



# Fake News Detection Using BERT and Bi-LSTM with Grid Search Hyperparameter Optimization

*Araştırma Makalesi/Research Article*

 Muhammet TAN<sup>1</sup>,  Halit BAKIR<sup>2</sup>

<sup>1</sup>Institute of Graduate Studies, Sivas University of Science and Technology, Sivas, Turkey

<sup>2</sup>Department of Computer Engineering, Sivas University of Science and Technology, Sivas, Turkey

[24011002@sivas.edu.tr](mailto:24011002@sivas.edu.tr), [halit.bakir@sivas.edu.tr](mailto:halit.bakir@sivas.edu.tr)

(Geliş/Received:24.07.2024; Kabul/Accepted:21.10.2024)

DOI: 10.17671/gazibtd.1521520

**Abstract**—Fake news and misinformation disseminated on social media can significantly distort public perception and behavior, leading to serious issues. These deceptive contents have the potential to increase societal polarization by causing individuals to make decisions based on false information. During crises, the spread of fake news can endanger public health, destabilize the economy, and undermine trust in democratic institutions. To address this critical issue, numerous studies today employ machine learning and deep learning models. In this study, the transformer architecture, widely used in natural language processing, was utilized. To process longer texts more reliably, Bidirectional LSTMs were hybridized with the transformer architecture in the model. For easier detection of fake tweets, the target categories in the dataset were balanced, and the TomekLinks algorithm was employed to enhance classification performance. To improve model performance, a parameter pool was established, and Grid Search was used to identify parameters yielding the most successful results. In our tests, all top 10 models achieved an accuracy of 99%. The highest-performing model achieved an impressive accuracy of 99.908%.

**Keywords**— fake news detection, natural language processing, bert, long-short term memory

## Transformer Modellerinden Bert ve İki Yönlü LSTM'lerin Hibrit Kullanılması ve Grid Search Hiperparametre Optimizasyonu ile Sahte Haber Tespiti

**Özet**— Sosyal medyada yayılan sahte haberler ve yanlış bilgiler, toplum algısını ve davranışlarını önemli ölçüde çarpıtabilir ve ciddi sorunlara yol açabilir. Bu yanıltıcı içerikler, bireylerin yanlış bilgilere dayanarak kararlar almasına neden olarak toplumsal kutuplaşmayı artırma potansiyeline sahiptir. Kriz zamanlarında, sahte haberlerin yayılması halk sağlığını tehlikeye atabilir, ekonomiyi istikrarsızlaştırabilir ve demokratik kurumlara olan güveni zedeleyebilir. Bu önemli sorunu ele almak amacıyla, günümüzde birçok çalışma makine öğrenimi ve derin öğrenme modellerini kullanmaktadır. Bu çalışmada, doğal dil işleme alanında yaygın olarak kullanılan transformer mimarisi tercih edilmiştir. Uzun metinlerin daha istikrarlı bir şekilde işlenmesi için modelde Bidirectional LSTM'ler (İki Yönlü Uzun-Kısa Vadeli Bellek) transformer mimarisine hibrit hale getirilmiştir. Sahte tweetlerin daha kolay tespit edilebilmesi amacıyla, veri setindeki hedef kategoriler dengelenmiş ve sınıflama başarımının artırılması için TomekLinks kütüphanesi kullanılmıştır. Model performansını artırmak için bir parametre havuzu oluşturulmuş ve Grid Search metodu ile en başarılı sonuçları veren parametreler belirlenmiştir. Yapılan testlerde, en iyi 10 modelin tamamı %99 doğruluk oranına ulaşmıştır. En yüksek performans gösteren model, %99.908 doğruluk oranı elde etmiştir.

**Anahtar Kelimeler**— sahte haber tespiti, doğal dil işleme, bert, uzun-kısa süreli bellek

## 1. INTRODUCTION

In today's digital age, the rapid dissemination of information through social media platforms has revolutionized how we interact with news and data. However, this newfound connectivity has also given rise to a concerning phenomenon: information pollution. From politics to education and even sports, the inundation of false or exaggerated information permeates our online spaces, blurring the lines between fact and fiction. Recent events such as the COVID-19 pandemic and the U.S. presidential elections vividly illustrate the severity of this issue. Platforms like X (formerly Twitter) have become breeding grounds for the spread of misinformation, where false narratives can quickly gain traction and influence public opinion. According to a Gartner analysis, by 2022, the majority of individuals in developed economies may consume more false knowledge than genuine information, highlighting the urgent need to address this growing problem [1]. The paper "Fake News and Social Media" by [2] details the profound effects of disinformation campaigns on society, highlighting the critical need for vigilance and effective countermeasures. Recognizing the critical importance of reliable information, researchers and developers have turned their focus to the development of machine learning and deep learning algorithms. These technologies aim to discern the veracity of information circulating on social media platforms, offering a potential solution to combat misinformation. Today, we see that transformer-based algorithms are frequently used to deal with these problems. The advantage of these models is that even if there is very little data in the dataset, higher success can be achieved compared to classical machine learning methods because they use a pre-trained data with transfer learning methods. In addition to the transformer architecture, CNN (Convolutional Neural Network), LSTM (Long Short-Term Memory), BI-LSTM (Bidirectional Long Short-Term Memory) and hybrid models are frequently used in fake news detection and filtering. In addition to the models and algorithms used, the main problem encountered in fake news detection is the difficulty in finding satisfactory data. Researchers and authors working on the subject have been closely interested in this problem and have carried out many pre-processing stages like IDF (Inverse Document Frequency), TF-IDF (Term Frequency-Inverse Document Frequency), BOW (Bag of Words), n-grams to provide better meaning connections on the data in order to get better results from the data they find. When we look at the studies, it is seen that the validation values of the algorithms working

with trained models in the step after the pre-processing stage are higher.

This article delves into the pervasive issue of information pollution, examining its implications across various sectors and underscoring the imperative for reliable information in today's digital landscape. Through the exploration of cutting-edge technologies and research endeavours, we aim to shed light on the ongoing efforts to safeguard the integrity of information in the age of social media.

### *Novelty*

In our study, we adopted a hybrid approach that integrates the Transformer model, a pivotal component in natural language processing. To address the learning deficiencies and forgetting issues frequently encountered in Recurrent Neural Network (RNN) methods, particularly when dealing with lengthy text sequences, we employed Long Short-Term Memory (LSTM) networks. We implemented a Bidirectional LSTM architecture that processes information in both forward and backward directions, thus facilitating deeper and more efficient learning. Furthermore, we applied the Tomek Links algorithm to mitigate classification errors and tackle data imbalance, along with implementing effective text preprocessing techniques to enhance the performance of our hybrid model. Although numerous studies in the domain of fake news detection have utilized various datasets and pre-trained models, our research distinguishes itself through the utilization of a well-annotated dataset comprising over 130,000 records. We performed hyperparameter tuning using the Grid Search method from the Optuna library, which significantly improved the model's performance. Notably, to our knowledge, there is no existing study that simultaneously incorporates all these methodologies—leveraging a large, well-annotated dataset, applying Tomek Links to address class imbalance, and integrating DistilBERT with Bidirectional LSTM while systematically optimizing hyperparameters across various machine learning models. This comprehensive approach fills a critical gap in the literature and highlights the novelty of our proposed method. As a result, we developed a robust model capable of effectively classifying fake and real tweets on social media.

