Enhancement of Fake Reviews Classification Using DeepLearning Hybrid Models

Vikas Attri[1*], **Isha Batra**[2], **Arun Malik**[3]

[1,2,3] Department of Computer Applications, Lovely Professional University, Phagwara, Punjab 144001, India.

*Corresponding author(s). E-mail(s): vikasmca09@gmail.com;
Contributing authors: Isha.batra2487@gmail.com; arunmalikhisar@gmail.com;

Abstract

Fake reviews can significantly affect consumer behaviour and a company's reputation, making it crucial to identify them in today's digital age. This study uses deep learning to detect fake reviews. The proposed approach captures review text contextual data using five deep learning designs: LSTM, Bidirectional LSTM, multi-dense LSTM, GRU, and Bidirectional GRU. The models were trained on a large annotated dataset of reviews and assessed using precision, recall, accuracy, and F1-score. This investigation's primary objective is to establish whether or not a review can be trusted as genuine. According to the findings, deep learning models perform noticeably better than more traditional machine learning-based approaches and provide a reliable solution for the detection of fake reviews. The proposed research proves that Bidirectional LSTM generates 99.9 accuracy results on training data and 85.59 for validation data, among other models. For future research, we will focus on developing text enriches columns by adding a polarity feature to the existing dataset and ensemble modelling for novelty purposes. This study helps researchers to take potential benefits in various application domains, including e-commerce and consumer feedback platforms, where accurately detecting fake reviews is essential for maintaining trust and transparency.

Keywords: Fake Reviews, Deep Learning, Word Embedding Techniques, Hybrid Modeling, Ensemble Modeling.

1. INTRODUCTION

Nowadays, fake reviews[FR] are everywhere you go, confusing shoppers about which products or businesses are genuinely good. There is always the possibility that the reviews you read could be better, regardless of whether you are shopping on Amazon, researching restaurants on Tripadvisor, or looking into potential employers on Glassdoor.

The rise of e-commerce has brought the challenge of FR, where individuals or organisations write false reviews to deceive potential customers [1]. Fake reviews have become particularly prevalent in recent years, leading to a loss of trust in online reviews as reliable data sources [2]. This is where deep learning has stepped in to help mitigate the issue.

DL is a subfield of ML that focuses on using ANN to model complex patterns & relationships in large datasets [3]. In the context of fake reviews, deep learning algorithms could identify FR by analysing patterns and relationships in large datasets of studies [4]. The deep learning algorithms can be trained on a vast dataset of reviews to learn the characteristics of fake & genuine reviews and then be applied to new reviews to categorise them as fake or real [5].

One of the main approaches to detecting FR using DL is sentiment analysis [6]. Sentiment analysis has been widely used to detect fake reviews [7].DL methods could be trained on extensive review databases to learn the patterns and relationships between the sentiment expressed in a review and the likelihood that the review is fake [8]. The DL algorithms could then be applied to new studies to determine the sentiment expressed and classify the review as fake or genuine [9].

Another approach to detecting FR using DL is text classification [10]. Text classification allocates a label or category to a piece of text, which has been widely used in fake review detection [11]. DL approaches could be trained on large review databases to learn the patterns and relationships between the text of a review and the likelihood that the review is fake. The DL algorithms could then be applied to new studies to classify them as fake or genuine [12].

1.1. Advantages and Disadvantages of Deep Learning

One of the main advantages of deep learning in fake review detection is its ability to learn patterns and relationships in large datasets. DL approaches could be trained on large review databases to understand the characteristics of fake and genuine reviews and then be applied to new studies to classify them as fake or real. This means that deep learning algorithms can be highly effective in detecting fake reviews, even in cases where the FR is written in a complex way to identify using traditional methods.

Another advantage of deep learning in fake review detection is its ability to tackle high-dimensional information. In the context of fake reviews [13], DL approaches could be educated on large databases of reviews, which can include a large number of features, such as the sentiment expressed in the review, the writing style of the reviewer, and the content of the review.

Despite its advantages, some challenges are associated with using deep learning in FR detection. One of the main challenges is the need for large review datasets to train deep learning algorithms. To effectively identify FR using deep learning algorithms, large datasets of fake and genuine reviews must be collected and labelled, which can be a time-consuming and costly process. There is also the challenge of avoiding bias in the training data, as the algorithms can only be as good as the data they are trained on. Finally, deep learning algorithms can also be computationally intensive, requiring significant computing resources and preparation time, which can limit their practical use in real-world applications [14].

1.2. Role of Classification in Deep Learning

Classification plays a crucial role in fake review detection using deep learning. An FR detection design aims to classify the reviews as either fake or real [15]. In this task, the input data is typically a written review, and the desired output is a class label indicating whether the review is real or fake [16]. A DL model can be educated on a large labelled dataset of real and fake reviews to learn the patterns and characteristics of fake and real reviews [17]. The trained model can then classify new reviews as either real or fake [18]. In conclusion, classification is essential in the fake review detection task as it allows DL designs to categorise the input information into two distinct categories of real and fake reviews, which has significant implications for e-commerce, sentiment analysis, and consumer protection [19].

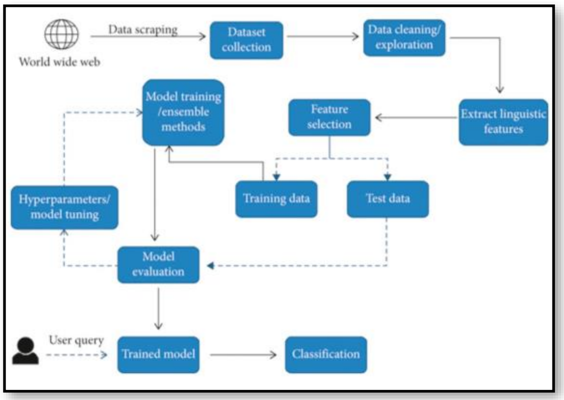


Fig. 1. Process of Algorithm Selection[72]

One of the reasons it might be challenging to identify fake reviews is that they can take on various shapes. They can be broken down into two primary categories: human-generated fake reviews and computer-generated fake reviews. However, whether written by a human or computer, reviews can have a positive or negative tone. Their purpose can be to either raise or lower the overall rating or increase the total number of reviews to lend more legitimacy to the score.

Identifying fake evaluations written by computers may become more challenging when more advanced forms of artificial intelligence come into use, such as GPT-2 and GPT-3. A recent study on reviews generated by GPT-2 (Salminen et al., 2022) demonstrates that models can detect GPT-2 reviews with relatively excellent accuracy and that models do better than people when it comes to their identification. On the other hand, more needs to be written about the detection of GPT-3 generated reviews up to this point.

2. LITERATURE SURVEY

Several researchers have studied the problem of fake reviews [20]. **Christopher et al. (2022)** merge multiple techniques to spot fake restaurant reviews on Yelp. Authors develop and deploy a bi-directional LSTM, a recurrent neural network, to benefit from the positional relevance of comments inside reviews. The precision of our prototype is improved by analysing various sections of text within the review because LSTMs are well-suited to studying linguistic features in the text. We apply a Kullback-Leibler (K-L) divergence technique to this element to investigate the disparity in term rankings among authentic and bogus Yelp reviews. This ensemble achieves an Average Precision of 0.5402 and an AOC of 0.866, distinguishing between fake and genuine reviews when applied to the same YelpNYC dataset, outperforming other cutting-edge methods[24].

Istiaq Ahsan et al. (2016) used a hybrid database of real-life or fake reviews to demonstrate an ensemble learning strategy that combines active and supervised learning. KL and JS distance, TF-IDF, and n-gram properties of review content determine this model's three filtering phases. It generates incredible outcomes with 3600 domain-specific evaluations. Accuracy, precision, recall, and f-score are all above 95% in the best case. Two thousand reviews were hand-labeled. After reviewing and comparing outcomes with other effective ways, this identifying methodology is practical and very perspective [25].

Navya Singh et al. (2022) have employed the LIAR dataset to identify fake news. 12.8k sentences from PolitiFact.com that have been previously classified and are from different contexts make up this data. In the suggested model, the authors implemented a novel concentration strategy for detection using BERT embedding. Researchers achieved the most advanced outcome by comparing our conceptual approach to the currently used methods. The results of tests demonstrate that additional investigation using policy is possible[26].

Hua Jiang et al. (2022) utilise Word2Vec-LSTM to analyse movie reviews' sentiment. Word2Vec creates a word vector using text context semantics, and LSTM categorizes positive and negative feelings using semantic data. The Word2Vec-classification LSTM's power is measured using Word Index & Hash Trick approaches. Writers combine the word index and hash trick with various prominent machine learning algorithms to generate the word index or Hash Trick-Based Classifiers. Experimental results show Word2Vec-LSTM performs best. For movie review sentiment analysis, the Word2Vec-LSTM hybrid model outperforms Word Index-Based Classifiers and Hash Trick-Based Classifiers by 29.12% and 18.84%, respectively [27].

Putra et al. (2021) suggested using the Word2Vec paradigm and the Long-Short Term Memory (LSTM) model to authenticate fake reviews. The Word2Vec architecture, the Word2Vec vector dimension, the Word2Vec evaluation method, the pooling approach, the dropout value, and the learning rate are the combined Word2Vec and LSTM variables utilized in this study. The best performance, which had an accuracy of 85.96%, was attained using a dataset of 2500 review texts in an experiment conducted. For Word2Vec, the variable choices are 300 as the vector dimension, Hierarchical Softmax as the evaluation technique, and Skip-gram as the structure. A dropout value of 0.2, average pooling as the pooling type, and 0.00 as the learning rate are the parameter values for LSTM, respectively[28].

Madhuri et al. (2022) proposed a more efficient BERT technique in feature extraction using a sizeable set of data and hybridised deep-learning approach. As a result, data from consumer reviews of Amazon products are used for the research. Amazon is used to retrieve reviews of mobile phones for SA. The data were first preprocessed to improve the classifier's performance and accuracy. Moreover, the BERT (Bidirectional Encoder Representations from Transformers) technique is used for feature extraction to condense large amounts of data into precise ones. With effective classification, analysing the review feelings is easier. The deep Bi-LSTM-GRU neural network methodology is compared in this study to how it is currently being used, and the efficiency of each is analyzed using this procedure. According to the research outcomes, depending on the accuracy level, frequency, precision, or recall measures, the suggested algorithm achieved 97.87, 98.36, 98.89, and 98.47, respectively [29].

