	93.77	80.40
Trigram	84.88	83.21	91.20	87.02	74.54	70.14	90.45	79.01
UBT	87.88	85.26	93.30	89.10	80.77	77.89	87.62	82.47
C3Gram	86.66	83.38	93.38	88.10	77.93	73.40	89.51	80.66
C4Gram	91.11	88.09	96.85	92.26	80.38	74.02	96.28	83.70
C5Gram	81.05	77.38	91.78	83.97	78.27	72.64	94.19	82.02
Fasttext	85.27	83.68	91.30	87.33	80.04	74.38	95.07	83.46
Word2Vec	90.27	88.09	94.45	91.16	77.15	71.19	93.65	80.89

Table 7. Performance of Random Forest (RF) machine learning classifier.

	Cross-Validation Performance				Test Performance			
Features	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**
Unigram	87.27	87.43	89.64	88.52	81.16	77.22	90.75	83.44
Bigram	87.72	92.53	84.07	88.10	80.71	91.67	69.69	79.18
Trigram	85.05	87.11	85.80	86.45	78.88	73.31	94.54	82.58
UBT	87.33	83.14	95.50	88.89	83.44	78.60	93.33	85.33
C3Gram	88.44	84.88	95.06	89.68	81.55	76.54	92.43	83.74
C4Gram	92.22	92.63	93.20	92.91	82.21	76.07	96.28	84.99
C5Gram	87.72	92.53	84.07	88.10	81.71	93.40	70.22	80.17
Fasttext	87.49	88.87	88.60	88.73	79.60	73.07	97.38	83.49
Word2Vec	90.66	87.44	96.23	91.63	78.60	71.90	96.12	82.27

Tables 4–10 display the results of the cross-validation and performance testing of different baseline models on the benchmark dataset. The LR model achieved the best accuracy of 86.44% for the Word2Vec feature extraction method, character 5-gram (C5Gram) feature has the best precision of 86.09%, the best recall is 96.42% for character 3-gram (C3Gram), and the best F1-Score is 89.94% for the combination of Unigram (U), Bigram (B), and Trigram (T) (UBT) with the IF-IDF feature extraction model in cross-validation performance.

Table 8. Performance of Stochastic Gradient Descent (SGD) machine learning classifier.

	Cross-Validation Performance				Test Performance			
Features	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**
Unigram	82.99	79.04	93.81	85.79	78.60	72.68	94.69	82.23
Bigram	83.60	79.89	93.11	86.00	77.10	71.52	93.88	81.19
Trigram	82.10	79.79	90.80	84.94	77.10	71.55	94.23	81.34
UBT	88.49	84.44	96.02	89.86	83.05	77.96	93.65	85.09
C3Gram	86.27	87.05	86.96	87.01	83.38	81.74	87.14	84.35
C4Gram	81.21	81.75	84.57	83.13	62.53	58.70	95.75	72.78

Table 8. *Cont.*

	Cross-Validation Performance				Test Performance			
Features	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**
C5Gram	85.66	86.29	87.36	86.82	80.04	72.51	86.98	84.07
Fasttext	84.38	84.94	87.40	86.15	79.32	71.92	98.56	83.67
Word2Vec	86.77	85.89	89.84	87.82	81.88	74.02	99.43	85.07

Table 9. Performance of Support Vector Machine (SVM) machine learning classifier.

	Cross-Validation Performance				Test Performance			
Features	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**
Unigram	86.21	82.76	94.52	88.25	84.94	80.23	94.47	86.77
Bigram	84.66	80.98	93.63	86.84	78.65	72.87	94.72	82.37
Trigram	82.49	79.81	91.70	85.34	77.71	72.08	94.54	81.80
UBT	88.99	85.41	95.60	90.22	83.21	78.07	93.86	85.24
C3Gram	88.99	84.38	97.16	90.32	81.49	75.83	93.95	83.92
C4Gram	89.66	86.55	96.04	91.05	77.32	70.30	98.09	81.90
C5Gram	84.66	80.98	93.63	86.84	79.77	73.34	96.73	83.42
Fasttext	85.33	82.68	93.10	87.58	81.38	75.20	96.75	84.63
Word2Vec	90.99	87.87	96.34	91.91	77.71	70.72	96.99	81.80

Table 10. Performance of K-Nearest Neighbor (KNN) machine learning classifier.