## 2. RELATED WORKS

The detection of fake news on social media platforms has been a prominent research focus, particularly with the rise of misinformation during global events such as the COVID-19 pandemic. Various methodologies and datasets have been developed to tackle this issue, leveraging machine learning and deep learning techniques. In the following sections, we explore significant contributions to the field, highlighting datasets and models that have advanced the detection capabilities for fake news, along with their respective performance metrics and application scenarios. For example, [3] introduced the TruthSeeker dataset for detecting fake news on social platforms, particularly Twitter. This dataset includes over 180,000 labelled tweets from 2009 to 2022, collected via Amazon Mechanical Turk with rigorous verification by multiple Turkers and institution employees. To analyse user behaviour and content impact, three auxiliary social media scores (Bot, credibility, and influence) were added. Various machine learning models, such as BERT, RoBERTa, DistilBERT, BERTweet, and ALBERT, were used to evaluate the dataset's effectiveness. Offering both binary and multi-class classifications, the TruthSeeker dataset shows promise for enhancing fake news detection on social media platforms. [4] utilize the XGBoost algorithm to classify tweet text, applying natural language processing techniques for preprocessing. Authors employ a hybrid CNN-RNN and BERT transformer for detection, analysing originator credibility and writing styles. Using the FakeNewsNet dataset, authors emphasize data cleaning due to Twitter's informality. XGBoost, which reduces overfitting, adjusts data point weights to correct misclassifications. While CNN-RNN and BERT are both used for tweet classification, BERT significantly outperforms CNN-RNN with 98% accuracy compared to XGBoost's 81%. [5] propose a hybrid approach for detecting fake news in COVID-19 datasets, combining BERT, SVM, and the NSGA-II algorithm. BERT extracts contextual meaning, SVM detects fake news patterns, and NSGA-II optimizes word embedding. This model aims to improve accuracy by 5.2% by reducing sentence ambiguity. The combination of BERT's contextual understanding, SVM's classification, and NSGA-II's optimization outperforms other models in predicting fake news in COVID-19 datasets. [6] highlight limitations in existing fake news detection methods and introduce FakeBERT, a novel BERT-based deep learning approach. FakeBERT uses bidirectional training to better capture semantic and long-distance dependencies in

sentences. The architecture combines BERT with three parallel 1D-CNN blocks of varying kernel sizes and filters, followed by max-pooling and densely connected layers. This setup effectively handles large-scale text and addresses natural language ambiguity. FakeBERT achieves 98.90% accuracy, outperforming existing benchmarks by 4%, and shows promise for fake news detection. [7] examined the effectiveness of various machine learning techniques in detecting COVID-19 misinformation, using Decision Trees, Naïve Bayes, Logistic Regression, and Support Vector Machines within the KNIME Analytics Platform. Their model differentiates between accurate information and false claims, addressing a class imbalance where 63% of the articles are fake and 37% are accurate. Experimental results show that Naïve Bayes outperforms other methods in accuracy, precision, recall, and F1 score. [8] developed an advanced ensemble learning-based system for fake news detection using datasets like LIAR, POLITIFACT, ISOT, and COVID-19. Their model operates in three stages: first, extracting and preprocessing features from news content using NLP techniques and n-gram TF-IDF representation; second, training multiple binary classifiers with deep learning architectures to identify latent features; and third, creating a multi-class classifier with a multi-layer perceptron (MLP) trained on features from the binary classifiers' outputs. Their model outperforms existing state-of-the-art systems in fake news detection. [9] propose an NLP-based fake news detection method using deep learning and CNN. Their system aims to detect fake news across various domains, including education, news, and politics. The model achieves up to 99% training accuracy and 97% test accuracy, with detailed descriptions of the system design and experimental methodology provided. However, they note a lack of data to further enhance the model's robustness. [10] utilize transfer learning to detect fake news in English and Spanish. Transfer learning enhances the target model's performance by using knowledge from a pre-trained model on a source dataset. Authors develop separate models for each language, involving two phases: Training the Language Model (LM) and the Target Model. Using 300 XML files per language, authors introduce the ULMFiT model for profiling fake tweet spreaders. Initially trained on general English/Spanish data from Wikipedia, the LM knowledge is transferred to the fake news detection task. Their model achieves 64% accuracy for Spanish and 62% for English. This LM can also be applied to other English/Spanish NLP tasks. To address fake news detection, [11] worked with datasets like

BuzzFeedNews, LIAR16, BS Detector, and CREDBANK19. Authors noted that no single dataset covers all relevant features, and each has limitations. Authors also performed operations like clickbait, spammer, and bot detection to validate dataset robustness. [12] present a model with three main phases: input, pre-processing, and output. Authors collect labelled and unlabelled news data in the input phase and preprocess it using NLP. The preprocessing phase involves vectorization, a Recommender System, and multi-class classification. Authors propose a novel multi-class semi-supervised approach for self-training, utilizing a combination of classified and predominantly unlabelled data. Their method incorporates a similarity algorithm to enhance self-training by assigning new labels to labelled data. Evaluation on two benchmark datasets using logistic regression, decision tree, naive Bayes, and linear SVM shows their method's effectiveness and robustness in multi-class fake news classification, contributing to more reliable predictive models. [13] studied fake news detection during the COVID-19 pandemic using Decision Tree, K-Nearest Neighbour, Logistic Regression, Support Vector Machine, and Random Forest algorithms on a new dataset. Random Forest consistently outperformed other algorithms, closely followed by Support Vector Machine, across all configurations. Although textual and linguistic features individually enhanced detection, combining them did not significantly improve results. Bigrams and part-of-speech tags showed varying effectiveness. The research suggests that traditional machine learning methods can effectively utilize textual and linguistic features for fake news detection, with Random Forest and SVM achieving over 95% accuracy and F1-scores. Their research contributes by analysing emotional aspects of fake news through two main steps: (RQ1) identifying fake news and (RQ2) identifying and characterizing emotions. For RQ1, authors evaluated various algorithms for detecting fake news. After an extensive review of literature, data collection from tweets, sampling, and applying machine learning and deep learning algorithms, dense neural networks (accuracy: 0.956), random forests (accuracy: 0.949), and LSTM networks (accuracy: 0.931) showed the highest average accuracy. Transformer-based models like BERT and DistilBERT also performed well in their evaluation. [14] initially explored machine learning experiments and speculated that deep learning algorithms might yield better results for fake news detection. Various word-embedding techniques such as Word2Vec, GloVe, and FastText were utilized to generate effective data representations. For classification, deep learning models including LSTM, BiLSTM,

CNN-LSTM, and CNN-Bi-LSTM were employed. Due to the absence of a single large, standard dataset for fake news detection, the study integrated two publicly available datasets – Fake and real news, and all data, resulting in a dataset comprising 64,934 labelled news articles. Among the techniques tested, Word2Vec word embedding combined with the CNN-BiLSTM model demonstrated the highest performance, achieving accuracy, precision, recall, F1 measure, and AUC-ROC values of 0.975, 0.984, 0.970, 0.977, and 0.992, respectively. [15] tackled the fake news problem by applying the XGBoost model to their dataset. Authors also implemented SVM (Support Vector Machines), RF (Random Forest), LR (Logistic Regression), CART (Classification and Regression Trees), and NNET (Neural Network) machine learning models to enhance their algorithm's robustness. To generalize these models, authors conducted cross-validation. According to their results, the RF model achieved the highest accuracy at around 94%, while NNET showed the lowest performance with approximately 92.1%. In [16], the authors investigate the application of DistilBERT, a condensed version of BERT, for detecting XSS attacks in web applications. Leveraging DistilBERT's strong NLP capabilities, authors extract semantic features from input data to identify malicious XSS payloads. Their approach is evaluated on a comprehensive dataset, achieving high accuracy (99.82%), precision (99.83%), recall (99.66%), and F1 score (99.75%). Visualizations including confusion matrices, ROC curves, and precision-recall curves illustrate the model's robust performance. This research underscores the effectiveness of transformer-based models in fortifying web application security against advanced cyber threats. In [17], the authors introduce a RoBERTa-based bi-directional Recurrent Neural Network model for spam detection on social networks. Using RoBERTa to learn contextualized word representations, authors enhance the performance of the stacked BiLSTM network. A comparative study with common transformer-based models shows that their RoBERTa-BiLSTM model outperforms others on three benchmark datasets, achieving accuracies of 98.15% on Twitter, 94.41% on YouTube, and 99.74% on SMS data. In [18], the authors propose a CBLSTM (Contextualized Bi-directional Long Short Term Memory neural network) model to address spam detection on social networks. This model leverages deep contextualized word representation to overcome the limitations of traditional word embedding models, such as the “out of vocabulary” problem and lack of context. Experimental results on three benchmark datasets demonstrate that