Muhammad et al. (2021) describe a method based on three models, each trained using the multi-view learning concept and then using an aggregation technique to combine all the predictions into an ensemble to provide final forecasts. The methodology primarily aims to extract detailed information from reviews' text by merging parallel CNN and bag-of-n-grams. Using the same amount of processing resources needed to train deep and intricate CNNs, can exploit local context by employing an n-gram embedding layer with modest kernel sizes. The parallel convolutional blocks in our CNN-based design extract richer image features from text using n-gram embeddings as input. The method of identifying fake reviews also integrates non-textual characteristics of reviewer behaviour with textual, and linguistic features. Authors test the methodology using a publicly accessible Yelp Filtered Set and identify fake reviews with F1 scores of up to 92%[30].

Bundit et al. (2021) propose a graph partitioning approach (BeGP) that differentiates between fraudulent and trustworthy reviewers that offer the graph partitioning method BeGP and its expansion BeGPX. BeGP's basic principle is to create a behavioural network in which reviewers are connected if they have shared traits that accurately represent their shared behaviour. After that, the search algorithm grows the subgraph by introducing additional linked suspicious reviewers after starting with a tiny subgraph of known false reviewers. The reviews of those suspects are therefore assumed to be fake. Furthermore, BeGPX incorporates additional research on the semantic content and emotions conveyed in studies to improve the effectiveness of false reviewer identification. To incorporate into the graph creation process, authors use the DNN to acquire word embedding representations with lexicon-based mood signals. We use two authentic Yelp.com review data to show the efficacy of BeGP & BeGPX. The findings demonstrate that both strategies perform better than cutting-edge techniques in identifying fake reviewers within the k-first order of ranks. Also, despite receiving a limited amount of educational labelled data, BeGPX exhibits notable improvement[31].

Dibyajyoti et al. (2021) used the glove embedding matrix and the BoW model, emphasising bogus reviews. Three fresh DL algorithms and two distinct feature extraction methods have been used to text classification. Compared to conventional ML algorithms, the experimental study of a dataset produced even good performance[32].

KadekSastrawan et al. (2022) employ a DL technique that combines word embedding that has been taught, using four different datasets with pre-trained architecture such as CNN, Bidirectional LSTM, or ResNet. Each piece of data is put through a data augmentation operation that uses the back- translation approach to bring the discrepancies between the various classes down to a more manageable level. According to the findings, the CNN and ResNet designs performed less favorably than the Bidirectional LSTM architecture [33] over the entire datasets that were examined.

Gourav Bathla et al. (2022) used sentiment polarity for the fraudulent review detection phase after being retrieved from reviews. For aspect replication learning, extracted features are input into CNN. To detect false reviews, the reproduced aspects are fed into LSTM. Reliability comparisons with more modern methods are made using the Ott or Yelp Filter databases. Analysis of the results of experiments shows that our suggested strategy surpasses current strategies[34].

Abrar et al. (2019) stated that the BERT method might retrieve word embeddings from texts (i.e., reviews). The production of word embeddings uses various foundational techniques, including SVM, RF, and NB. In addition to that, we used a method called the confusion matrix to evaluate and illustrate the results. The accuracy of the SVM classifiers is 87.81%, which is 7.6% greater than the accuracy of the classifier utilized in the earlier investigation. According to the data, the SVM classifiers are superior to the others regarding reliability and f1-score [35].

David et al. (2021) perform detailed experiments on various online review databases using the cutting-edge language design BERT. The model achieves 91% accuracy on the balanced crowd-sourced database of hotel, restaurant, & doctor reviews as well as 73% accuracy on the unbalanced third-party Yelp database of restaurant reviews, outperforming previous detection techniques[36].

Umer Hayat et al. (2022) proposed work made two contributions: (1) Building the new Roman Urdu Fake Reviews Detector Corpus (RU-FRDC), which consists of 5150 annotated reviews; & (2) Comparing different DL architectures, such as Simple RNN, LSTM, GRU, Bi-LSTM, and Bi-GRU. Precision, Recall, F1, and ACC-ROC are four commonly used evaluation metrics used in the assessment. The stacked LSTM design produced the best outcomes (ACC - ROC = 0.943 and F1=0.88)[37].

Chunyong et al. (2021) present a system for identifying fraudulent reviews based on active learning and vertical ensemble tri-training (VETT-AL). As part of the feature extraction process, the system combines user behaviour characteristics with review text elements. Integration within the group and horizontal integration between groups are the two unique components that comprise the iterative process utilized by the VETT-AL technique. The intra-group integration attempts to merge three original classifiers using the earlier iterative models of the classifiers. Integration between groups will use active learning, based on entropy, to classify the information with the highest possible degree of significance. As a direct result, the second generation of classifiers will be trained to utilize the tried-and-true method to improve the dependability of the label [38].

Budhi et al. (2021) suggest 133 distinct features by integrating content and behaviour-based elements (80 for content features, 29 behaviours, and 24 product features). They used a sampling procedure, which may have included over- or under-sampling, to resolve unbalanced data and increase minority class accuracy. MLP, LR, DT, CNN, and SVM classifiers were used in 10-fold cross-validation experiments. [43].

The study that was carried out by **Wang et al. (2018)** makes use of a framework known as an RNN, which is equipped with LSTM. Analytical findings from this study were compared to results from earlier studies utilizing a real-life case study of a fake review from Taiwan. To assess the performance of deep learning models, accuracy, precision, recall, and F-score were utilized. While comparing the LSTM method to the Support Vector Machine (SVM) technique, [44] it was found that the LSTM algorithm performed better.

Wang and Zhang(2017) presented a bidirectional LSTM and RNN-based method for automatic keyword extraction. Using an LSTM RNN-based filter, they determined whether the reviews were relevant. The review was forwarded to the keyword extraction stage if it was suitable. LSTM RNN -the classifier was trained to convert the original sentences into labels at the keyword extraction stage. Labels that occurred more frequently were used as keywords. The authors achieved 98.9% accuracy in word filtering and 91.5% accuracy in keyword extraction[45].

Girgis et al.(2018) use RNN technique models (Vanilla, GRU) and LSTMs to build a classifier that can predict whether a message is a forgery based solely on its content, looking at the problem from a pure deep learning perspective. To compute and evaluate the results, the LIAR dataset was utilized. The GRU algorithm (0.217) produces the best results, followed by the LSTM algorithm (0.2166) and then by the Vanilla algorithm (0.215)[46].

The article by **Lin et al. (2014)** compares CNNs, GRUs, and LSTMs. On the NLP front, they have worked on sentiment/relation categorization, textual entailment, answer selection, Freebase question-relation matching, Freebase path query answering, and part-of-speech tagging, among other things. The authors concluded that they achieved an accuracy of 86.32 percent for the Text GRU, 68.56 percent for the Semantic GRU, and 68.56 percent for the RC GRU.In the rematch, AS CNN was 65.01% accurate, QRM CNN was 71.50% valid, and TE GRU was 78.78% accurate. GRU achieved a Reorder (PQA) accuracy of 55.67%. In Contested, Bi-LSTM achieved 94.35% accuracy (POS tagging)[47].

uthor /Year	Aim	Algorithm/Technique	Dataset	Conclusion
Istiaq Ahsan et al., (2016)	Identification of Fake reviews	Ensemble learning approach	Hybrid database containing both actual and fake reviews.	Accuracy = above 88
Madhuri et al.,(2022)	Identification of Fake reviews	Deep Bi-LSTM-GRU neural network	Amazon products` customer review datasets	Depending on accuracy,frequency, precision, orrecall, the values are 97.87, 98.36, 98.89, and 98.47.
Muhammad et al.,(2021)	Using parallel CNN & bag-of-n-grams, derive detailed information from review text.	Convolution neural networks(CNNs).	Yelp Filtered Dataset	F1 scores = 92%
David et al.,(2021)	Identification of Fake reviews	BERT	Crowdsourced dataset	73% accuracy
Christopher et al.,(2022)	Analyze linguistic elements in text while assessing various textualareas within the evaluation.	LSTM	YelpNYC dataset	Average Precision (AP) of 0.5402
Umer Hayat et al.,(2022)	Roman Urdu Fake Reviews Identification Corpus	RNN, LSTM, GRU, Bi-LSTM, Bi-GRU.	5150 annotated reviews	ACC - ROC = 0.943 and F1=0.88
Gourav Bathla et al., (2022)	Only these elements and their corresponding emotions are used for theextraction of bogus reviews.	CNN, LSTM	Ott and Yelp Filter datasets	Proposed approach outperforms recent approaches.
Budhi et al. (2021)	Suggest 133 distinct features by integrating content and behaviour-based elements (80 for content features, 29 behaviours, and 24 product features).	MLP, LR, DT, CNN, and SVM	Resolve unbalanced data	increase minority class accuracy
Wang and Zhang (2017)	Automatic keyword extraction.	LSTM RNN	Data crawled from jd.com	Filtered Accuracy: • Character = 99.5% • Word = 98.9% Keywords Extraction Accuracy: • Character = 93.7%Word = 91.5%
Girgis et al.,(2018)	Detecting Fake News	RNN technique	LIAR	Accuracy = 0.217 (GRU)

Table 1: Comparison of existing work in tabular form

3. PROPOSED TECHNIQUES

3.1. Word Embedding Techniques

Fake reviews have become a widespread problem in today's digital world. Detecting fake reviews is a challenging task, but it can be made more accessible by using word embedding techniques. Word embedding techniques are prevalent in natural language processing (NLP) applications, such as text categorization, sentiment analysis, and detecting fake review sites. In fake review detection, word embedding techniques represent the text of reviews as numerical vectors that capture the semantic and syntactic relationships between words.

3.1.1. Static Word Embedding Techniques

Techniques for static word embedding are founded on pre-trained word embeddings acquired through the study of a substantial text corpus. While detecting fake reviews, these embeddings will remain unchanged because they are fixed. Word2Vec, GloVe, and FastText are the three methods of static word embedding with the most widespread use[41]. These techniques have been used for fake review detection with promising results. For instance, in a study by [39], Word2Vec and GloVe were utilized to extract features from reviews to detect fake reviews. The study's findings demonstrated that these embeddings performed better than conventional bag-of-words and tf-idf approaches. Similarly, in [40], FastText was used to learn word embeddings for fake review detection. The results showed that the FastText embeddings achieved higher accuracy than traditional bag-of-words and tf-idf methods.

- **Word2Vec**

The Word2vec approach is a well-known choice for creating word embeddings and vector representations of words in a space with a high dimension. Natural language processing (NLP) applications like language modelling, sentiment analysis, and machine translation extensively use word embeddings, which record the semantic and syntactic links between words.

In 2013, Tomas Mikolov and his coworkers at Google presented for the first time the methodology for Fast Prediction of Language Modeling in Embeddings [48]. Since then, word2vec has become one of the most widely used algorithms in NLP due to its efficiency and effectiveness in generating high-quality word embeddings.

The Continuous Bag-of-Words (CBOW) and the Skip-gram architectures are the two primary ones that Word 2 vec uses. Within the CBOW architecture, the model predicts a target word based on the context of the surrounding terms. Given a target word, the model predicts the words surrounding that word using the Skip-gram architecture. Both architectures use a neural network with a single hidden layer to learn word embeddings.