	Cross-Validation Performance				Test Performance			
Features	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**	**A (%)**	**P (%)**	**R (%)**	**F1-S (%)**
Unigram	84.44	82.85	90.25	86.39	82.88	79.72	90.22	84.65
Bigram	81.10	83.63	80.88	82.24	75.15	72.44	85.22	78.31
Trigram	78.43	79.94	81.70	80.81	76.10	74.42	83.63	78.75
UBT	85.21	84.83	87.85	86.32	81.60	79.66	86.44	82.91
C3Gram	87.44	87.56	88.85	88.20	79.16	77.12	84.54	80.66
C4Gram	80.32	76.58	92.28	83.70	76.93	69.40	99.36	81.93
C5Gram	85.49	86.48	86.74	86.61	74.87	67.69	98.67	80.73
Fasttext	84.27	84.37	88.00	86.15	79.32	71.92	99.16	83.67
Word2Vec	86.77	82.68	94.97	88.40	81.93	74.39	99.14	85.00

In terms of test performance, the LR model achieved the best accuracy, precision, and f1-score with UBT and the best recall with Word2Vec (accuracy = 86.21%, precision = 81.62%, recall = 96.59%, f1-score = 87.64%). The MNB model achieved the best accuracy of 82.05% with Word2Vec, best precision of 80.39 with C3Gram, best recall of 99.56 with C5Gram, and best f1-score of 85.11% with Word2Vec on the test data. In cross validation, the DT model achieved the highest performance with C4Gram (accuracy = 91.11%, precision = 88.09%, recall = 96.85% and f1-score = 92.26%), whereas the lowest performance was achieved with Bigram. The RF model achieved the best accuracy, precision, and f1-score with C4Gram (92.22%, 92.63%, 92.91%) and the best recall with Word2Vec (94.45%). In test performance, the DT model achieved the best accuracy and precision with UBT, while its recall and f1-score were best with C4Gram (accuracy = 80.77%, precision = 77.898%, recall = 96.28% and f1-score = 83.70%). The RF model obtained the best accuracy and f1-score with UBT and the

best precision and recall with C5Gram and Fasttext (accuracy = 83.44%, precision = 93.40%, recall = 97.38, f1-score = 85.33%).

In cross-validation, the SGD model achieved the highest accuracy and recall with UBT, highest precision with C3Gram, highest f1-score C4Gram (accuracy = 88.49%, precision = 87.05%, recall = 96.02% and f1-score = 89.86%). The SGD model achieved high accuracy and precision with C3Gram, high recall with Word2Vec, and a high f1-score with UBT (accuracy = 83.38%, precision = 81.74%, recall = 99.43% and f1-score = 85.09%). The SVM model achieved the best accuracy, precision, and f1-score with Word2Vec and the highest recall with C3Gram (accuracy = 90.99%, precision = 87.87%, recall = 97.16%, f1-score = 91.91%) in cross-validation. In test performance, the SVM model achieved the best accuracy, precision, and f1-score with Unigram and the best recall with Word2Vec (accuracy = 84.94%, precision = 80.23%, recall = 94.47% and f1-score = 86.77%).

The KNN model obtained the best accuracy and precision with C3Gram, and the best recall and f1-score with Word2Vec in cross-validation (accuracy = 87.44%, precision = 87.56%, recall = 94.97% and f1-score = 88.40%). The best accuracy and precision were achieved with unigram, the best recall with Fasttext, and the best f1-score with Word2vec (accuracy = 82.88%, precision = 79.72%, recall = 99.16%, f1-score = 85.00%). Furthermore, the f1-score of both class (fake and true) for the above-mentioned models is illustrated in Figures 8 and 9.