their proposed method achieves high accuracy and outperforms existing state-of-the-art methods in detecting spam on social networks. In [19], the authors introduce ALBERT4Spam, a deep learning methodology for identifying spam on social networking platforms. This model leverages the ALBERT model for contextualized word representations and is built upon the Bidirectional Long Short-Term Memory neural network (BLSTM). Using random search to fine-tune hyperparameters, their model achieves optimal performance. Experiments on three benchmark datasets show that ALBERT4Spam outperforms widely used methods in spam detection, with precision results of 0.98 for Twitter, 0.96 for YouTube, and 0.98 for SMS datasets. In [20] the authors conducted an efficient analysis utilizing transformer-based BERT models, CNN, and BiLSTM architectures. Authors tested five different models, including variants of BERT such as BERT, DistilBERT, and BERTurk, as well as CNN architectures, across eight different datasets including LIAR, ISOT, GossipCop, and BuzzFeedNews. Through a comparative analysis, authors evaluated and reported the performance of the models across these diverse datasets. In [21] authors investigated the transformation of news dissemination in the context of social media, highlighting the shift from traditional media platforms to user-generated content. They defined fake news as information produced by deceptive or sensationalist users aimed at manipulation or provocation. The study emphasized the rapid spread of fake news among ordinary social media users, underscoring the critical need for swift detection mechanisms. Recognizing the limitations of expert systems, which struggle to keep pace with the high volume of social media traffic, the authors advocated for the development of semi-automatic and automatic fake news detection systems. By collecting and annotating data from Twitter, they implemented various supervised (K-Nearest Neighbor, Support Vector Machines, and Random Forest) and unsupervised (K-means, Non-Negative Matrix Factorization, and Linear Discriminant Analysis) machine learning algorithms. The results demonstrated that supervised learning approaches achieved the highest performance, with an average F1-score of 0.86, while unsupervised methods yielded a lower F1-score of 72%. The authors of [22] investigated the challenges associated with the spread of fake news in the digital age, examining its adverse effects on public perception and trust. In their study, authors developed a supervised machine learning algorithm designed to classify social media data as fake news. The methodology included five main components: data acquisition from Twitter,

data preprocessing, data transformation, model development utilizing Naive Bayes, decision tree, and support vector machine (SVM), and model evaluation through accuracy, precision, recall, and F1-score metrics. The results indicated that the decision tree algorithm achieved the highest accuracy for textual data and metadata, while also performing well in terms of precision, recall, and F1-score for the classification tasks. Additionally, SVM exhibited strong precision and recall metrics in the metadata classification.

## *Background*

### *2.1. Artificial Intelligence (AI)*

Artificial intelligence (AI) today is changing many fields of technology [23], [24], [25], [26]. This affects differently fields such as healthcare, finance, transportation, and communications. In the medical field, AI helps with diagnoses, treatment plans and research for new drugs. This improves patient care. In finance, AI helps with transactions, risk assessment and fraud detection. This improves decision making and the market works well. AI also supports traffic by creating self-driving cars and creating roads More safety and traffic management. In communication, AI helps talk with machines and translate language and emotional understanding. Overall, AI is important in creating new technologies and the idea was born.

#### *2.1.1. Machine Learning*

Machine learning, an important subfield of artificial intelligence, encompasses a diverse set of algorithms and methods that enable computer systems to learn from data and make predictions or decisions without being explicitly programmed. At their core, machine learning algorithms leverage statistical techniques to identify patterns and relationships in data, thereby deriving insights and facilitating autonomous decision-making. These algorithms are often classified into supervised learning, unsupervised learning, and reinforcement learning models, each suitable for different learning situations. Supervised learning involves training algorithms on labelled data sets, where input-output pairs are provided, allowing the algorithm to learn the mapping between the input and the corresponding output. In contrast, unsupervised learning tasks involve extracting patterns and structures from unlabelled data, facilitating tasks such as clustering and anomaly detection. On the other hand, reinforcement learning focuses on training agents to interact with the environment

with the goal of maximizing cumulative rewards, often used in dynamic decision-making situations. Through these models, machine learning continues to drive innovation in fields ranging from healthcare and finance to natural language processing and computer vision, paving the way for groundbreaking advances. Transformative computing in data, prediction, and decision support systems.

### 2.1.2. Deep Learning

Deep learning, a subset of machine learning, includes a class of algorithms inspired by the structure and function of neural networks in the human brain. These algorithms are characterized by using multiple interconnected layers of artificial neurons to extract high-level features from raw data. Deep learning models excel at automatically learning complex patterns and representations from large volumes of unlabelled data, enabling tasks such as image and speech recognition, language processing natural and automatic decision making. The success of deep learning can be attributed to its ability to exploit hierarchical representations of data, extracting and incrementally refining features from each layer of the network. Using techniques such as backpropagation and stochastic gradient descent, deep learning models are trained to minimize errors and optimize performance on specific tasks. Deep learning has revolutionized many different sectors, from healthcare and finance to transportation and entertainment, driving innovation and breakthroughs in artificial intelligence research and applications.

### 2.1.3. Long Short-Term Memory (LSTM) and Bidirectional LSTM

Recurrent Neural Networks (RNNs) are a class of artificial neural networks commonly used for processing sequential data. However, they often encounter challenges, particularly the 'forgetting problem,' when dealing with long sequences. To address this, Long Short-Term Memory (LSTM) networks were developed. LSTMs are specifically designed to overcome the limitations of standard RNNs, especially their susceptibility to long-term dependency issues. Unlike traditional RNNs, which may struggle to retain information across extended sequences, LSTMs utilize memory cells that enable more effective handling of long-term dependencies. Each LSTM network consists of a chain of recurrent network modules, which are more complex than the single-layer structures found in

standard RNNs (e.g., a single tanh layer). LSTMs are explicitly designed to mitigate the exploding and vanishing gradient problems, making them well-suited for capturing longer-term dependencies in sequence data.

### 2.2. Text Processing and Feature Extraction Methods

#### 2.3. Feature Extraction Methods

Text data vectorization involves converting text into interactive vectors, enabling machines to solve math problems and process language. Researchers have developed various models for this purpose:

TF-IDF: This common method assigns importance to terms in documents, enhancing search engine performance. However, its adaptability is limited due to the selectiveness of the IDF term. In a more formal mathematical context, the computation of the TF-IDF score for the term  $t$  within the document  $d$  from the document set  $D$  is articulated as in question 1.

$$TF-IDF(t,d,D)=TF(t,d)\times IDF(t,D) \quad (1)$$

Word2Vec: This model generates semantic representations for words, aiming to capture their senses and relationships.

SentenceToVec: Extending Word2Vec, this approach averages word vectors to represent sentences. Notable advancements include Skip-Thought Vectors.

Doc2Vec: Extending Word2Vec to handle entire documents, Doc2Vec uses a similar process as SentenceToVec.

#### 2.3.1. Text Processing Methods

##### Text Tokenization

The BERT (Bidirectional Encoder Representations from Transformers) token engine is a basic one component of natural language processing (NLP) systems, known for their ability to capture contextual information and semantic nuances in text strings. Developed by Google AI in 2018, BERT token uses a complex tokenization strategy to split input text into sequence of sub-word tokens, allowing the model to consider contextual relationships between Speech is two-way. Unlike traditional tokenization methods that represent words in isolation, BERT tokenizer considers the entire context of the sentence, capturing dependencies and semantics links between words. This contextual understanding helps improve performance of downstream NLP tasks, such as text classification, named entity recognition, and sentiment analysis. By leveraging the BERT token, NLP practitioners can harness the power of Contextual integration to uncover deeper insights from text data, paving the way for more

powerful solutions and the system understands language accurately as seen in Tables 1 and Table 2.

Table 1. Overview of Transformer Based Text Tokenization

Special Tokens	[PAD]	[UNK]	[CLS]	[SEP]	[MASK]
Special Token ID	0	100	101	102	103

Table 2. Sample Text Tokenization

Sample Tweet	“Hello how are you?”
Tokenized version of the Tweet	[101, 7592, 2129, 2128, 2017, 102]

Positional embedding

One fundamental challenge in processing sequential data like text is capturing positional information. Transformer addresses this through positional embedding, where each token in the input sequence is augmented with positional information as seen in Figure 1. This allows the model to discern the order of tokens, crucial for understanding the context of the input.