One recent development in word2vec research is using subword information to improve word embeddings. In languages with complex morphology, such as Finnish and Turkish, words can have multiple inflexions and derivations that can change their meaning. By using subword information, such as character n-grams, the algorithm can capture these morphological variations and generate more robust word embeddings. The performance of natural language processing tasks such as language modelling and named entity recognition has improved when using this methodology, known as subword modelling [49].

Another recent development is contextualized word embeddings, which consider the context in which a word appears. Contextualized word embeddings can be generated by relevant neural networks using vast amounts of text input, such as in the BERT (Bidirectional Encoder Representations from Transformers) model [50]. In various natural language processing applications, such as question answering and sentiment analysis, it has been demonstrated that these embeddings perform better than static word embeddings.

- **GloVe**

Another well-known technique for constructing word embeddings is "Global Vectors for Word Representation," or GloVe for short. Its name comes from the acronym. GloVe, much like word2vec, constructs word embeddings by first undergoing training on enormous amounts of text data. In this way, the two systems are very similar. On the other hand, GloVe adopts a unique technique, which involves taking into account the co-occurrence statistics of terms throughout the corpus.

In 2014, J**effrey Pennington** and his fellow Stanford University researchers published a paper titled " Glo Ve: Global Vectors for Word Representation" [51]. In this publication, they presented the GloVe algorithm. The algorithm is predicated on the concept that the ratio of the co-occurrence probabilities of two words should indicate the percentage of those words that are similar to one another. GloVe employs a weighted least squares model to achieve the best possible ratio when learning word embeddings.

One advantage of GloVe over word2vec is that it can capture global context rather than just local context information. This means that GloVe can capture the immediate context of a word and the more significant semantic relationships between words in the corpus.

GloVe has been shown to perform well on various NLP tasks, such as sentiment analysis, named entity recognition, and machine translation. One recent development in GloVe research is using hierarchical structures to improve the algorithm's performance. Hierarchical GloVe models use a tree structure to organise groups of words that are semantically related to one another and to learn embeddings at various levels of the tree. It has been demonstrated that using this strategy can increase the embedding's quality and decrease the time needed for training [52].

- **FastText**

The Artificial Intelligence Research team at Facebook (FAIR) developed an algorithm known as FastText in 2016 [53] to produce word embeddings. Word embeddings represent reconstructions of phrases that take the form of vectors and can encapsulate the semantic meaning of the words as well as the relationships between the words. The activities of text classification, sentiment analysis, and machine translation are all examples of standard natural language processing (NLP) applications that use them.

FastText differs, in that it considers the subword information of words, such as character n-grams, in addition to the word itself. This approach allows FastText to capture the morphology and compositionally of words, which is particularly useful for languages with complex morphology.

The FastText algorithm breaks words into smaller subword units, such as character n-grams, and learning embeddings for each subword unit. The embeddings for the subword units are then combined to generate an embedding for the whole word. This approach allows FastText to handle out-of-vocabulary (OOV) words, not in the training corpus. FastText can generate embeddings for OOV words by combining the embeddings of their subword units.

FastText has been shown to perform well on various NLP tasks and languages and has been used in production systems at Facebook for functions such as ad targeting and content ranking. One advantage of FastText is its ability to handle OOV words, which benefits low-resource languages.

Recent research on FastText has focused on improving the quality and efficiency of the algorithm. One approach is to use subword sampling to reduce the computational cost of training FastText models [54]. Another method uses hierarchical softmax to speed up the training process and reduce memory requirements [55]. Additionally, there has been researched on using FastText for other applications, such as knowledge graph completion [56] and text-to-speech synthesis [57].

FastText is a powerful and versatile algorithm for generating word embeddings and can improve the performance of many NLP tasks and applications.

3.1.2. Dynamic Word Embedding Techniques

Dynamic word embedding techniques are based on word embeddings learned from task-specific data during the fake review detection process. These embeddings change during the process, and the model updates them with each iteration. The most popular dynamic word embedding techniques are ELMo and BERT.

- **BERT**

BERT is an architecture for a deep neural network that can understand the contextual links between words in a sentence by being pre-trained on a significant amount of unlabelled text input[58].

BERT is designed to be bidirectional, meaning it can process both the left and proper contexts of a word, unlike previous language models that only processed one direction. In addition to that, it makes use of a transformer architecture, which is a self-attention mechanism[21].

BERT's capability of handling long-term dependencies in the text is one of its advantages. This capability is vital for activities requiring knowledge of the context of a phrase or paragraph, and BERT succeeds in this area.

After its initial release, other variants of BERT have been developed and made available, such as Roberta [59], which enhances the pre-training process, and ALBERT [60], which decreases the number of parameters while retaining performance. Both of these variants of BERT have been made available. Recently, BERT has been fine-tuned for specific domains, such as biomedical text [61] and legal text [62], and has been used for applications such as chatbots and virtual assistants.

One limitation of BERT is its computational cost, as it requires many computing resources to pre-train and fine-tune the model. To address this, there have been efforts to optimise the BERT architecture and reduce its computational requirements, such as DistilBERT [63], which uses a minor architecture and a knowledge distillation technique to achieve comparable performance with fewer computing resources.

Overall, BERT has significantly impacted the field of NLP and has led to improvements in a wide range of NLP tasks.

- **ELMo**

In 2018, Allen Institute for Artificial Intelligence researchers introduced ELMo, a deep contextualized word embedding model [64]. ELMo generates context-sensitive graphics instead of fixed word embeddings.

ELMo was developed using a deep bidirectional language model (biLM) trained on a significant text corpus. The film is made up of two layers of stacked LSTMs, which are also known as long-term, short-term memories. These LSTMs are responsible for processing the input text in both the forward and the backward directions.

For researchers to use ELMo embeddings in downstream natural language processing tasks, they often have to fine-tune a task-specific model built atop pre-trained ELMo embeddings. This makes it possible for the model to learn the given job while s imultaneously utilizing the contextualized word embeddings that Elmo has generated.

For tasks like information retrieval, computational linguistics, and semantic segmentation recognition [64], ELMo has achieved performance levels that are considered cutting-edge. In addition, it is beneficial for vocations that need an understanding of the context in which a word or phrase is employed.

Since the advent of ELMo, there have been various more contextualized word embedding models created, including GPT and BERT [65, 50]. Even though ELMo remains a popular choice for NLP applications that require fine-grained contextual information.

3.2. Deep Learning Techniques

3.2.1. LSTM

Long-Term Short-Term Memory (LSTM) is a solution to the issues of vanishing and exploding gradients that are present in normal RNNs. The critical feature of LSTM [66] is its ability to selectively remember or forget information over long periods, making it especially effective for tasks that require the model to retain information over a sequence of inputs. This is achieved through memory cells, which allow the model to store and update information over time.

In 1997, Hochreiter and Schmidhuber were the ones that popularized the use of LSTMs, which have since gone on to become a well-liked option for various applications. They have been used in multiple applications, including speech recognition, translation, image captioning, etc.

Recently, there have been several advancements in the field of LSTMs. One study published in 2021 proposed a new type of LSTM called the “coupled LSTM,” which outperformed traditional LSTMs on several tasks. Another study published in 2022 introduced a novel LSTM-based architecture called “GLSTM,” which used a gated linear unit (GLU) activation function [67].

Overall, LSTMs remain a popular and powerful tool for processing sequential data, and ongoing research continues to explore new variations and improvements to the architecture.

3.2.2. Bidirectional LSTM

The standard LSTM design is expanded in Bidirectional LSTM (BLSTM) [68], which processes the input sequence in both the forward and backward directions. Because of this, the model can capture dependencies between the current input and previous inputs, and future inputs, making it particularly useful for tasks that require a deeper understanding of the input sequence, such as speech recognition, natural language processing, and video analysis.

The input sequence is split into two distinct LSTM layers in a BLSTM [69], with one layer processing the line in the forward direction and the other in reverse. After the completion of each layer, the results are concatenated and sent to the subsequent layer.

BLSTMs were first introduced by Schuster and Paliwal in 1997 and have since become a popular choice for a wide range of applications. They have been used for speech recognition, language translation, image captioning, and many other tasks.

Current developments in BLSTMs have primarily focused on enhancing the devices' levels of both efficiency and accuracy. One study that was published in 2021 presented a new sort of BLSTM that was referred to as the "depth-wise bidirectional LSTM." This implementation of the BLSTM takes advantage of depth-wise separable convolutions to reduce the total number of parameters and improve the model's computational efficiency. Another study was conducted and released in 2022, and it introduced a breakthrough BLSTM-based architecture that was given the term "FusionLSTM." Its architecture combines convolutional and recurrent layers to provide state-of-the-art performance on various natural language processing applications [22,70].

Overall, BLSTMs remain a powerful tool for processing sequential data, and ongoing research continues to explore new variations and improvements to the architecture.

3.2.3. Multi-dense LSTM Model

A Multi-dense LSTM Model in deep learning is a kind of NN architecture that combines multiple dense (fully connected) and LSTM layers. The thick layers process the input features and extract useful information. In contrast, the LSTM layers handle sequential data and preserve the memory of past events to inform future predictions. The combination of dense and LSTM layers enables the Multi-dense LSTM Model to effectively capture both non-sequential and sequential relationships within the data, making it a powerful tool for tasks that involve both types of information, such as sentiment analysis and time series prediction.

3.2.4. GRU

In deep learning, GRUs are a specific sort of RNN. The GRU was developed to process sequential input and to keep a persistent recollection of events that have occurred in the past to influence predictions. In contrast to more conventional LSTM and RNN systems, GRUs are equipped with two gates that govern data flow into and out of the hidden state. This makes GRUs more computationally efficient and easier to train while effectively capturing the dependencies between elements in sequential data. GRUs have been implemented for numerous tasks in NLP, speech recognition, and time series prediction.

3.2.5. Bidirectional GRU

Bidirectional GRU (BiGRU) is a variant of the GRU network in deep learning that processes sequential information in both forward & backward directions. A BiGRU network reads the input sequence forwards and backward and concatenates the resulting hidden states to represent the input more comprehensively. This bidirectional processing enhances the ability of BiGRUs to capture contextual information and dependencies in sequences, making them useful in a scale of sequential data processing tasks, including sentiment analysis & speech recognition. By combining the strengths of GRUs and bidirectional processing, BiGRUs offer a more powerful and flexible solution for handling sequential data compared to traditional GRUs or other types of RNNs.

3.3. Ensemble and Hybrid Modeling

Ensemble and Hybrid Modeling are popular techniques for developing more accurate models for detecting fake reviews.