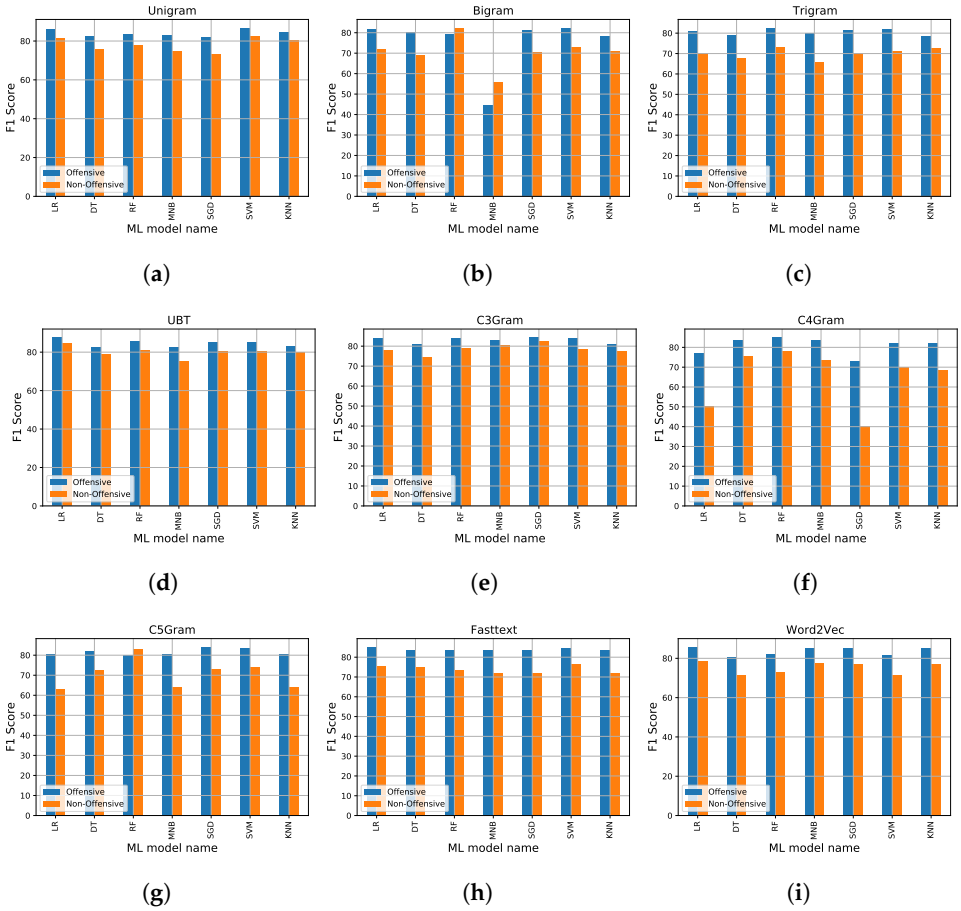


Figure 8. F1-score of various baseline methods on test dataset.

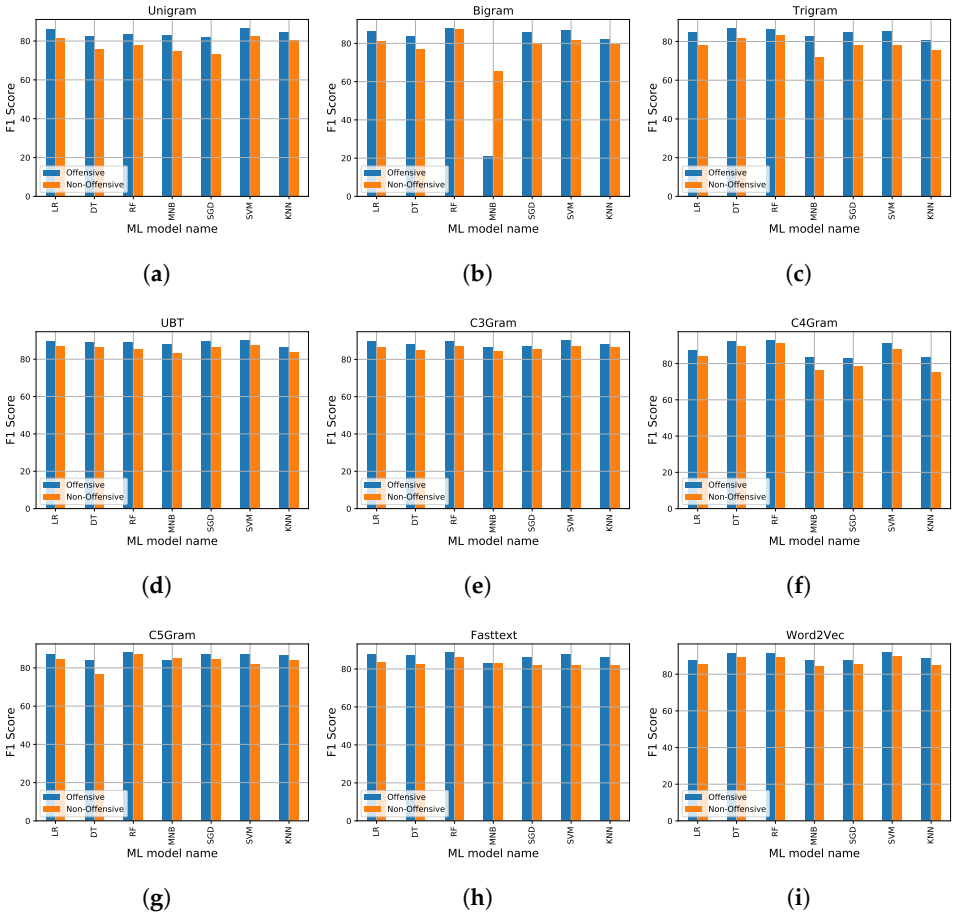


Figure 9. F1-score of various baseline methods on cross-validation datasets.

4.4.2. Deep Learning Classifiers

The Dl-based models used for comparison with our proposed model are CNN, Bi-LSTM, and M-BERT. In a variety of problems, CNNs are beneficial in classifying short and long texts. We used kernels with lengths ranging from 1 to 4. As a pooling layer, we examined the global max pool and average pool. The outputs of the global max-pooling layer were concatenated and supplied as a pooled feature vector to the following layer. The hidden layer of the network was a dense fully-connected layer preceded by a dropout layer. Consequently, classification was performed using the output layer, which consisted of a dense layer with an activation function. Within the network, we employed ReLU as the activation function. Table 11a displays the optimized hyperparameters of the CNN model for fake news classification.