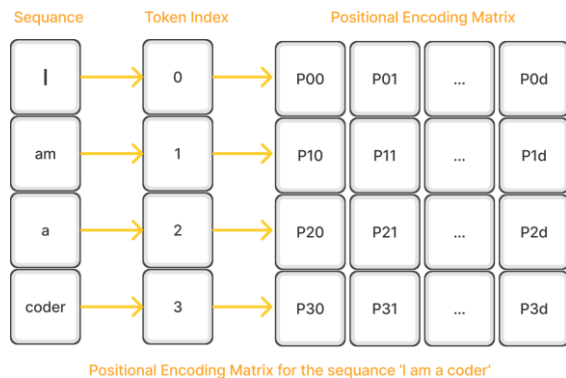


Figure 1. Visualization of Positional embedding

Padding

To accommodate variable-length inputs within a fixed-size matrix, padding is employed. This involves adding placeholder tokens, typically zeros, to shorter sequences to match the length of the longest sequence in the batch as seen in Table 3. Padding ensures uniformity in input dimensions, facilitating efficient batch processing.

Table 3. Overview Of Padding

Unpadded Input	Padded Input
[ [1,2,3], [4,5], [6,7,8,9,10] ]	[ [1,2,3,0,0], [4,5,0,0,0], [6,7,8,9,10], ]

2.4. Transformer Architecture

2.5. Overview of Transformer Architecture

Since the aim of this study is to evaluate the sentences in the tweets in terms of emotion and semantics, to make a reality prediction by taking advantage of their importance in the sentence on a word basis, we benefited from the transformer architecture as seen in Figure 2, which is frequently and successfully used in the field of natural language processing today. The Transformer architecture has emerged as a pivotal advancement in deep learning, particularly within the realm of Natural Language Processing (NLP). Developed on the foundation of attention mechanisms, it represents a paradigm shift in sequence modelling, enabling more effective handling of sequential data such as text. In this article, we delve into the key components of the Transformer architecture and explore some of the most prominent Transformer-based models shaping the landscape of NLP today.

The transformative impact of the Transformer architecture cannot be overstated. Its inception marks a watershed moment in the field of NLP, revolutionizing the way we process and understand language. At its core, the Transformer architecture harnesses the power of attention mechanisms, allowing models to focus on relevant parts of the input sequence with unprecedented precision. This not only enhances the model's ability to capture intricate linguistic patterns but also significantly improves its performance across various NLP tasks.

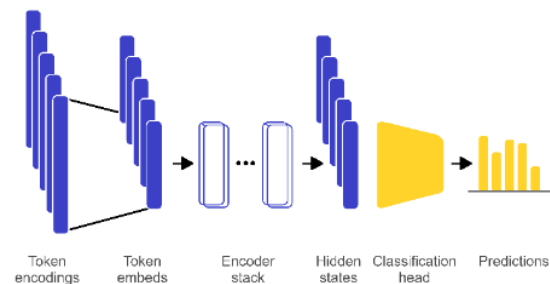


Figure 2. The architecture used for sequence classification with an encoder-based transformer.

One of the defining features of the Transformer architecture is its inherent scalability. Unlike traditional recurrent neural networks (RNNs) or convolutional neural networks (CNNs), Transformers exhibit superior parallelizability, making them well-suited for processing large volumes of text data efficiently. This scalability has played a crucial role in democratizing

NLP, enabling researchers and practitioners to tackle increasingly complex language processing tasks with ease.

### Attention Mechanism

The attention mechanism in deep learning was created to enhance machine translation by focusing on key parts of the input, like zooming in on one conversation in a noisy room. It copies how our brain highlights important sounds and ignores distractions, helping neural networks focus on different parts of the input. This is vital in areas like natural language processing (NLP), where attention helps match parts of a sentence during translation or answering questions. Attention also improves tasks in computer vision, such as pinpointing house numbers in Google Streetview. This guide explores the types, uses, and setup of attention mechanisms in TensorFlow to improve model performance by focusing on important details.

$$\text{Attention}(q,k,v) = \sum \text{similarity}(q,k_i) * v_i \quad (2)$$

- The attention mechanism assesses the likeness between the query  $q$  and every key-value pairs as seen in Figure 3.
- This similarity generates a weight for each key value.
- Ultimately, it generates an output that is the weighted amalgamation of all the values in our dataset.

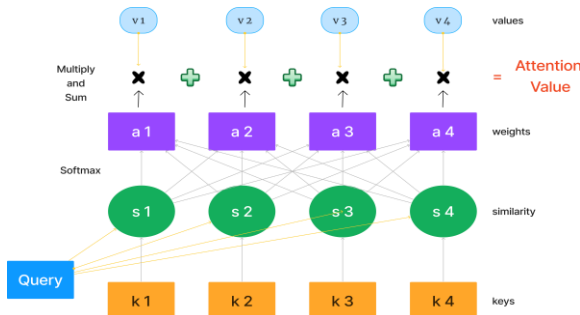


Figure 3. The Simple Overview of Attention Mechanism

#### 2.5.1. Masked language model

A core innovation introduced by models like BERT (Bidirectional Encoder Representations from Transformers) is the masked language model objective. Here, a certain percentage of tokens in the input sequence are masked, and the model is trained to predict these masked tokens based on the surrounding context as seen in Figure 4. This fosters a deeper understanding of inter-token relationships and enhances the model's ability to capture nuanced linguistic structures.

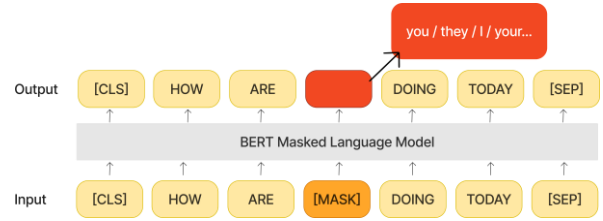


Figure 4. The Simple Example of Masked Language Model

Modern transformer-based models such as BERT, GPT and T5 have revolutionized Nature Language Processing (NLP) by excelling at tasks ranging from understanding language in context to text creation and multitasking learning. The Transformer architecture has revolutionized NLP, allowing models to solve various linguistic tasks with unprecedented accuracy and efficiency. From BERT's contextual language understanding to GPT and T5's language generation capabilities multitasking capabilities, Transformer-based models continue to push the boundaries of what's what feasible in understanding and producing natural language. As research advances in this area, we can anticipate other innovations and applications that harness the transformative power of Transformer-based architecture.

#### 2.6. Data Collection and Preprocessing

#### 2.7. Dataset

#### 2.8. Truth Seeker Dataset

For this study, we employed the Truth Seeker which was published by [3], a comprehensive collection of samples specifically curated to support the development and evaluation of deep learning and machine learning models in fake news detection. The examples in this dataset were labelled by real people from the well-equipped Amazon Mechanical Turk service, which worked meticulously to label each tweet in the dataset as true or false. The target category distribution in this dataset was 68930 for fake tweets and 65268 for real tweets as seen in Figure 5. As can be seen from the numbers, we were able to make a successful classification thanks to the data labelled in a balanced way. Of course, when we carefully examine the content of the texts shared on social media, especially the content of the tweets, we had to correct the grammatical complexities in the tweets shared by many bot accounts and the hashtags, mentions, usernames or spelling mistakes in the tweets shared by real people, which prevented the proposed model from classifying or at least did not contribute to the classification.



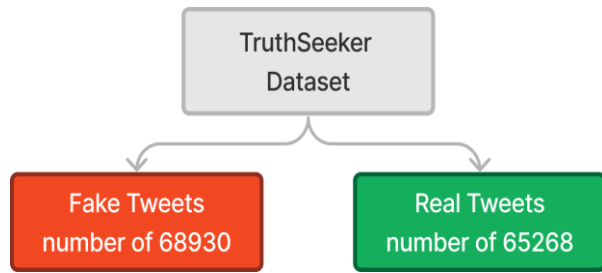


Figure 5. Distribution of the Dataset According to Target Categories

Here are the sample records from the TruthSeeker dataset as seen in Table 4.