3.3.1. Ensemble Modeling

The predictions of several different models are combined in ensemble modelling, which results in a more reliable and accurate model. The premise underlying ensemble modelling is that by integrating several other models, the benefits of each model can be maximized, and the drawbacks of the models may be reduced.

There are various types of ensemble models, such as:

- ***Voting Classifier:*** By taking a majority vote, a Voting Classifier combines the predictions of numerous models to arrive at a single conclusion, which is the most frequently reached by all models.

- ***Bagging:*** Bagging stands for Bootstrap Aggregating, which involves creating multiple samples of the training dataset and training multiple models on each of these samples. The overall forecast considers all the models' predictions and uses either their average or majority vote.
- ***Boosting:*** Boosting is a method that trains a sequence of models progressively, with each succeeding model attempting to fix the faults caused by the model that came before it.
- ***Stacking:*** Stacking is a technique that involves training numerous models and applying their predictions as input characteristics to a higher-level model. Stacking is also known as "stacking up."

Ensemble modelling effectively improves the performance of fake review detection models, as it can help overcome the limitations of individual models and reduce bias.

3.3.2. Hybrid Modeling

Hybrid Modeling involves combining multiple types of models or various data sources to create a more accurate model. For fake review detection, mixed Modeling can be achieved by combining various features, such as text-based features, metadata features (such as review rating, timestamp, and reviewer profile information), and behavioural features (such as click stream data and purchase history).

Hybrid Modeling can also involve combining multiple models, such as text-based and metadata-based models, and using both models' outputs as input features to a higher-level model.

A model that is constructed via the use of hybrid Modeling seems to have the capability to be more precise and comprehensive since it relies on the most beneficial parts of a variety of different models and data sources. This is because hybrid Modeling draws upon the most practical aspects of several different models. On the other hand, building it and keeping it up might be more difficult, necessitating a more significant expenditure of resources to compensate for the increased difficulty brought on by the greater complexity.

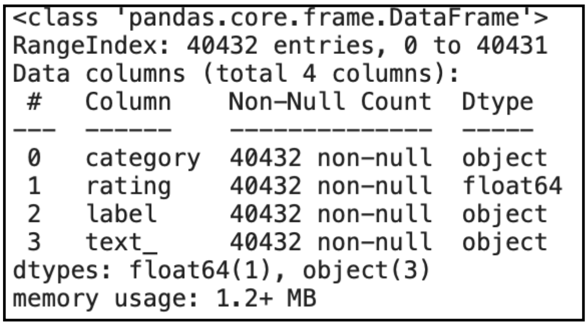
Overall, both Ensemble Modeling and Hybrid Modeling are effective techniques for improving the accuracy and robustness of fake review detection models. The approach chosen depends on the specific problem and available data, as well as the resources and expertise available for model development.

4. FLOW OF WORK

Designing a fake review detection model using deep learning involves several steps, which can be summarized as follows:

4.1. Collection of Raw Data

The first step is to store an extensive database of fake and genuine reviews. The dataset should be balanced, i.e., containing an equal number of fake and real reviews. The dataset can be obtained from websites like Amazon, Yelp, TripAdvisor, and OSF. The OSF Fake Review Dataset (https://osf.io/tyue9/)is used for this research work.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40432 entries, 0 to 40431
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   category  40432 non-null  object
 1   rating    40432 non-null  float64
 2   label     40432 non-null  object
 3   text_     40432 non-null  object
dtypes: float64(1), object(3)
memory usage: 1.2+ MB
```

Fig. 2. Data Information

4.2. Data Pre-processing

The information gathered will need to be preprocessed after this stage. This entails organising the data, removing any extra details, and converting the textual material into numerical representations that could be input into a DL design. Tokenization, stemming, and vectorisation are three methods that could be used to accomplish this goal.

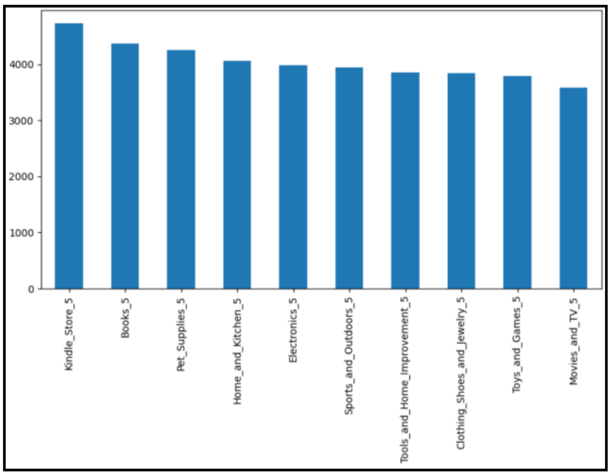


Fig. 3. Reviews Categories

4.2.1. Sentence Tokenization

The entire review is provided as input, and through the NLTK software, it is tokenized into phrases.

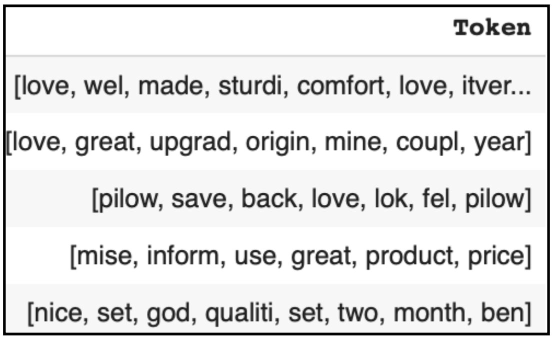
4.2.2. Removal of Punctuation Marks and Repeating Words

There is no longer any punctuation at the beginning or conclusion of the comments, as well as more white areas. For example:

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

4.2.3. Word Tokenization

To facilitate recovery, each individual review is tokenized into words and saved in a list.

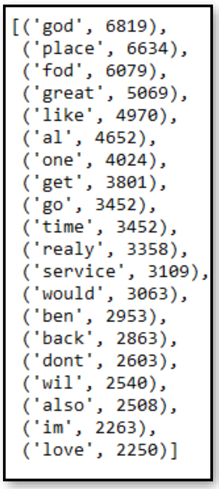


Token
[love, wel, made, sturdi, comfort, love, itver...
[love, great, upgrad, origin, mine, coupl, year]
[pilow, save, back, love, lok, fel, pilow]
[mise, inform, use, great, product, price]
[nice, set, god, qualiti, set, two, month, ben]

Fig. 4. Tokenization

4.2.4. Description of Text Information

A total of 1276605 words is used for this work, with a vocabulary size of 34355, and the max sentence length is 256. Given below is the list of the top 20 words obtained after tokenization.