LSTM networks are the most extensively utilized models in text categorization and generation challenges thanks to their ability to capture sequential information in an effective manner. Bidirectional LSTM (Bi-LSTM) has displayed particularly outstanding ability in catching sequential information from both directions in texts [40,47]. In this paper, we investigated LSTM and Bi-LSTM model with our dataset. We used a total of 256 LSTM units. Table 11b displays the optimized hyperparameters of the LSTM and Bi-LSTM models for classifying fake news. In addition, a combination of CNN and LSTM was implemented to compare them with our model. The hyperparameters of CNN-LSTM were identical to those of the CNN and LSTM models.

Table 11. Optimized hyperparameters adapted for CNN, LSTM, Bi-LSTM, and CNN-LSTM.

(a) Hyperparameters for CNN and CNN-LSTM.	
Parameters	**Parameters Value**
Kernel size	3
Filter Size	64
Activation	ReLU
Dropout Rate	0.2
Epochs	4
Batch Size	32
(b) Hyperparameters for LSTM, and Bi-LSTM.	
Parameters	**Parameters Value**
Neurons	100
Activation	ReLU, Sigmoid
Recurrent	
Dropout Rate	0.2
Epochs	4
Batch Size	32

Recently pre-trained multilingual language models have made significant progress in a variety of NLP tasks. BERT and its variant models, in particular, have outperformed the GLUE benchmark for Natural Language Understanding (NLU) [27]. To assess the scope of such a model in our research, we classified news articles using the multilingual BERT model. We used the pre-trained model from the HuggingFace Transformers libraries. A comparison of the above-mentioned models on the original dataset and augmented dataset is illustrated in Figure 10. The figure shows that handling imbalanced data improved the performance of these models. We experimented with undersampling the true news class as well. We trained our AugFake-BERT model with 4000 Fake news data items and 2000 true news data items. The model achieved an accuracy of 80.21% and a precision of 81.12%, whereas it achieved a low recall (56.35%) and f1-score (60.21%). Figure 11 illustrates the comparison of precision, f1-score, and recall on the imbalanced original dataset.

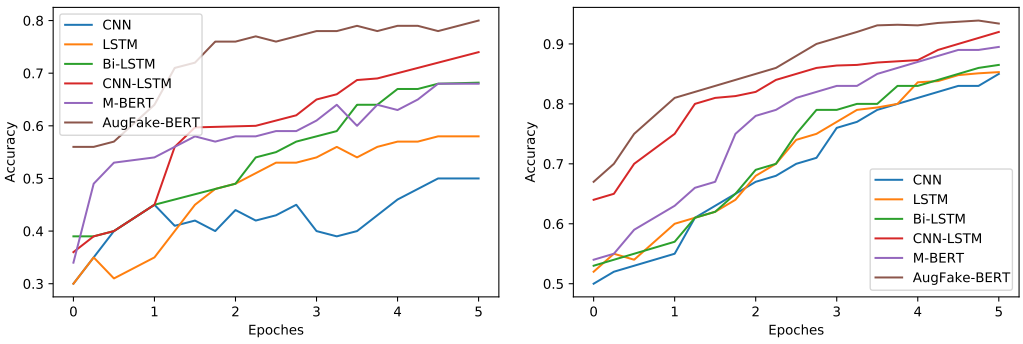


Figure 10. The effect of accuracy of the CNN, LSTM, Bi-LSTM, CNN-LSTM, M-BERT, and AugFake-BERT models on the original and augmented dataset. (**a**) Original dataset; (**b**) Augmented dataset.

Table 12 presents the testing result of the AugFake-BERT, M-BERT, CNN-LSTM, Bi-LSTM, LSTM, and CNN models on the test set of the augmented data, showing the highest performance obtained by our proposed AugFake-BERT model (accuracy = 92.45%, precision = 92.86%, recall = 91.23% and f1-score = 91.85%). The lowest performance is that of the CNN model, with an accuracy of 83.34%, precision of 80.23%, recall of 84.48%, and f1-score of 83.52%. The Bi-LSTM model has better results compared to CNN, with an accuracy of 85.10%, precision of 86.86%, recall of 85.43%, and f1-score of 86.85%. The combination of CNN-LSTM is the second highest in terms of performance (accuracy = 88.83%,

precision = 87.68%, recall = 87.23% and f1-score = 88.35%). The M-BERT obtained an accuracy of 88.69%, precision of 86.86%, recall of 87.33%, and f1-score of 89.55%. These classifiers' performance indicates that the dataset has a positive influence on performance.