We allocated 80% of the records from the TruthSeeker dataset for training and reserved the remaining 20% for validation purposes. After this division, the training set comprised 107,358 samples, while the validation set contained 26,840 samples. Furthermore, as there were no missing values in the relevant columns of the dataset, there was no necessity for data removal or imputation of missing values with averages or other methods.

2.8.1.1. Politifact Dataset

In this study, the publicly available PolitiFact dataset was used to evaluate the model's classification ability and to derive more objective inferences. The dataset, consisting of 19,422 labeled records, underwent data cleaning to remove any empty entries, followed by a text preprocessing phase. Afterward, the dataset was split into training and testing sets with an 80/20 ratio. As a result, 14,511 samples were prepared for training, and 4,837 samples for testing. The prepared data was then used to train a binary classification model, following the same approach as with the TruthSeeker dataset.

Table 4. Example Entries from the Dataset

Tweet	Label
"@AndreaR03428969 People vote with their pockets. Working class Americans (especially Obama-Trump-Biden voters) will remember that extra money from Trump, Bidens reconciliation failure & \$15 minimum wage failure, the ending of child tax benefits and eviction moratorium, and vote for Trump again."	1
@JackRichardso99 @Thee_Roxy_Cox @gnomeicide @glenn_coin @malaconotus @JAGLeMans @Bluesterge2 @lovejoy92 @UKCCovid19Stats This virus mutates, seemingly quite readily. The more transmission, the more likely a vaccine-resistant escape variant	0

will develop. Plus you'll subject the unvaccinated to a small risk of death, and a greater risk of long covid.	
--	--

2.8.2. Preprocessing Phase

2.8.3. Text Preprocessing

For the machine learning and deep learning models we will use in this study to be successful, the data had to be open to study as much as possible. Although the proposed model employs the transformer architecture, known for its success in understanding word relationships and identifying noteworthy words in a sentence, we performed preprocessing on the tweets in our dataset. This preprocessing aimed to reduce ambiguity and eliminate unnecessary learning parameters, thereby preventing longer and less successful training. We can list the cleaning processes performed on the tweets in our dataset as follows:

- Cleaning E-Mail Addresses
- Cleaning URL Addresses
- Cleaning Retweet Tokens
- Cleaning HTML Tags
- Cleaning Mentions Dealing with Abbreviations

Here is the sample implementation of preprocessing step on the TruthSeeker dataset as seen in Table 5.

Table 5. Example Implementation of Preprocessing Phase

Before Preprocessing	@POTUS Biden Blunders - 6 Month Update\n\nInflation, Delta mismanagement, COVID for kids, Abandoning Americans in Afghanistan, Arming the Taliban, S. Border crisis, Breaking job growth, Abuse of power (Many Exec Orders, \$3.5T through Reconciliation, Eviction Moratorium)...what did I miss?
After Preprocessing	biden blunders 6 month update inflation, delta mismanagement, covid kids, abandoning americans afghanistan, arming taliban, s. border crisis, breaking job growth, abuse power (many exec orders, \$3.5t reconciliation, eviction moratorium).what miss?

In deep learning, balanced data is essential for accurate model training, yet datasets often exhibit imbalances across classes, posing challenges. To address this, researchers utilize data balancing methods, although their indiscriminate use may lead to overfitting or loss of information. Meanwhile, TomekLinks removes pairs of instances from different classes that are nearest neighbours, enhancing boundary discernment and generalization as depicted in Figure 6. TomekLinks improves model robustness and efficiency, fostering equitable learning and reliable insights in scientific research. In summary, Tomek Links are crucial for reducing imbalance in datasets by removing instances from the majority class close to those in the minority class.



Figure 6. Simple Visualization of Tomek Links

#### 2.8.4. Hyperparameter Tuning

Hyperparameter tuning is a critical process in machine learning and deep learning, aimed at optimizing model performance by systematically adjusting hyperparameters. Hyperparameters, such as learning rate, batch size, and regularization strength, govern the learning process and are distinct from model parameters learned during training. The efficacy of a machine learning model depends greatly on the selection of appropriate hyperparameters, which can significantly impact its performance, convergence, and generalization ability. Optimization techniques like grid search, random search, and Bayesian optimization are commonly used for this purpose in this work, we utilized Grid Search for hyperparameter optimization due to its methodical and exhaustive characteristics. In contrast to random search or genetic algorithms, which depend on stochastic techniques to navigate the hyperparameter space, Grid Search guarantees that all potential combinations within the defined grid are examined [27]. This thorough approach facilitates a more accurate determination of the optimal hyperparameters, especially in cases where the search space is limited. Although stochastic methods like random search can be more effective in larger search spaces, Grid Search provides a more structured and deterministic strategy, ensuring that no viable solution is missed. The hyperparameters

and their range values used in this experiment are shown in Table 6.

Table 6. Parameters used in Hyperparameter Optimization

Hyper Parameter	Ranges And Values
Activation Functions	relu,tanh, gelu
Kernel Initializers	uniform, lecun_uniform, normal
Optimizers	Adam, SGD, Adadelta, RMSprop, Adagrad, Adamax, Nadam
Learning Rates	1e-5, 1e-6
Dense Layers	32, 256
Bidirectional LSTM Layer Unit	128,256

#### 2.9. Proposed Model

While LSTMs are specifically designed to address the long-term dependency problem inherent in traditional RNNs, they still encounter limitations when processing particularly long sequences. Despite their ability to mitigate vanishing and exploding gradient issues through the use of memory cells, LSTMs can struggle with computational inefficiency and performance degradation as sequence length increases. The sequential nature of LSTMs leads to longer training times and can make them less effective at capturing complex contextual relationships over very long text sequences.

To overcome these limitations, a hybrid approach combining the strengths of Transformer-based models like BERT with LSTM networks is proposed. BERT excels in capturing context by utilizing a self-attention mechanism, which allows it to model long-range dependencies more efficiently than LSTMs alone. The bidirectional nature of both BERT and LSTM ensures that information is processed from both directions in the text, enhancing the model's understanding of context. By leveraging the robust contextual representation of BERT and combining it with the sequential processing power of Bidirectional LSTM, this hybrid model can more effectively handle both long-term dependencies and complex linguistic patterns, leading to superior performance in tasks such as fake news detection.

In this study, the Distilbert model, which is a simplified version of the BERT model, was used to check whether the tweets were real or fake. The reason why we made this choice was that despite the high performance in interpretation speed and performance, it gave little loss in terms of achievement. According to many studies conducted in the field, the Distilbert model Its duration

is 60 percent shorter than the Bert model. This speed difference provides a great advantage in using Distilbert for researchers and developers working with large language models. Considering the model size, DistilBERT has 44 million fewer parameters than the BERT model, making it approximately 40% smaller. Despite its reduced size, performance comparisons have shown that DistilBERT retains 97% of BERT's performance, as demonstrated in Figure 7 and supported by several benchmarks [28]. This reduction in model size offers significant trade-off, providing a reasonable balance between performance and faster inference speed.