```
[('god', 6819),
 ('place', 6634),
 ('fod', 6079),
 ('great', 5069),
 ('like', 4970),
 ('al', 4652),
 ('one', 4024),
 ('get', 3801),
 ('go', 3452),
 ('time', 3452),
 ('realy', 3358),
 ('service', 3109),
 ('would', 3063),
 ('ben', 2953),
 ('back', 2863),
 ('dont', 2603),
 ('wil', 2540),
 ('also', 2508),
 ('im', 2263),
 ('love', 2250)]
```

Fig. 5. Most Frequent Words

4.2.5. Exploratory Data Analysis:

EDA is a procedure that involves summarizing the data through statistical and visualization tools to bring the most significant aspects of the data into focus for further analysis. This requires analyzing the dataset from various perspectives, characterizing it, and summarizing its findings without making any presumptions about the information it contains. The frequency distribution of labels and Top words in the corpus are shown in Figures 7 and 8, respectively.

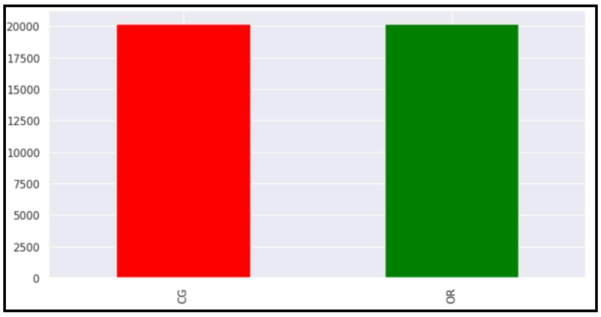


Fig. 6. Frequency Distribution of Label

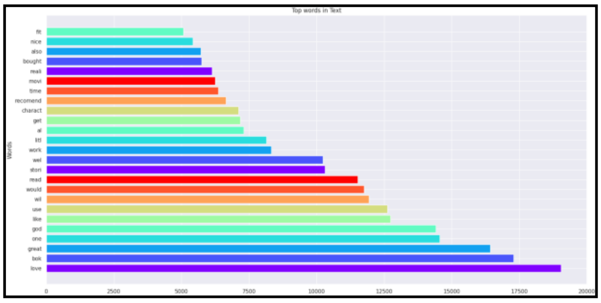


Fig. 7. Top Words in Text

A word cloud, a tag cloud, or a text cloud is a graphical representation of how frequently and prominently specific terms appear in a given body of text. The presentation of the words is frequently arbitrary, with the size of each word being proportional to the number of points of time it appears in the text. The most frequently occurring words in the text are usually displayed in larger font sizes, while less frequent words are shown in smaller font sizes. WC is often used for text analysis, as they provide a quick and intuitive overview of the content of a document and highlight the most important topics and keywords. Word clouds can be created using various software tools and online services, and they can be customized with different colours, shapes, and font styles to make them more visually appealing. Figures 9 and 10 show word clouds for fake reviews and genuine reviews.

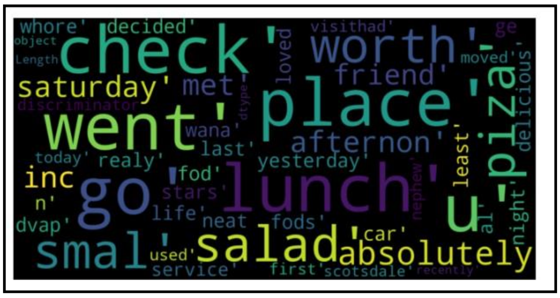


Fig. 8. Word Cloud for Label 0

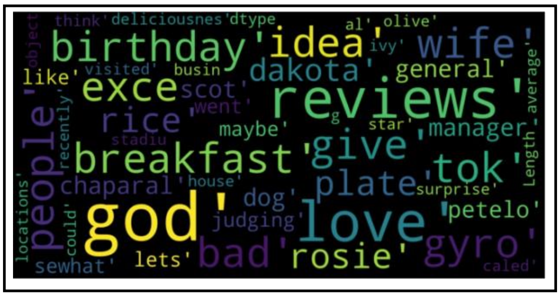


Fig. 9. Word Cloud for Label 1

4.3. Selection of Model

In the first step of this process, we analyse the results obtained from eight different machine-learning models to identify fake reviews. These models are as follows: Gaussian, Multinomial, Bernoulli, Complement Naive Bayes Classifiers, k-nearest neighbors, Logistic Regression, Stochastic Gradient Descent (SGD), and Random Forest. The subsequent step is to select an appropriate deep-learning model to detect false reviews, which brings us to the final stage. Second, five different DL models, including LSTM, Bi-LSTM, Multi-dense LSTM, GRU, and Bidirectional GRU, have been applied to the same corpus for this work.

4.4. Feature Extraction

Feature extraction is the practice of finding and extracting useful information from the review data so that it can always train the model. It is an important stage in constructing a deep-learning model for detecting fake reviews.

Many researchers have explored different feature extraction techniques in this field, including sentiment analysis, sentiment polarity, text length, word frequency, and n-grams, among others. For instance, in the paper [23], the authors use sentiment analysis to remove features from online reviews.

TfidfTransformer & TfidfVectorizer are two popular NLP tools used in this research work for feature extraction and representation of text data. They are part of the scikit-learn library in Python.

4.4.1. TfidfVectorizer

TfidfVectorizer, on the other hand, combines both the feature extraction and feature representation steps into a single function. The output is Tf-Idf vectors, and the raw text data is used as the input. Tfidf Vectorizer has an advantage over TfidfTransformer because it handles the text pre-processing and cleaning steps, such as removing stop words, stemming and lemmatizing, and converting the text into numerical form. In addition, TfidfVectorizer can convert text into a vector format. To determine the tf-idf use the given equations:

$$tf-idf(t,d)=tf(t,d)*\log\frac{n}{df(t)+1}$$

When smooth_idf=True, which is also the default setting. In this equation:

- The set of term occurrences within the specified document is denoted by the notation tf(t, d). This is identical to the results that the Count Vectorizer produced for us.
- The number of documents that have been mainly carried is n.
- The phrase "document frequency," abbreviated as "df," refers to the percentage of complete texts in the corpus of documents that contain the word "t."
- Adding each to the denominator in the above formula reduces the weight of words with an idf of zero or terms included in every document that makes up the training set. Each row is modified to possess a unit Euclidean norm (by dividing the l2 bar of itself).

4.4.2. Tfidf Transformer

The Tfidf Transformer is needed to convert raw count-based feature vectors into corresponding Tf-Idf representations. A statistical metric that depicts the significance of a term within a document is referred to as the Term Frequency-Inverse Document Frequency (Tf-Idf for short). The Tf-Idf score of a word is determined by looking at its frequency in the given document and its rarity across the entire corpus.

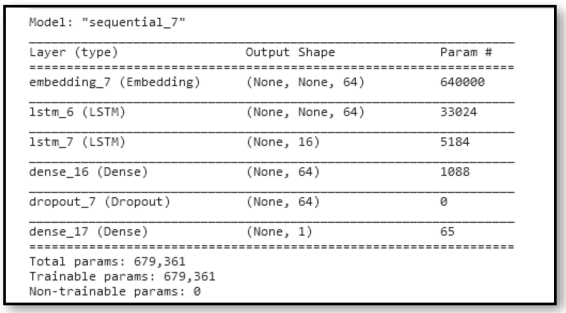
Applications like text clustering, and sentiment analysis, extensively use Tfidf Transformer and Tfidf Vectorizer.

Tokens naturally occurring frequently in a document corpus are given less weight in the analysis than tokens that appear in a much smaller proportion of the training corpus. This is because the TF-IDF metric scales characteristics rather than the natural frequency of token occurrences in a document corpus.

4.5. Testing and Training

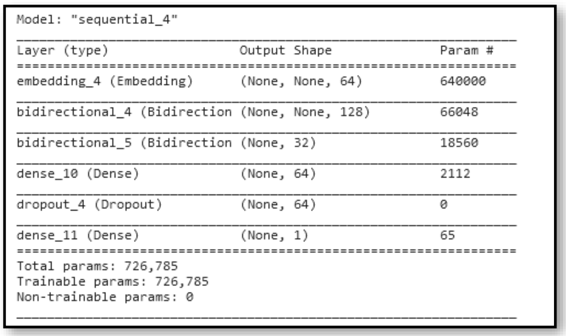
In the processes of feature extraction, training, and testing, data are employed. A model is then trained and evaluated based on these data. The design is prepared using the training data by adjusting its variables to reduce the error caused by making predictions using the training data. As this is going on, an analysis of the performance of the trained design is being carried out based on the information gathered from the testing.

The model is evaluated using these data to determine how well it can generalize to data it has not seen before and how accurately it can predict the outcomes of future experiments. To avoid overfitting, keeping the testing data distinct from the training data is standard practice.



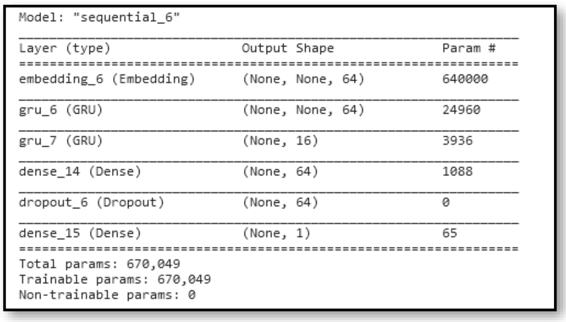
```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_7 (Embedding)      (None, None, 64)          640000
_________________________________________________________________
lstm_6 (LSTM)                (None, None, 64)          33024
_________________________________________________________________
lstm_7 (LSTM)                (None, 16)                5184
_________________________________________________________________
dense_16 (Dense)             (None, 64)                1088
_________________________________________________________________
dropout_7 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_17 (Dense)             (None, 1)                 65
=================================================================
Total params: 679,361
Trainable params: 679,361
Non-trainable params: 0
```

Fig. 10. LSTM



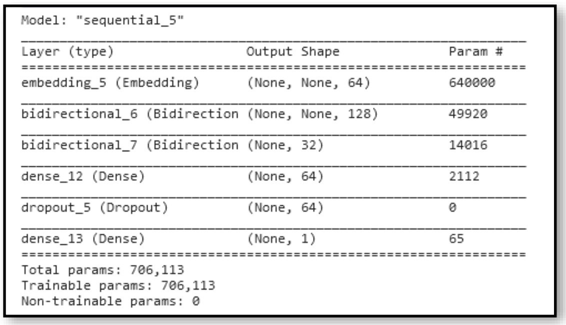
```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_4 (Embedding)      (None, None, 64)          640000
_________________________________________________________________
bidirectional_4 (Bidirection (None, None, 128)         66048
_________________________________________________________________
bidirectional_5 (Bidirection (None, 32)                18560
_________________________________________________________________
dense_10 (Dense)             (None, 64)                2112
_________________________________________________________________
dropout_4 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_11 (Dense)             (None, 1)                 65
=================================================================
Total params: 726,785
Trainable params: 726,785
Non-trainable params: 0
_________________________________________________________________
```

Fig. 11. Bidirectional LSTM



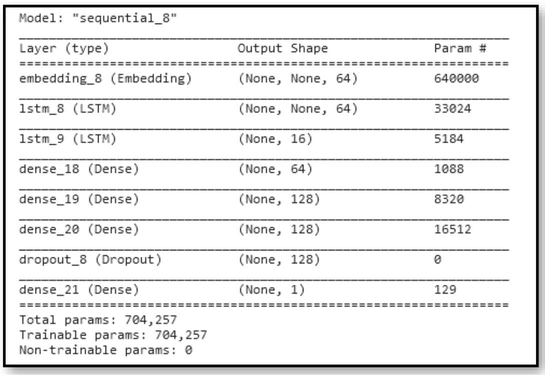
```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_6 (Embedding)      (None, None, 64)          640000
_________________________________________________________________
gru_6 (GRU)                  (None, None, 64)          24960
_________________________________________________________________
gru_7 (GRU)                  (None, 16)                3936
_________________________________________________________________
dense_14 (Dense)             (None, 64)                1088
_________________________________________________________________
dropout_6 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_15 (Dense)             (None, 1)                 65
=================================================================
Total params: 670,049
Trainable params: 670,049
Non-trainable params: 0
```

Fig. 12. GRU



```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_5 (Embedding)      (None, None, 64)          640000
_________________________________________________________________
bidirectional_6 (Bidirection (None, None, 128)         49920
_________________________________________________________________
bidirectional_7 (Bidirection (None, 32)                14016
_________________________________________________________________
dense_12 (Dense)             (None, 64)                2112
_________________________________________________________________
dropout_5 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_13 (Dense)             (None, 1)                 65
=================================================================
Total params: 706,113
Trainable params: 706,113
Non-trainable params: 0
```

Fig. 13. GRU Bidirectional