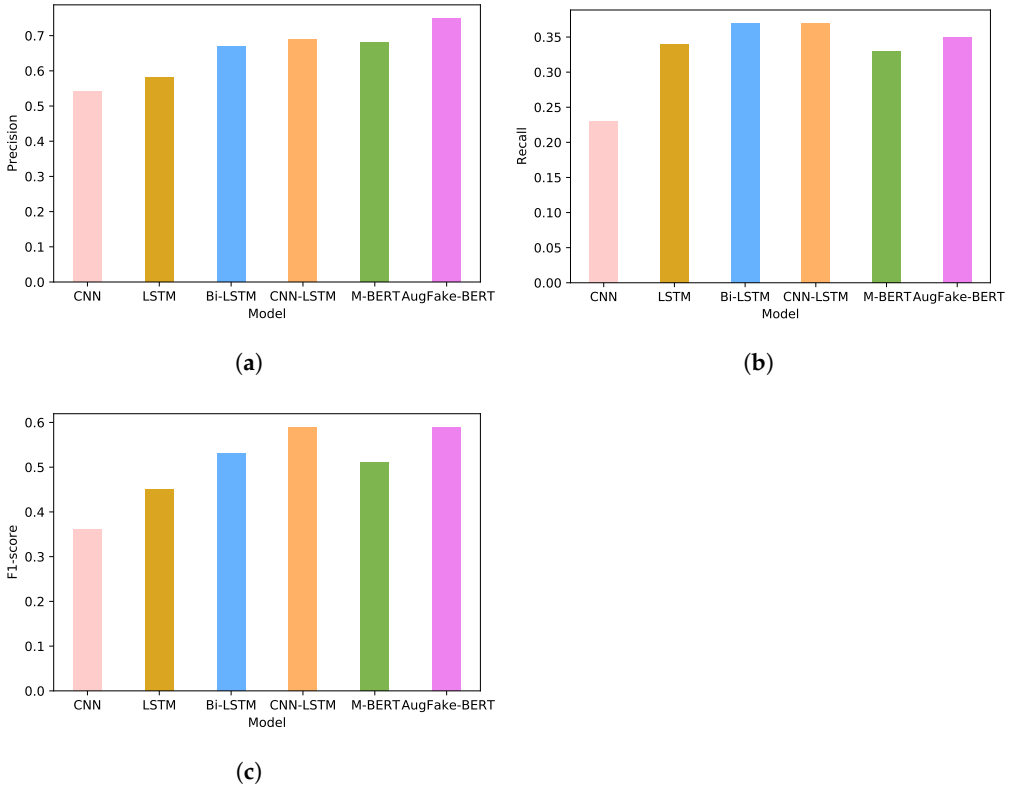


Figure 11. Comparison of evaluation metrics of CNN, LSTM, Bi-LSTM, CNN-LSTM, M-BERT, and AugFake-BERT models on the original dataset. (**a**) Precision; (**b**) Recall; (**c**) F1-score.

Table 12. Accuracy, precision, recall, and F1-score on the augmented dataset of deep learning classifiers.

	Test Performance			
Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
CNN	83.34	80.23	84.48	83.52
LSTM	84.69	84.86	85.72	84.15
Bi-LSTM	85.10	86.86	85.43	86.85
CNN-LSTM	88.83	87.68	87.23	88.35
M-BERT	88.69	86.86	87.33	89.55
AugFake-BERT	**92.45**	**92.86**	**91.23**	**91.85**

It is clear that the performance of all classifiers improves significantly after balancing the dataset. Thus, we trained the existing classifiers with the generated embeddings from BERT and performed the classification on both datasets. Furthermore, our proposed AugFake-BERT model outperformed the other state-of-the-art models. Table 13 shows the performance metrics of all used classifiers on both datasets (imbalanced and balanced) for comparison. The table proves that dataset balancing plays a significant role in classification results.

Table 13. Performance metrics of the classifiers on the balanced and imbalanced datasets.

	Imbalanced Dataset				Balanced Dataset			
Model	**A(%)**	**P(%)**	**R(%)**	**F1-S(%)**	**A(%)**	**P(%)**	**R(%)**	**F1-S(%)**
LR	50.27	51.63	24.31	37.52	88.66	85.11	87.58	86.16
MNB	52.16	52.10	22.19	46.29	86.16	82.42	83.45	85.60
DT	54.27	54.00	20.80	37.06	86.71	85.43	84.39	83.59
RF	54.16	53.00	21.50	39.49	85.21	84.26	84.62	84.74
SGD	55.49	55.21	22.42	40.90	88.41	87.63	87.62	88.67
SVM	55.17	55.70	23.83	40.24	88.32	87.28	88.00	88.57
KNN	53.71	51.09	20.77	30.29	87.60	87.54	87.70	87.56
CNN	54.13	54.34	23.46	36.34	83.34	80.23	84.48	83.52
LSTM	58.41	58.43	34.00	45.32	84.69	84.86	85.72	84.15
Bi-LSTM	67.24	67.45	37.78	53.68	85.10	86.86	85.43	86.85
CNN-LSTM	69.10	69.13	37.45	59.82	88.83	87.68	87.23	88.35
M-BERT	68.54	68.21	33.67	51.23	88.69	86.86	87.33	89.55
AugFake-BERT	**75.12**	**75.21**	**35.43**	**59.47**	**92.45**	**92.86**	**91.23**	**91.85**