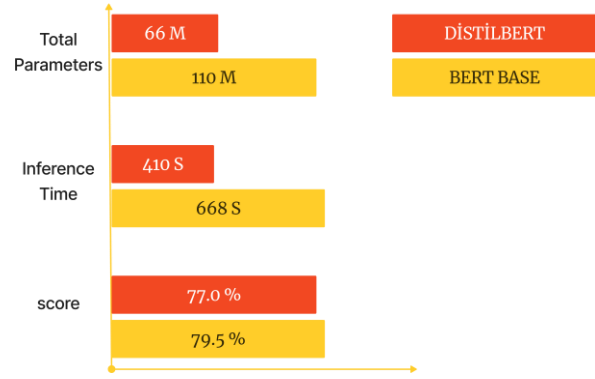


Figure 7. Comparison Of Bert and Distilbert Models

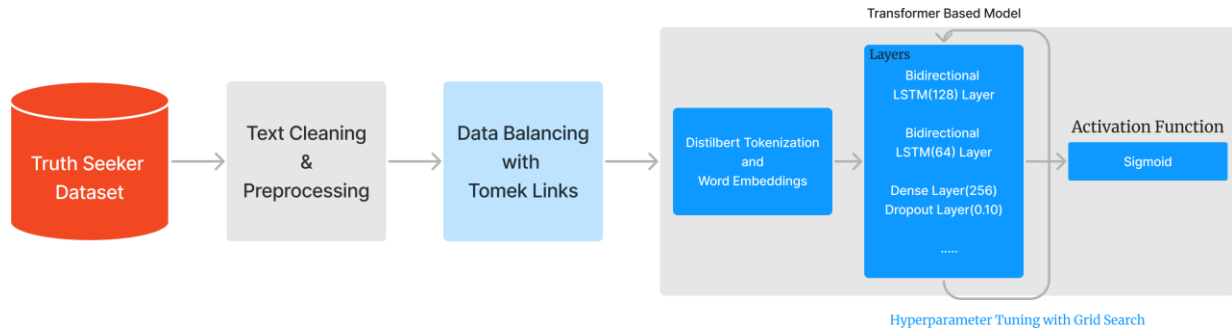


Figure 8. Proposed Model

To summarize our proposed model, as shown in Figure 8, data from the Truth Seeker dataset undergo a series of text preprocessing steps. To minimize classification errors, we utilized the Tomek Links algorithm for under sampling the majority class. This resulted in a more balanced and accurate dataset. Subsequently, the processed data were trained using the transformer based DistilBERT model and our defined list of hyperparameters.

#### 2.10. Experimental Results

##### 2.11. Experiments setup

The experiments in this study were performed on a computer with an i7 12th generation processor. A GTX 3060 video card was used as a GPU accelerator. All experiments were carried out using the TensorFlow library.

##### 2.11.1. Evaluation Metrics

The experiments aim to test how well different computer programs can find fake news. The measures we use to evaluate something include precision, recall, F-score, and accuracy. Precision is the number of right

decisions divided by the total number of decisions in a specific category. It is figured out as:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

True Positive means the right fake news decisions, and False Positive means the wrong fake news decisions. The recall is the number of right decisions made by the machine compared to all the news in a specific category. It is figured out by:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

False Negative is when something that is not fake news is incorrectly labelled as fake news. Accuracy tells us how correct the decisions are compared to the real classification. The machine learning model's decision is only considered correct if it matches the real fake news class in the dataset. It is figured out by adding up some numbers.

$$\frac{TruePositive + TrueNegative}{TrueNegative + FalsePositive + FalseNegative}$$

Where True Negative is the correct not-fake news decision. Finally, the F1-score is the harmonic mean of precision and recall. It is calculated as:

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

### 2.11.2. Obtained Results

#### **Deep Learning**

In this study in order to prevent misclassification we utilized from Tomek Links algorithm. Thanks to this algorithm we obtained very good results on classifying fake and real tweets. In our study we conducted lots of test thanks to Grid Search. It allows us to try and select best hyperparameter which leverages our model's robustness. As we discussed previously, we used a trans-

former-based model to distinguish semantic relationship between words in tweets. In order to get concrete model, we applied several and important text preprocessing methods to eliminate meaningless and redundant words which prevent model to extract and learn important pieces of the sentences. Outputs comes from transformer were fed to LSTM layer. Again, as we discuss, due to RNN algorithm's deficit and forgetting problem in long sequences, we used LSTM layer to overcome the problem. In order to get more fertile result from LSTM, we used Bidirectional LSTM to get more reliable information from the sentences. Outputs that come from Bidirectional LSTM were passed through in Dense layers with different hyperparameters. Evaluated hyperparameters are listed in Table 7. Here are the 10 best models show the best performance with Bidirectional LSTM using output of transformer layer.

Table 7. Experimental Results of 10 Trials

Order	Optimizer	Dropout Layer	Learning Rate	Score
1	Adamax	True	1e-05	0.99908
2	RMSprop	True	1e-05	0.99867
3	RMSprop	True	1e-06	0.99836
4	Nadam	False	1e-06	0.99646
5	Adam	True	1e-05	0.99641
6	Adamax	True	1e-06	0.99621
7	Adamax	False	1e-05	0.99609
8	Adam	True	1e-06	0.99585
9	Nadam	False	1e-05	0.99487
10	RMSprop	False	1e-06	0.99429

The hyperparameter optimization process was conducted using Grid Search, with the results summarized in Table 7. The Adamax optimizer yielded the best performance, achieving an accuracy of 0.99908, highlighting its effectiveness for this specific task. RMSprop and Adam optimizers also demonstrated competitive performance, with accuracy values of 0.99867 and 0.99641, respectively, indicating their suitability for the Bidirectional LSTM-based model. In contrast, models trained with the Nadam optimizer performed slightly lower, with the highest accuracy being 0.99646.

Regarding the dropout layer, models incorporating dropout consistently outperformed those without it across different optimizers, underscoring the importance of regularization in preventing overfitting—particularly in recurrent neural networks like LSTMs. The learning rate also played a critical role in the model's performance. A learning rate of

0.00001 proved most effective for the top-performing models, while a lower rate of 0.000001 led to marginally reduced accuracy, demonstrating the importance of tuning the learning rate for optimal convergence.

Additionally, the 'uniform' kernel initializer was the most optimal choice across the top-performing models. The best-performing model, identified through Grid Search, was trained for 10 epochs, which was sufficient for convergence without overfitting. As seen in Figure 9, since there was no increase in the training curve as training progressed, the training was limited to 10 epochs. The hyperparameter optimization process had a significant impact on model performance, with fine-tuning of parameters resulting in near-perfect accuracy. The best model from the Grid Search was trained for 10 epochs, and the results are shown in Figures 9.

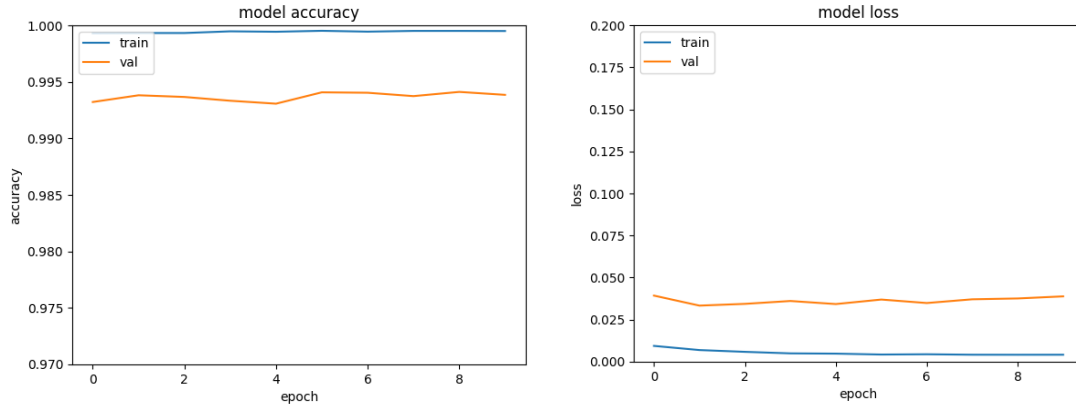


Figure 9. Model Accuracy and Loss of Best Model

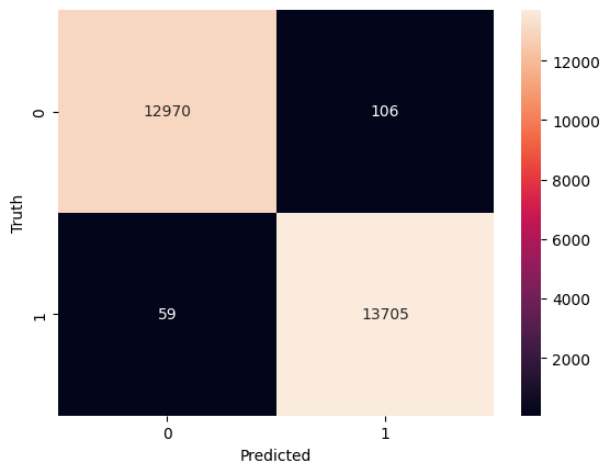


Figure 10. Confusion Matrix of Best Model

Figures 9 and 10 illustrate that our proposed model made more accurate classifications compared to traditional machine learning models (The confusion matrix shown in figure 11). When examining the number of misclassifications, our proposed model made nearly 50 fewer errors. After performing hyperparameter optimization using Grid Search, the best configuration was identified and subsequently used to train the final model. The model with the optimal hyperparameters was retrieved using `tuner.get_best_models(num_models=1)[0]`, and was trained for 10 epochs. Upon completion of training, the model demonstrated exceptional performance on the training data, achieving a loss of 0.0040 and an accuracy of 0.9995. This indicates that the model was able to almost perfectly fit the training data, with only minimal error. The low training loss suggests that the model's predictions closely matched the actual labels, while the extremely high accuracy indicates that very few classification errors occurred during training. The model was also evaluated on a separate validation dataset, where it achieved a

validation loss of 0.0388 and a validation accuracy of 0.9939. Although the validation accuracy is slightly lower than the training accuracy, this still represents outstanding performance. The slight increase in validation loss and reduction in accuracy suggests that the model generalized well to unseen data, with only a marginal degree of overfitting, if any. The gap between the training and validation results is relatively small, indicating that the model maintained strong predictive power even on data it had not encountered during training.