```
Model: "sequential_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_8 (Embedding)      (None, None, 64)          640000
_________________________________________________________________
lstm_8 (LSTM)                (None, None, 64)          33024
_________________________________________________________________
lstm_9 (LSTM)                (None, 16)                5184
_________________________________________________________________
dense_18 (Dense)             (None, 64)                1088
_________________________________________________________________
dense_19 (Dense)             (None, 128)               8320
_________________________________________________________________
dense_20 (Dense)             (None, 128)               16512
_________________________________________________________________
dropout_8 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_21 (Dense)             (None, 1)                 129
=================================================================
Total params: 704,257
Trainable params: 704,257
Non-trainable params: 0
```

Fig. 14. Multi-Dense LSTM

When the training and testing phases of the model have been completed, the artificial intelligence system that will be used to identify false reviews can then be put into production.

5. RESULTS AND DISCUSSIONS

5.1. Dataset Used

In the proposed mythology, we use the Amazon Review Data (2021) dataset housed on the OSF platform and accessible to the general public. This dataset is comprehensive and reliable (Joni Salminen et al., 2021). The Open Science Framework (OSF) is a software project that is free and open-source, and it encourages open cooperation in scientific research. There are 40,000 genuine product reviews and 40,000 fraudulent reviews in the dataset constructed with fake reviews. Original Reviews (supposedly authored by humans and real) = OR; Fake Reviews Generated by Computers = CG.

The Amazon Dataset is a popular dataset for research and development in NLP & SA. It provides researchers with a large and diverse collection of customer reviews and feedback, which is essential for developing and evaluating advanced algorithms in natural language processing and sentiment analysis.

It consists of millions of reviews of local businesses, such as restaurants, bars, and shops, along with metadata such as user ratings, location, and business categories. The dataset provides a rich and diverse collection of customer reviews and feedback. The Amazon Dataset has been widely used for various tasks in NLP, including sentiment analysis, text classification, and recommendation systems. I

5.2. Performance Metrics

Instead of using the raw frequencies of the occurrence of a token in a particular document, TF-IDF is utilized since it helps minimize the effect of tokens that commonly arise in a corpus and are, as a consequence, experimentally less impactful than features that seem so in a small chunk of the original dataset.

5.2.1. Accuracy

The degree to which the model can correctly forecast positive and negative outcomes is called its accuracy.

5.2.2. Precision

The percentage of correct optimistic forecasts made by a model compared to the overall number of optimistic predictions generated by the model is referred to as the model's precision. Expected positive cases are another term for accurate optimistic forecasts. To explain it, we use the proportion of accurate or inaccurate positive predictions that it bears about the overall number of such forecasts.

Precision=True Positive/
(True Positive + False Positive)

5.2.3. Recall

The recall is a measurement that determines the percentage of actual positive cases correctly recognized by the model as positive. It is described as the proportion of precise prediction to the total of both accurate or erroneous predictions.

Recall=True Positive/
(True Positive + False Negative)

5.2.4. F1-Score

F1-score is a weighted average of precision & recall and is utilized to balance the trade-off between the two metrics. As the harmonic mean of recall & precision, it is so named. The F1-score provides a single value that summarizes the model's performance, taking into account both the ability of the design to identify positive cases correctly & the ability to avoid FP predictions.

F1 Score=2□ (Precision □Recall)/
(Precision
+Recall)

The proportions of data considered positive and negative, respectively, are referred to as true positive (TP) and true negative (TN). Data labeled as positive but negative is referred to as having a false positive (FP), and data labelled as negative but positive is referred to as having a false negative (FN). The recall is the percentage of times that the system successfully detects tags. The F1 score is arrived at by taking the average of the recall and precision scores [1].

5.3. Algorithm Selection and Experimental Procedure:

Throughout our investigation, we chose to utilize a total of eight distinct machine-learning models to spot fake reviews. The names of these models are as follows: Gaussian, Multinomial, Bernoulli, Complement Naive Bayes Classifiers, k-nearest neighbours, Logistic Regression, Stochastic

Gradient Descent (SGD), and Random Forest. After finishing the investigation, it was found that the algorithms LR and SGD had a greater accuracy, with a score of 82.51 and 82.70, respectively.

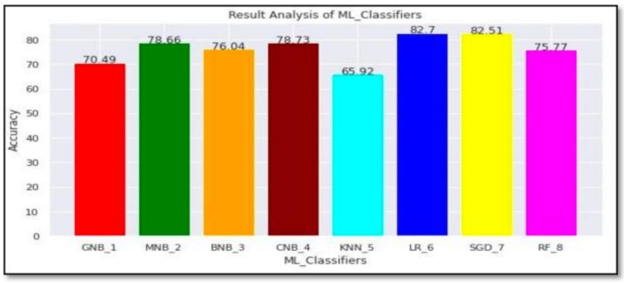
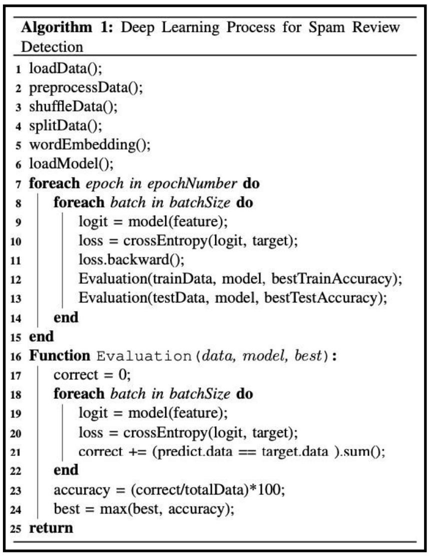


Fig. 15. Accuracy Graph of ML Classifiers

Deep Learning systems require less human engagement than traditional ones since numerous layers of neural networks process the input. These networks eventually learn from their mistakes and setbacks, reducing the amount of human oversight required. For this study, deep learning was selected over machine learning because the input database provided to the machines substantially impacts the model's accuracy.