5. Limitations and Future Work

This article presents an experimental analysis of an unbalanced Bengali dataset to detect fake news on social media sites using BERT. The proposed AugFake-BERT model can monitor numerous posts, comments, and messages in multiple social media networks to detect fake news. This model can be used by anyone willing to incorporate artificial intelligence into their systems to detect fake news. The proposed AugFake-BERT model had the highest accuracy in the standard Bengali language. As Bengalis speak differently depending on their location, this method may provide fairly low accuracy in such situations. In the future, multimodal learning might offer an extended version of the proposed work. Multimodal learning perceives images combined with textual content for the classification of fake news.

6. Conclusions

This paper introduces the AugFake-BERT technique, which generates synthetic fake data to circumvent the problem of minority classes and performs classification using the most advanced language model, Bidirectional Encoder Representation of Transformers (BERT). This strategy consists of a pre-trained multilingual BERT for enhancing text data and feeding the augmented data to a fine-tuned BERT for generating embeddings. We generated an augmented dataset which escalates the minority class (Fake). Furthermore, we fed the augmented data to the embedding BERT model. We experimented with a frozen and unfrozen BERT base uncased architecture to perform the embedding generation. Finally, a feed-forward neural network with softmax activation was used to classify the data. The experimental results indicate that the AugFake-BERT technique is superior to twelve other state-of-the-art algorithms in identifying the veracity of a news article (LR, DT, MNB, SGD, SVM, KNN, RF, CNN, LSTM, Bi-LSTM, CNN-LSTM, and M-BERT). We utilized various performance evaluation parameters, such as F1-score, precision, recall, etc., to validate the results. The comparison table indicates that the proposed unfrozen BERT trained with an augmented dataset is competitive in terms of various performance metrics. The proposed model outperformed the current models utilized by other researchers, with a final accuracy score of 92.45%. The performance of the proposed method is prominent when considering the small increase in the number of the minority class. We demonstrated that applying text augmentation significantly affects the classification outcome of the existing classifiers. We strongly believe that this work can motivate future researchers to invest effort in a robust self-supervised language model-based fake news detection system.

Author Contributions: Conceptualization, A.J.K., M.A.H.W. and M.F.M.; methodology, A.J.K.; data curation, M.F.M. and A.J.K.; software, A.J.K., M.A.H.W.; validation, M.F.M., M.A. and M.A.H.; formal analysis, M.A. and M.F.M.; investigation, A.J.K.; resources, M.F.M. and M.A.H.W.; writing—original draft preparation, A.J.K.; writing—review and editing, M.A., M.A.H.W. and M.A.H.; visualization, A.J.K. and M.A.H.W.; supervision, M.F.M.; project administration, M.F.M. All authors have read and agreed to the published version of the manuscript