The confusion matrix further illustrates the model's classification performance. It is structured as follows: The confusion matrix provides a detailed breakdown of the model's classification results. Out of 13076 samples in the first class (true negatives), the model correctly identified 12970, with only 106 misclassified as false positives. For the second class (true positives), the model correctly identified 13705 out of 13764 samples, with 59 misclassified as false negatives. These results demonstrate a strong balance between precision and recall for both classes. Specifically, the model achieved a very low false positive rate (106 out of 13076) and a similarly low false negative rate (59 out of 13764). This shows that the model was able to correctly distinguish between the two classes with high reliability. In summary, the model performed remarkably well, achieving near-perfect accuracy and exhibiting only minor misclassifications in both positive and negative classes. The combination of low training and validation losses, coupled with high accuracy scores and a well-balanced confusion matrix, suggests that the model is highly effective for this classification task, with minimal overfitting and strong generalization capabilities.

### Comparing with Classical Methods

As we discussed in Deep Learning section, we carried out data balancing and useful text preprocessing steps. Then in contrast to deep learning model based on transformer architecture, we used Count Vectorizer and TF-IDF vectorizer to get text embeddings. After that we put it to test 12 machine learning models (Logistic Regression, Decision Tree Classifier, Extra Tree Classifier, XGB Classifier, XGBRF Classifier, AdaBoost

Classifier, Random Forest Classifier, Extra Trees Classifier, Gradient Boosting Classifier, Bagging Classifier, SGD Classifier, Support Vector Classifier) to classify the tweets with their default constructors.

According to the results, Support Vector Classifier stands out as the machine learning model with the highest accuracy level with a value of 98.94. When we give the training results of our Support Vector Classifier model to the configuration matrix, we get the results as seen in Table 8.

Table 8. Result of Machine Learning Models

Model-Name	Accuracy	ROC_AUC	F1_Score	Precision	Recall
SVC	98.945171	0.989455	0.989724	0.990119	0.989328
Extra Trees Classifier	98.863170	0.988649	0.988919	0.989818	0.988022
Random Forest Classifier	98.315256	0.983191	0.983564	0.985355	0.981779
Bagging Classifier	97.655522	0.976583	0.977131	0.978733	0.975535
Logistic Regression	97.640613	0.976310	0.977089	0.974303	0.979891
Decision Tree Classifier	97.368519	0.973600	0.974435	0.972112	0.976770
SGD Classifier	97.200790	0.971870	0.972856	0.968759	0.976987
XGB Classifier	96.574602	0.965311	0.967130	0.953187	0.981488
Extra Tree Classifier	92.631108	0.926129	0.928574	0.924266	0.932922
Gradient Boosting Classifier	86.410228	0.861418	0.878991	0.809661	0.961307
AdaBoost Classifier	85.303217	0.850706	0.867529	0.807442	0.937278
XGBRF Classifier	72.928547	0.722704	0.785885	0.661620	0.967623

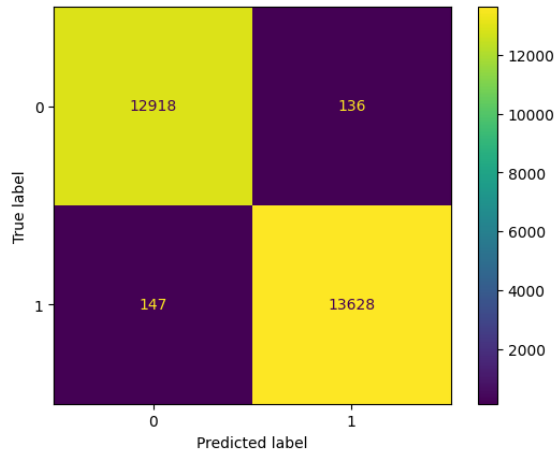


Figure 11. Confusion Matrix of SVC Model

Although we followed the same pre-processing steps and subjected the same data balancing processes to the dataset. We handled with 2 different approaches, when we compare our deep learning model, in which we use the Transformer architecture with Bidirectional LSTM layers, with classical machine learning methods, we can clearly see the difference in classification performance as seen in Figure 11. Our deep

learning-based model we created made approximately 50 fewer classification errors in both areas in classifying Fake and Real tweets than the Support Vector Classifier machine learning model.

## 4. DISCUSSION AND COMPARISON WITH OTHER STUDIES

When examining the studies presented in Table 9, we observe that numerous works in the field of fake news detection have employed machine learning and deep learning models using various datasets. The results in Table 9 demonstrate that studies in the field of fake news detection often achieve more effective outcomes when various machine learning and deep learning techniques are hybridly integrated. Models relying on a single architecture and approach tend to exhibit lower performance compared to hybrid models. Particularly, trained models, when combined with effective natural language processing approaches, demonstrate a heightened capability for high-level classification in fake news detection.

As mentioned in the Dataset section, in addition to the TruthSeeker dataset, we also conducted tests using the PolitiFact dataset to evaluate the model's

classification capabilities. During testing, we maintained the optimal parameters obtained through grid search and used the same natural language processing methods to train the model for 10 epochs, ensuring consistency for clearer comparison and more accurate inferences. This training resulted in an accuracy of 80.40%. The reason for this accuracy being lower than that achieved with the TruthSeeker dataset may be attributed to the PolitiFact dataset not having a sufficient number of instances for the model to learn all patterns effectively. Additionally, labels such as 'mostly-true' and 'barely-true' in the PolitiFact dataset may introduce ambiguity, leading to less definitive conclusions and causing uncertainty in the classification process.

Drawing from the outcomes of our tests and a survey of other research in the field, it is clear that in tasks such as fake news detection, the dataset used for training is as crucial as the models and hyperparameters applied. For a model to be viable in real-world applications and deployments, it needs to be trained on data that is both diverse and extensive. Insufficient variety and volume in the training data can hinder the model's ability to generalize, increasing the risk of misclassifications when exposed to new or domain-specific scenarios. This underscores the importance of using comprehensive datasets to prevent

the model from making erroneous predictions in unfamiliar contexts and to ensure strong performance in practical environments. Furthermore, a diverse dataset helps reduce biases and improves the model's flexibility, enabling it to operate effectively across a broad spectrum of subjects and situations.

## 5. CONCLUSION AND FUTURE STUDY

This research presents a novel model aimed at identifying fake news on social media, addressing an increasingly critical concern for society. Our approach leverages the BERT Transformer architecture, renowned for its efficacy in natural language processing tasks. To enhance the model's effectiveness and accuracy in classifying information, we integrated Bidirectional LSTM layers, a widely adopted technique in the field. Prior to feeding data into the model, we employed comprehensive text-cleaning methods to eliminate irrelevant words, symbols, and usernames from social media content. Furthermore, we standardized commonly used social media abbreviations to their full forms, ensuring clarity in the text input. To mitigate bias and classification errors within the target categories of our dataset, we utilized the Tomek Links algorithm, which further refined our data.