```
Algorithm 1: Deep Learning Process for Spam Review Detection
loadData();
preprocessData();
shuffleData();
splitData();
wordEmbedding();
loadModel();
foreach epoch in epochNumber do
    foreach batch in batchSize do
        logit = model(feature);
        loss = crossEntropy(logit, target);
        loss.backward();
        Evaluation(trainData, model, bestTrainAccuracy);
        Evaluation(testData, model, bestTestAccuracy);
    end
end
Function Evaluation(data, model, best):
    correct = 0;
    foreach batch in batchSize do
        logit = model(feature);
        loss = crossEntropy(logit, target);
        correct += (predict.data == target.data ).sum();
    end
    accuracy = (correct/totalData)*100;
    best = max(best, accuracy);
return
```

Fig. 16. Deep Learning Algorithm

When the data collection is relatively small, ML Models are the method of choice to analyze it. The use of DL designs is recommended whenever the data collection in consideration is enormous. In addition, it is affected by the quality of the information presented during training. Even with relatively small data sets, ML models might produce unsatisfactory results if the feature engineering process is not carried out appropriately. Second, this work applied five distinct DL models, including LSTM, Bi-LSTM, Multi-dense LSTM, GRU, and Bidirectional GRU, to the same corpus.

Models	Accuracy	Loss	RMSE Values
Bidirectional LSTM	99.9	0.001	0.031623
LSTM	90.71	0.20	0.447214
GRU	96.24	0.37	0.608276
Bidirectional GRU	98.78	0.04	0.2
Multi Dense LSTM Model	91.69	0.45	0.67082

Table 1. Training Results

Models	Accuracy	Loss	RMSE Values
Bidirectional LSTM	85.59	1.82	1.349074
LSTM	80.62	0.50	0.707107
GRU	88.00	0.61	0.781025
Bidirectional GRU	78.90	1.30	1.140175
Multi Dense LSTM Model	84.62	0.59	0.768115

Table 2. Validation Results

Models	Categories	Precision	Recall	F1 Score
Bidirectional LSTM	0	77	72	70
	1	89	86	87
LSTM	0	66	71	78
	1	88	85	87
GRU	0	85	70	63
	1	73	90	74
Bidirectional GRU	0	79	75	78
	1	84	86	85
Multi Dense LSTM Model	0	96	76	46
	1	96	76	85

Table 3. Category-wise Validation Results

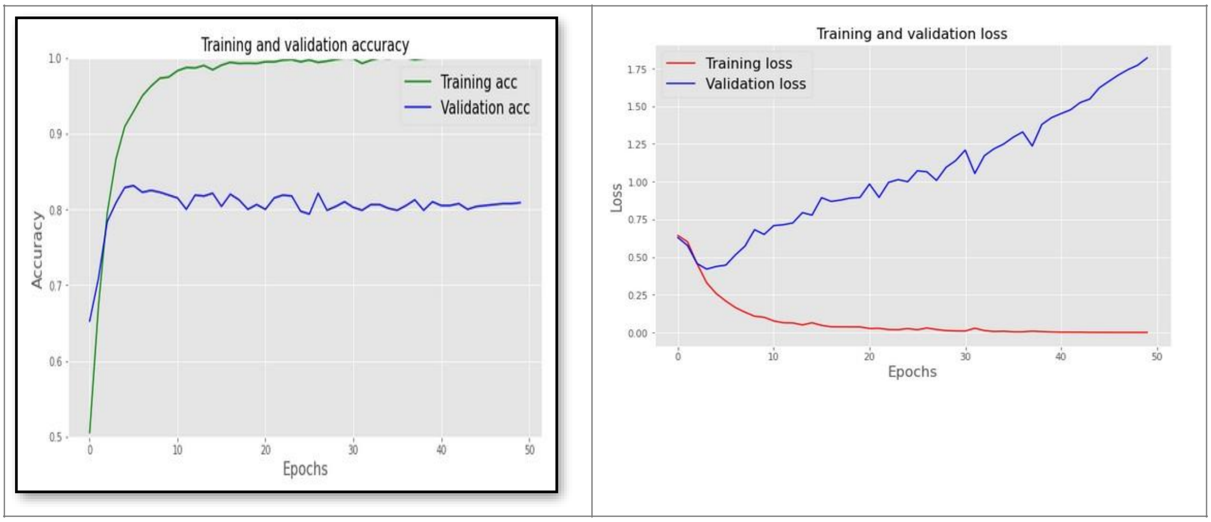


Fig. 17. Accuracy and Loss Graph for LSTM Bidirectional

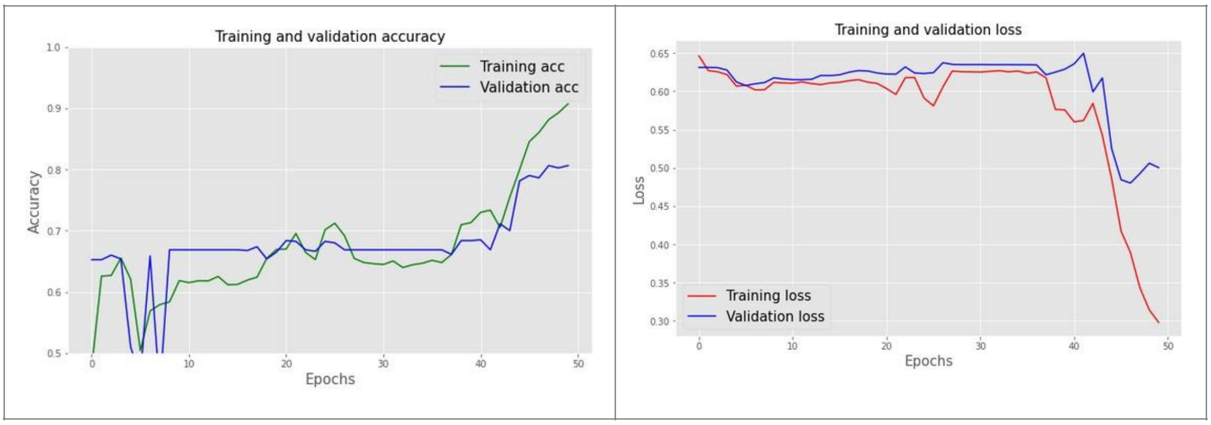


Fig. 18. Accuracy and Loss Graph for LSTM Model

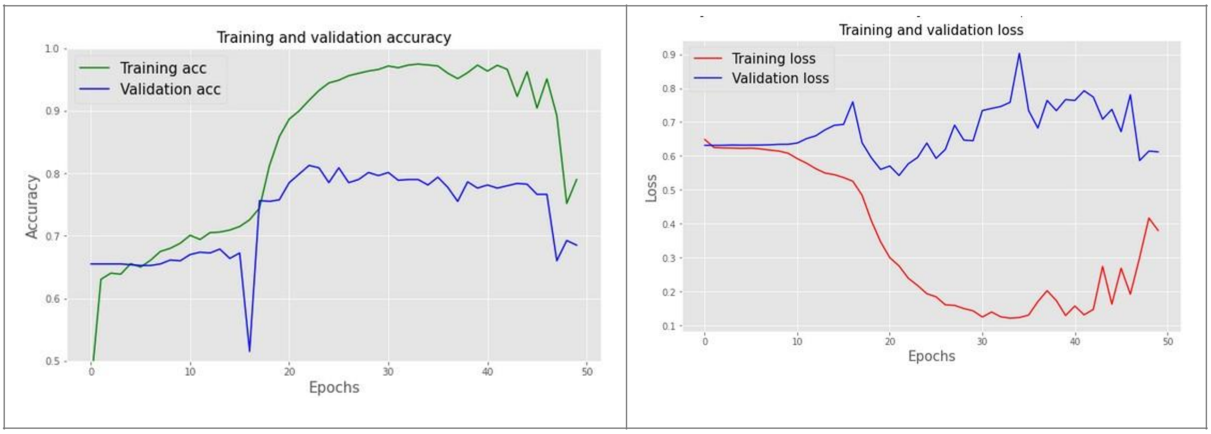


Fig. 19. Accuracy and Loss Graph for GRU Model

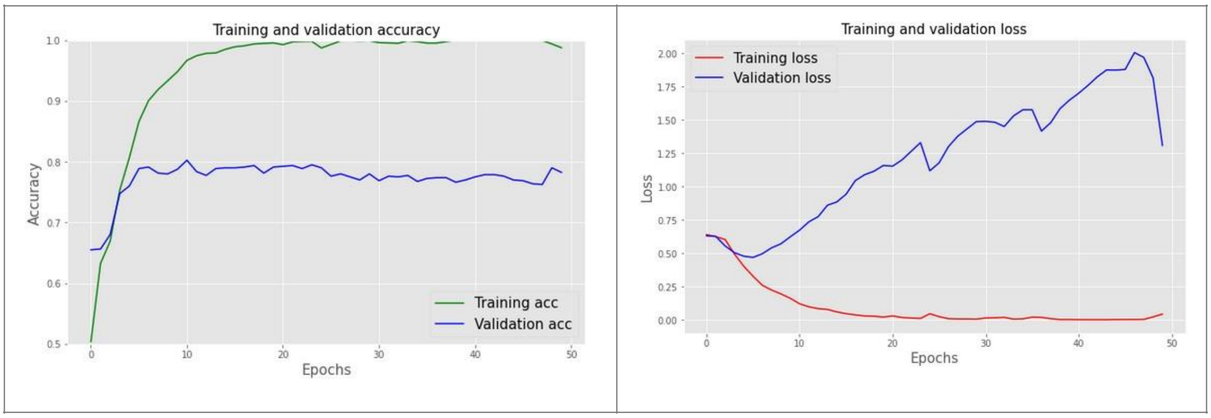


Fig. 20. Accuracy and Loss Graph for Bidirectional GRU Model

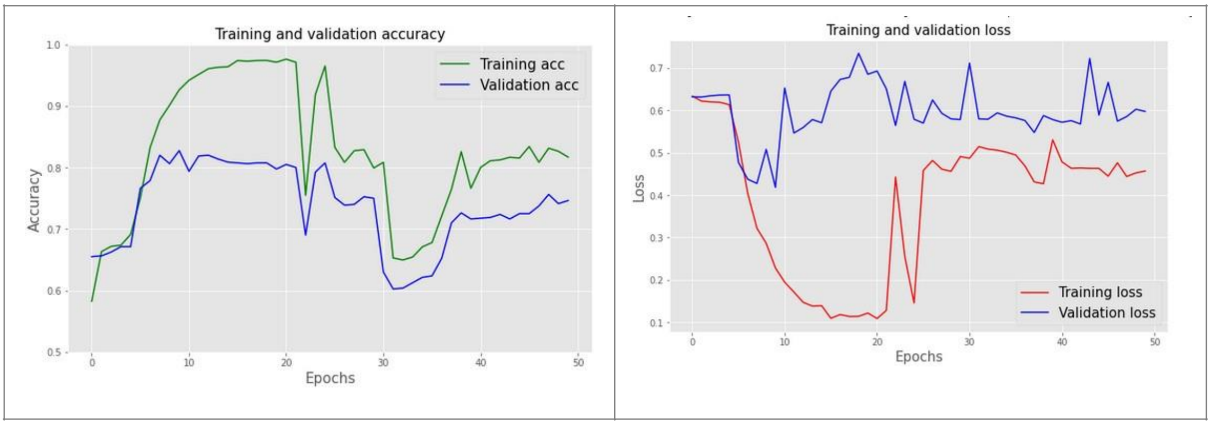


Fig. 21. Accuracy and Loss Graph for Multi-Dense LSTM Model

According to the findings, out of the five different DL models tested, including LSTM, Bi-LSTM, Multi-dense LSTM, GRU, and Bidirectional GRU, the Bi-LSTM outperforms best (Epochs=50, Training and validation) the rest of the classifiers in terms of accuracy. This was determined by comparing the Bi-LSTM to the other classifiers.

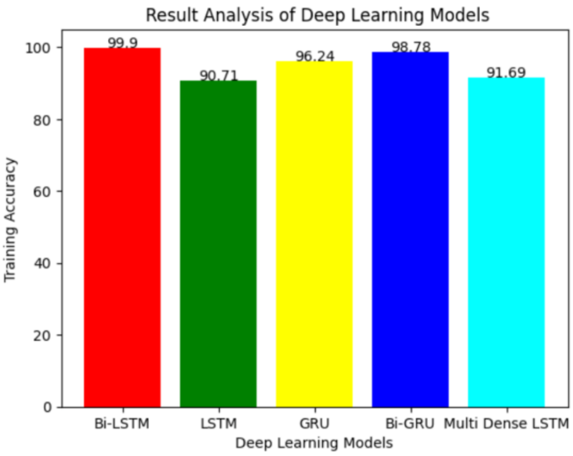


Fig. 22. Accuracy Graph of Deep Learning Models

Compared to the other classifiers utilized by Joni Salminen et al., 2021, which implemented fewer Epochs than our model, our results indicate that the accuracy of the Bi-LSTM for detecting fake reviews is slightly higher than what is obtained when comparing the Bi-LSTM to the other classifiers.

6. CONCLUSION AND FUTURE SCOPE

Identifying fake reviews is challenging for researchers, e-commerce websites, and companies conducting business online. According to the findings of our study, the text generation technologies currently in use produce fake evaluations with an appearance that is so similar to the actual thing that it is difficult for a person to spot them. This paper proposes a detailed comparison investigation based on detecting fake reviews to date utilising machine learning and deep learning algorithms. First, we have examined the many feature extraction methods researchers have used. After that, we analysed the existing datasets in-depth, detailing their creation processes. Following that, we provided an overview of some classic machine learning methods in addition to neural network models that have been applied to identify fraudulent reviews using summary tables.

Improving feature extraction and classifier construction through traditional statistical machine learning helps boost the overall performance of text classification models. On the other hand, deep learning can enhance performance by using the hybrid modelling paradigm better. The development of text enhances columns will be the primary focus of our future study. To do this, we will augment the currently available dataset with a polarity feature and perform ensemble modelling for novelty reasons.

REFERENCES

[1] X. Zhang and Y. Wang, "A Survey on Fake Reviews Detection: Techniques, Datasets and Tools," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 5, pp. 4073-4094, Oct. 2020.

[2] J. Z. Hu and B. Liu, "Mining Opinion Features in Customer Reviews," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 758-763, 2004.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, May 2015.

[4] M. Alhoshani, A. Aljarah, and A. Alkhateeb, "Fake Reviews Detection Using Deep Learning Techniques," *Proceedings of the 2020 International Conference on Computer, Information and Telecommunication Systems*, pp. 1-6, Oct. 2020.

[5] J. Kim, B. Kwon, and H. Kim, "Fake Review Detection based on Deep Learning," *Proceedings of the International Conference on Information and Communication Technology*, pp. 722-725, Nov. 2016.

[6] X. Liu, Y. Liu, and H. Wang, "Sentiment Analysis and Opinion Mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1-167, 2012.

[7] I. Konstas and V. Lampos, "Overview of Text Classification," *ACM Computing Surveys (CSUR)*, vol. 50, no. 1, pp. 1-36, Mar. 2017.

[8] S. Kim, J. Kim, and H. Kim, "An Analysis of Fake Reviews Detection Using Machine Learning Algorithms," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 5, pp. 4211-4226, Oct. 2020.

[9] J. Park and S. Kim, "Fake Review Detection using Artificial Neural Networks," *Expert Systems with Applications*, vol. 44, no. 1, pp. 309-318, Jan. 2016.

[10] Y. Sun, Y. Liu, and X. Liu, "Opinion Spam Detection with Deep Learning," *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 3, pp. 1-32, Apr. 2019.

[11] J. Li, X. Liu, and H. Wang, "A Deep Learning Framework for Fake Review Detection," *Journal of Computer Science and Technology*, vol. 33, no. 4, pp. 661-676, Jul. 2018.

[12] S. K. Kim, J. Y. Kim, and H. Kim, "Deep Learning Based Fake Review Detection Using Multi-Granular Attention Mechanism," *Expert Systems with Applications*, vol. 153, pp. 1-12, Nov. 2021.

[13] X. Wang, Y. Liu, and X. Liu, "A Deep Transfer Learning Approach for Fake Review Detection," *Knowledge-Based Systems*, vol. 220, pp. 1-14, Nov. 2020.

[14] M. Al-Hazmi, M. Al-Hazmi, and K. Al-Khalifa, "A Deep Convolutional Neural Network for Fake Review Detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 5, pp. 8659-8676, May 2021.

[15] R. Bhatia, S. Chawla, and S. Bhatnagar, “Detection of fake reviews using deep learning,” *in Proceedings of the IEEE International Conference on Advances in Computing, Communications and Informatics*, pp. 1527–1531, 2017.

[16] J. Li, L. Liu, J. Wang, and X. Liu, “Detection of fake hotel reviews based on deep learning,” *in Proceedings of the IEEE International Conference on Computer and Communications*, pp. 1–6, 2017.

[17] J. Ma, H. Wang, X. Liu, and J. Zhao, “Fake review detection based on deep learning and sentiment analysis,” *in Proceedings of the IEEE International Conference on Web Services*, pp. 112–117, 2018.

[18] J. Ma, J. Wang, X. Liu, and J. Zhao, “Fake review detection based on deep learning and sentiment analysis,” *Expert Systems with Applications*, vol. 92, pp. 64–74, 2017.

[19] J. Li, L. Liu, and X. Liu, “Sentiment analysis and fake review detection based on deep learning,” *International Journal of Web Services Research*, vol. 14, no. 4, pp. 1–15, 2017.

[20] Zhang, D., Li, W., Niua, B., & Wuc, C., “A deep learning approach for detecting fake reviewers: Exploiting reviewing behavior and textual information,” *Decision Support Systems*, Nov. 2022.

[21]Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, ‘‘RoBERTa: A robustly optimized BERT pretraining approach,’’ 2019, arXiv:1907.11692. [Online]. Available: http://arxiv.org/abs/1907.11692

[22] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K., “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018. arXiv preprint arXiv:1810.04805.

[23] Aslam, Andleeb; Qamar, Usman; Saqib, Pakizah; Ayesha, Reda; Qadeer, Aiman., “A Novel Framework For Sentiment Analysis Using Deep Learning,” *2020 22 nd International Conference on Advanced Communication Technology (ICACT)*, pp. 525–529, 2020. doi:10.23919/ICACT48636.2020.9061247

[24] Christopher G. Harris, “Detecting Fake Yelp Reviews Using a Positional LSTM / K-L Divergence Ensemble Approach”, *1st International Conference on Information System & Information Technology (ICISIT)*,2022.

[25] M.N. Istiaq Ahsan, Tamzid Nahian, Abdullah All Kafi, Md. Ismail Hossain, Faisal Muhammad Shah, “An ensemble approach to detect review spam using hybrid

machine learning technique", *19th International Conference on Computer and Information Technology (ICCIT)*,2016