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Olan, F.; Jayawickrama, U.; Arakpogun, E.O.; Suklan, J.; Liu, S. Fake news on Social Media: The Impact on Society. *Inf. Syst. Front.* **2022**. [CrossRef] [PubMed]
2. The True History of Fake News. 2021. Available online: https://www.economist.com/1843/2017/07/05/the-true-history-of-fake-news (accessed on 1 September 2021).
3. Kogan, S.; Moskowitz, T.J.; Niessner, M. Fake news: Evidence from financial markets. *SSRN Electron. J.* **2019**. [CrossRef]
4. Allcott, H.; Gentzkow, M. Social media and fake news in the 2016 election. *J. Econ. Perspect.* **2017**, *31*, 211–236. [CrossRef]
5. Rapp, D.N.; Salovich, N.A. Can't we just disregard fake news? The consequences of exposure to inaccurate information. *Policy Insights Behav. Brain Sci.* **2018**, *5*, 232–239. [CrossRef]
6. Deepak, S.; Chitturi, B. Deep neural approach to Fake-News identification. *Procedia Comput. Sci.* **2020**, *167*, 2236–2243. [CrossRef]
7. Mridha, M.F.; Keya, A.J.; Hamid, M.A.; Monowar, M.M.; Rahman, M.S. A Comprehensive Review on Fake News Detection with Deep Learning. *IEEE Access* **2021**, *9*, 156151–156170. [CrossRef]
8. Moscadelli, A.; Albora, G.; Biamonte, M.A.; Giorgetti, D.; Innocenzio, M.; Paoli, S.; Lorini, C.; Bonanni, P.; Bonaccorsi, G. Fake news and COVID-19 in Italy: Results of a quantitative observational study. *Int. J. Environ. Res. Public Health* **2020**, *17*, 5850. [CrossRef]
9. Mridha, M.F.; Wadud, M.A.H.; Hamid, M.A.; Monowar, M.M.; Abdullah-Al-Wadud, M.; Alamri, A. L-Boost: Identifying Offensive Texts From Social Media Post in Bengali. *IEEE Access* **2021**, *9*, 164681–164699. [CrossRef]
10. Jiang, T.; Li, J.P.; Haq, A.U.; Saboor, A.; Ali, A. A novel stacking approach for accurate detection of fake news. *IEEE Access* **2021**, *9*, 22626–22639. [CrossRef]
11. Islam, N.; Shaikh, A.; Qaiser, A.; Asiri, Y.; Almakdi, S.; Sulaiman, A.; Moazzam, V.; Babar, S.A. Ternion: An Autonomous Model for Fake News Detection. *Appl. Sci.* **2021**, *11*, 9292. [CrossRef]
12. Tao, C.C.; Cheung, Y.L.J. Social Media Mining on Taipei's Mass Rapid Transit Station Services based on Visual-Semantic Deep Learning. *WSEAS Trans. Comput.* **2022**, *20*, 110–117. [CrossRef]
13. Umer, M.; Imtiaz, Z.; Ullah, S.; Mehmood, A.; Choi, G.S.; On, B.W. Fake News Stance Detection Using Deep Learning Architecture (CNN-LSTM). *IEEE Access* **2020**, *8*, 156695–156706. [CrossRef]
14. Girgis, S.; Amer, E.; Gadallah, M. Deep learning algorithms for detecting fake news in online text. In Proceedings of the 2018 13th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 18–19 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 93–97.
15. Wadud, M.A.H.; Mridha, M.; Rahman, M.M. Word Embedding Methods for Word Representation in Deep Learning for Natural Language Processing. *Iraqi J. Sci.* **2022**, *63*, 1349–1361. [CrossRef]
16. Amine, B.M.; Drif, A.; Giordano, S. Merging deep learning model for fake news detection. In Proceedings of the 2019 International Conference on Advanced Electrical Engineering (ICAEE), Algiers, Algeria, 19–21 November 2019; pp. 1–4. [CrossRef]
17. Bugueño, M.; Sepulveda, G.; Mendoza, M. An Empirical Analysis of Rumor Detection on Microblogs with Recurrent Neural Networks. In Proceedings of the International Conference on Human-Computer Interaction, Paphos, Cyprus, 2–6 September 2019.
18. Ahmad, T.; Faisal, M.S.; Rizwan, A.; Alkanhel, R.; Khan, P.W.; Muthanna, A. Efficient Fake News Detection Mechanism Using Enhanced Deep Learning Model. *Appl. Sci.* **2022**, *12*, 1743. [CrossRef]
19. Saleh, H.; Alharbi, A.; Alsamhi, S.H. OPCNN-FAKE: Optimized convolutional neural network for fake news detection. *IEEE Access* **2021**, *9*, 129471–129489. [CrossRef]
20. Keya, A.J.; Afridi, S.; Maria, A.S.; Pinki, S.S.; Ghosh, J.; Mridha, M.F. Fake News Detection Based on Deep Learning. In Proceedings of the 2021 International Conference on Science & Contemporary Technologies (ICSCT), Dhaka, Bangladesh, 5–7 August 2021; pp. 1–6. [CrossRef]
21. Ajao, O.; Bhowmik, D.; Zargari, S. Fake News Identification on Twitter with Hybrid CNN and RNN Models. In Proceedings of the 9th International Conference on Social Media and Society (SMSociety'18), Copenhagen, Denmark, 18–20 July 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 226–230. [CrossRef]