[26] Navya Singh; Rohit Kumar Kaliyar; Thoutam Vivekanand; Kumar Uthkarsh; Vipul Mishra; "B-LIAR: A novel model for handling Multiclass Fake News data utilizing a Transformer Encoder Stack- based architecture", *1st International Conference on Informatics (ICI)* ,2022.

[27] Hua Jiang, Chengyu Hu, Feng Jiang, "Text Sentiment Analysis of Movie Reviews Based on Word2Vec-LSTM", *14th International Conference on Advanced Computational Intelligence (ICACI)* , 2022.

[28] Putra Fissabil Muhammad, Retno Kusumaningrum, Adi Wibowo, " Sentiment Analysis Using Word2vec And Long Short-Term Memory (LSTM) For Indonesian Hotel Reviews ", *Procedia Computer Science,* 2021.

[29] Madhuri V. Joseph, "A Bi-LSTM and GRU Hybrid Neural Network with BERT Feature Extraction for Amazon Textual Review Analysis", *International Journal of Engineering Trends and Technology,* Volume70 Issue 5,2022.

[30] Muhammad Saad Javed, Hammad Majeed, Hasan Mujtaba, Mirza Omer Beg, Fake reviews classification using deep learning ensemble of shallow convolutions",*Journal of Computational Social Science*, 2021.

[31]Bundit Manaskasemsak, Jirateep Tantisuwankul, Arnon Rungsawang, " Fake review and reviewer detection through behavioral graph partitioning integrating deep neural network ", *Neural Computing and Applications,* 2021.

[32] Dibyajyoti Baishya, Joon Jyoti Deka, Gaurav Dey & Pranav Kumar Singh , "SAFER: Sentiment Analysis-Based FakE Review Detection in E-Commerce Using Deep Learning",*SN Computer Science*, Vol. 2, 2021.

[33] I. KadekSastrawanI.P.A.BayupatiDewa Made SriArsa, "Detection of fake news using deep learning CNN–RNN based methods", *ICT Express* , Volume 8, Issue 3, Pages 396-408,2022.

[34] Gourav Bathla, Pardeep Singh, Rahul Kumar Singh, Erik Cambria & Rajeev Tiwari, "Intelligent fake reviews detection based on aspect extraction and analysis using deep learning", *Neural Computing and Applications*, volume 34, pp. 20213–20229, 2022.

[35] Abrar Qadir Mir, Furqan Yaqub Khan, Mohammad Ahsan Chishti, "Online Fake Review Detection Using Supervised Machine Learning And Bert Model", *IEEE*, 2019.

[36] David Refaeli,Petr Hajek, "Detecting Fake Online Reviews using Fine-tuned BERT", *ICEBI 5th International Conference on E-Business and Internet,* Pages 76–80,2021.

[37]Umer Hayat, Ali Saeed, Muhammad Humayon Khan Vardag, Muhammad Farhat Ullah & Nadeem Iqbal, "Roman Urdu Fake Reviews Detection Using Stacked LSTM Architecture", *Springer*, September 2022.

[38]ChunyongYin, Haoqi Cuan, Yuhang Zhu,Zhichao Yin, "Improved Fake Reviews Detection Model Based on Vertical Ensemble Tri-Training and Active Learning", *ACM Transactions on Intelligent Systems and Technology,* Vol. 12, No. 3,2021.

[39] Zhang, Y., Wang, X., Liu, Y., & Zhang, Y., "Fake review detection using machine learning techniques: A review," *Journal of Ambient Intelligence and Humanized Computing*, 9(2), pp. 383-395, 2018.

[40] He, Y., Lee, W. C., & Chan, J. W., "Fake review detection using FastText with convolutional neural network," *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC)*, pp. 1486-1492, 2019.

[41] Hajek, P., Barushka, A. and Munk, M., 2020. Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. Neural Computing and Applications, 32 (23) , pp.17259-17274.

[42]Shahariar, G. M., et al. "Spam review detection using deep learning." 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE, 2019.

[43] Chiong, Raymond, et al. "A textual-based featuring approach for depression detection using machine learning classifiers and social media texts." *Computers in Biology and Medicine* 135 (2021): 104499.

[44] Wang, Chih-Chien; Day, Min-Yuh; Chen, Chien-Chang; Liou, Jia-Wei., "Detecting spamming reviews using long short-term memory recurrent neural network framework", *Proceedings of the 2nd International Conference on E-commerce, E-Business and E-Government - ICEEG ' 18* , pp. 16– 20, 2018. doi:10.1145/3234781.3234794

[45]Wang, Y.; Zhang, J., "Keyword extraction from online product reviews based on bi-directional LSTM recurrent neural network," *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* , pp. 2241 – 2245 , 2017 . doi:10.1109/IEEM.2017.8290290

[46] Sherry Girgis, Eslam Amer, Mahmoud Gadallah, "Deep Learning Algorithms for Detecting Fake News in Online Text", *IEEE*, 2018.

[47] Lin, Yuming; Zhu, Tao; Wu, Hao; Zhang, Jingwei; Wang, Xiaoling; Zhou, Aoying., "Towards online anti-opinion spam: Spotting fake reviews from the review sequence," 2014 *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*,pp.261–264, 2014.

[48] Mikolov, T., Chen, K., Corrado, G., & Dean, J., "Efficient estimation of word representations in vector space," *Computation and Language*, pp. 1-12, 2013. arXiv preprint arXiv:1301.3781.

[49] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T., "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135-146, 2017.

[50] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K.., "Bert: Pre-training of deep bidirectional transformers for language understanding, *Computation and Language,* 2018. arXiv preprint arXiv:1810.04805.

[51] Pennington, J., Socher, R., & Manning, C. D., “Glove: Global vectors for word representation,” *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532-1543, 2014.

[52] Chaudhari, N., & Sohoni, M., “Hierarchical GloVe for word embeddings,” *Information Processing & Management*, vol. 58, no. 1, 2021. 102364.

[53] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T., “Bag of tricks for efficient text classification,” *Computation and Language*, 2016. arXiv preprint arXiv:1607.01759.

[54] Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T., “Learning word vectors for 157 languages,” *In Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2018.

[55] Arefyev, N., & Wolf, L., “Speeding up fastText training using hierarchical softmax,” *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7042-7047), 2020.

[56] Vashishth, S., Khare, R., Bhattacharyya, P., & Talukdar, P., “Reside: Improving Distantly-Supervised Neural Relation Extraction using Side Information,” *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1257-1266, 2018. arXiv preprint arXiv:1810.02840.

[57] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., & Wu, Y., “Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions,” *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 2237-2246, 2018.

[58] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K., “BERT: Pre-training of deep bidirectional transformers for language understanding,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171-4186, 2018.

[59] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., & Stoyanov, V., “RoBERTa: A robustly optimized BERT pretraining approach, *Computation and Language*, 2019. arXiv preprint arXiv:1907.11692.

[60] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R., “ALBERT: A lite BERT for self-supervised learning of language representations,” *Computation and Language* , 2020. arXiv preprint arXiv:1909.11942.

[61] Beltagy, I., Lo, K., & Cohan, A., “SciBERT: A pretrained language model for scientific text,” *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. 3606-3611, 2019.

[62] Chalkidis, I., Fergadiotis, M., & Androutsopoulos, I., “Legal-BERT: The Muppets straight out of law school,” *Computation and Language*, 2020. arXiv preprint arXiv:2010.02558.

[63] Sanh, V., Debut, L., Chaumond, J., & Wolf, T., “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” *Computation and Language*, 2020. arXiv preprint arXiv:1910.01108.

[64] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L., “Deep contextualized word representations,” 2018. arXiv preprint arXiv:1802.05365.

[65] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I., “Improving language understanding by generative pre-training,” 2018. URL https://s3-us-west-2. amazonaws. com/ openai- assets/ researchcovers/ languageunsupervised/language understanding paper. pdf.

[66] Li, Z., Wang, B., & Ma, W. Y., “Coupled LSTM: Learning long-term dependencies with coupled memory cells,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3259-3272, 2021.

[67] Liu, X., Huang, L., Li, X., & Zhou, M., “GLSTM: A novel gated linear unit based LSTM for natural language processing tasks,” *Knowledge-Based Systems*,” vol. 236, 107230, 2022.

[68] Jang, Beakcheol; Kim, Myeonghwi; Harerimana, Gaspard; Kang, Sang-ug; Kim, Jong Wook., “Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism,” *Applied Sciences*, vol. 10, no. 17, 1-14, 2020. doi:10.3390/app10175841

[69] Wang, C., Ding, X., Zhou, X., Wang, C., & Liu, T., “Depthwise bidirectional LSTM for speech emotion recognition,” *IEEE Signal Processing Letters*, vol. 28, pp. 114-118, 2021.

[70] Zhang, X., Lu, Y., Zheng, Y., & Wang, B., “FusionLSTM: A convolutional and recurrent neural network architecture for natural language processing tasks,” *Information Processing & Management*, vol. 59, no. 1, 102606, 2022.

[71] Salminen, J., Kandpal, C., Kamel, A. M., Jung, S., & Jansen, B. J. (2022). Creating and detecting fake reviews of online products. Journal of Retailing and Consumer Services, 64, 102771. https://doi.org/10.1016/ j.jretconser.2021.102771

[72]Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf, Muhammad Ovais Ahmad, "Fake News Detection Using Machine Learning Ensemble Methods", *Complexity*, vol. 2020, Article ID 8885861, 11 pages, 2020. https://doi.org/10.1155/2020/8885861