22. Wadud, M.A.H.; Mridha, M.; Shin, J.; Nur, K.; Saha, A.K. Deep-BERT: Transfer Learning for Classifying Multilingual Offensive Texts on Social Media. *Comput. Syst. Sci. Eng.* **2022**, *44*, 1775–1791. [CrossRef]
23. Wadud, M.A.H.; Rakib, M.R.H. Text coherence analysis based on misspelling oblivious word embeddings and deep neural network. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*. [CrossRef]
24. Ye, H.; Li, Y. Fuzzy Cloud Evaluation of Service Quality Based on DP-FastText. *WSEAS Trans. Comput.* **2021**, *20*, 149–167. [CrossRef]
25. Ding, J.; Hu, Y.; Chang, H. BERT-Based Mental Model, a Better Fake News Detector. In Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence (ICCAI'20), Tianjin, China, 23–26 April 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 396–400. [CrossRef]
26. Jwa, H.; Oh, D.; Park, K.; Kang, J.M.; Lim, H. exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert). *Appl. Sci.* **2019**, *9*, 4062. [CrossRef]
27. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
28. Wei, J.; Zou, K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv* **2019**, arXiv:1901.11196.
29. Rizos, G.; Hemker, K.; Schuller, B. Augment to prevent: Short-text data augmentation in deep learning for hate-speech classification. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 991–1000.
30. Hiriyannaiah, S.; Srinivas, A.; Shetty, G.K.; Siddesh, G.; Srinivasa, K. A computationally intelligent agent for detecting fake news using generative adversarial networks. In *Hybrid Computational Intelligence: Challenges and Applications*; Academic Press: Cambridge, MA, USA, 2020; p. 69.
31. Szmigiera, M. Most Spoken Languages in the World. 2021. Available online: https://www.statista.com/statistics/266808/the-most-spoken-languages-worldwide/ (accessed on 1 September 2021).
32. Wadud, M.A.H.; Kabir, M.M.; Mridha, M.; Ali, M.A.; Hamid, M.A.; Monowar, M.M. How can we manage Offensive Text in Social Media-A Text Classification Approach using LSTM-BOOST. *Int. J. Inf. Manag. Data Insights* **2022**, *2*, 100095. [CrossRef]
33. Al-Zaman, M.S. COVID-19-related social media fake news in India. *J. Media* **2021**, *2*, 100–114. [CrossRef]
34. Al-Zaman, M.S. Social media fake news in india. *Asian J. Public Opin. Res.* **2021**, *9*, 25–47.
35. Hossain, M.Z.; Rahman, M.A.; Islam, M.S.; Kar, S. BanFakeNews: A dataset for detecting fake news in bangla. *arXiv* **2020**, arXiv:2004.08789.
36. Seddari, N.; Derhab, A.; Belaoued, M.; Halboob, W.; Al-Muhtadi, J.; Bouras, A. A Hybrid Linguistic and Knowledge-Based Analysis Approach for Fake News Detection on Social Media. *IEEE Access* **2022**, 10, 62097–62109. [CrossRef]
37. Lai, C.M.; Chen, M.H.; Kristiani, E.; Verma, V.K.; Yang, C.T. Fake News Classification Based on Content Level Features. *Appl. Sci.* **2022**, *12*, 1116. [CrossRef]
38. Singhania, S.; Fernandez, N.; Rao, S. 3han: A deep neural network for fake news detection. In Proceedings of the International Conference on Neural Information Processing, Long Beach, CA, USA, 4–9 December 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 572–581.
39. Jin, Z.; Cao, J.; Guo, H.; Zhang, Y.; Luo, J. Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In Proceedings of the 25th ACM international conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 795–816.
40. Aloshban, N. ACT: Automatic Fake News Classification Through Self-Attention. In Proceedings of the 12th ACM Conference on Web Science, Southampton, UK, 6–10 July 2020; pp. 115–124.
41. Lu, Y.J.; Li, C.T. GCAN: Graph-aware co-attention networks for explainable fake news detection on social media. *arXiv* **2020**, arXiv:2004.11648.
42. Zhang, T.; Wang, D.; Chen, H.; Zeng, Z.; Guo, W.; Miao, C.; Cui, L. BDANN: BERT-Based Domain Adaptation Neural Network for Multi-Modal Fake News Detection. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8. [CrossRef]
43. Kaliyar, R.K.; Goswami, A.; Narang, P. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimed. Tools Appl.* **2021**, *80*, 11765–11788. [CrossRef]
44. Wu, C.L.; Hsieh, H.P.; Jiang, J.; Yang, Y.C.; Shei, C.; Chen, Y.W. MUFFLE: Multi-Modal Fake News Influence Estimator on Twitter. *Appl. Sci.* **2022**, *12*, 453. [CrossRef]
45. Yu, L.; Zhang, W.; Wang, J.; Yu, Y. Seqgan: Sequence generative adversarial nets with policy gradient. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
46. Kula, S.; Choraś, M.; Kozik, R. Application of the BERT-based architecture in fake news detection. In Proceedings of the Computational Intelligence in Security for Information Systems Conference, Seville, Spain, 13–15 May 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 239–249.
47. Liao, Q.; Chai, H.; Han, H.; Zhang, X.; Wang, X.; Xia, W.; Ding, Y. An Integrated Multi-Task Model for Fake News Detection. *IEEE Trans. Knowl. Data Eng.* **2021**. [CrossRef]