Table 9. Comparison of Our Model with Previous Studies on Fake News Detection

Work	Year	Method	Dataset / Inputs	Performance
<b>Proposed Model</b>	<b>2024</b>	<b>BERT, BiLSTM</b>	<b>TruthSeeker, PolitiFact</b>	<b>99,90%, 80.40%</b>
Seddari et al.[29]	2022	Hybrid approach that consists of language and knowledge-based methods	BuzzFeedNews	94.4%
Sahoo et al.[30]	2021	LSTM	FakeNewsNet	99.4%
Jarrahi et al.[31]	2021	UPFD framework	PolitiFact, Gossipcop	90.6%, 97.8%
Wang et al. [32]	2021	BERT, BiLSTM, CNN	COVID-19	93.47%
Ni et al. [33]	2021	Multi-View Attention Networks	Twitter15, Twitter16	92.34%, 93.65%
Lu et al.[34]	2020	Graph-aware CoAttention Networks (GCAN)	Twitter15, Twitter16	87.67%, 90.84%
Zhou et al.[35]	2020	Supervised model using linguistic and psychological features to detect fake news	PolitiFact, BuzzFeedNews	60% -70% 50%-60%
Shu et al. [36]	2019	Linguistic and structural approaches (STFN-HPFN)	PolitiFact, Gossipcop	85.6%, 86.3%
Kesarwani et al. [37]	2020	K-Nearest Neighbor classifier	BuzzFeedNews	79.0%
Yang et al. [38]	2019	UFD, Gibbs sampling	LIAR, BuzzFeedNews	75.9%, 67.9%



Shu et al. [39]	2019	LSTM	PolitiFact, BuzzFeed	67%, 74.2%
Traylor et al. [40]	2019	SciPy, NLP, Textblob	News Articles	63.3%
Rasool et al.[41]	2019	Dataset relabeling and iterative learning	LIAR	66.29%
Kayakuş et al. [42]	2023	Naive Bayes, Decision Trees	Twitter API	89.3 %, 84.2%
Taşkın et al. [21]	2021	Supervised and unsupervised learning algorithms	Twitter API	86.0%, 72.0%
Koru et al. [20]	2024	BERT, Bi-LSTM, CNN	BuzzFeedNews, GossipCop and other 5 datasets	94%

A significant emphasis was placed on hyperparameter optimization to enhance model performance. Each hyperparameter was meticulously evaluated through Grid Search, leading to a training process involving a vast array of parameter combinations. This extensive tuning resulted in all top 10 models demonstrating exceptional performance, with accuracy rates exceeding 99%. The practical implications of this research are significant. The model's application can extend to various social media platforms, where the spread of misinformation poses substantial risks. By providing real-time detection capabilities, our model could assist users in discerning credible information from false narratives, thereby fostering a more informed society. However, it is essential to acknowledge potential risks associated with implementing such technology, including reliance on automated systems and the challenge of adapting to the evolving nature of misinformation. Looking ahead, future studies will focus on enhancing the model's robustness against adversarial attacks. This involves investigating the model's vulnerability to manipulated inputs designed to deceive it, as well as developing techniques to strengthen its resilience. By addressing these challenges, we aim to ensure that our fake news detection system remains reliable and effective in the face of sophisticated misinformation tactics.

### Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

### Conflict of interest

The authors declare that they have no conflict of interest.

### Data availability statement

The datasets generated during and/or analyzed during the current study are available from the authors on reasonable request.

### REFERENCES

- [1] Janelle B. Hill, "Gartner insights on spotting and responding to digital disruption Leading Through Digital Disruption EDITED BY Janelle B. Hill, Gartner Research Vice President and Distinguished Analyst," 2017. Accessed: Jan. 04, 2025. [Online]. Available: [https://www.gartner.com/imagesrv/books/digital-disruption/pdf/digital\\_disruption\\_ebook.pdf](https://www.gartner.com/imagesrv/books/digital-disruption/pdf/digital_disruption_ebook.pdf)
- [2] G. Mavridis, "Fake news and Social Media: How Greek users identify and curb misinformation online," 2018, Accessed: Jan. 03, 2025. [Online]. Available: <https://urn.kb.se/resolve?urn=urn:nbn:se:mau:diva-23196>
- [3] S. Dadkhah, X. Zhang, A. G. Weismann, A. Firouzi, and A. A. Ghorbani, "TruthSeeker: The Largest Social Media Ground-Truth Dataset for Real/Fake Content." [Online]. Available: <https://www.unb.ca/cic/datasets/truthseeker-2023.html>.
- [4] S. Sharma, M. Saraswat, and A. K. Dubey, "Fake news detection on Twitter," *International Journal of Web Information Systems*, vol. 18, no. 5–6, pp. 388–412, Dec. 2022, doi: 10.1108/IJWIS-02-2022-0044.
- [5] A. Ali and M. Gulzar, "An Improved FakeBERT for Fake News Detection," *Applied Computer Systems*, vol. 28, no. 2, pp. 180–188, Dec. 2023, doi: 10.2478/acss-2023-0018.
- [6] R. K. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach," *Multimed Tools Appl*, vol. 80, no. 8, pp. 11765–11788, Mar. 2021, doi: 10.1007/s11042-020-10183-2.
- [7] H. Alsaidi and W. Etaïwi, "Empirical Evaluation of Machine Learning Classification Algorithms for Detecting COVID-19 Fake News," *International Journal of Advances in Soft Computing and its Applications*, vol. 14, no.



- 1, pp. 49–59, 2022, doi: 10.15849/IJASCA.220328.04.
- [8] A. M. Ali, F. A. Ghaleb, B. A. S. Al-Rimy, F. J. Alsolami, and A. I. Khan, “Deep Ensemble Fake News Detection Model Using Sequential Deep Learning Technique,” *Sensors*, vol. 22, no. 18, Sep. 2022, doi: 10.3390/s22186970.
- [9] B. Fang and H. Zhou, “Fake news text detection based on convolutional neural network,” *Applied and Computational Engineering*, vol. 41, no. 1, pp. 202–209, Feb. 2024, doi: 10.54254/2755-2721/41/20230744.
- [10] “(PDF) ULMFiT for Twitter Fake News Spreader Profiling Notebook for PAN at CLEF 2020.” Accessed: Jan. 04, 2025. [Online]. Available: [https://www.researchgate.net/publication/359024571\\_ULMFiT\\_for\\_Twitter\\_Fake\\_News\\_Spreader\\_Profiling\\_Notebook\\_for\\_PAN\\_at\\_CLEF\\_2020](https://www.researchgate.net/publication/359024571_ULMFiT_for_Twitter_Fake_News_Spreader_Profiling_Notebook_for_PAN_at_CLEF_2020)
- [11] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake News Detection on Social Media,” *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, Sep. 2017, doi: 10.1145/3137597.3137600.
- [12] O. Stitini and S. Kaloun, “An improved self-training model to detect fake news categories using multi-class classification of unlabeled data: fake news classification with unlabeled data,” *Int. J. Systematic Innovation*, vol. 8, no. 1, p. 4, 2024, doi: 10.6977/IJoSI.202403\_8(1)0002.
- [13] V. Balakrishnan, H. L. Zing, and E. Laporte, “COVID-19 INFODEMIC – UNDERSTANDING CONTENT FEATURES IN DETECTING FAKE NEWS USING A MACHINE LEARNING APPROACH,” *Malaysian Journal of Computer Science*, vol. 36, no. 1, pp. 1–13, 2023, doi: 10.22452/mjcs.vol36no1.1.
- [14] A. K. Yadav et al., “Fake News Detection Using Hybrid Deep Learning Method,” *SN Comput Sci*, vol. 4, no. 6, pp. 1–15, Nov. 2023, doi: 10.1007/S42979-023-02296-W/METRICS.
- [15] M. Park and S. Chai, “Constructing a User-Centered Fake News Detection Model by Using Classification Algorithms in Machine Learning Techniques,” *IEEE Access*, vol. 11, pp. 71517–71527, 2023, doi: 10.1109/ACCESS.2023.3294613.
- [16] “(13) (PDF) DISTILBERT FOR WEB SECURITY: ENHANCED DETECTION OF XSS ATTACKS USING NLP APPROACH.” Accessed: Jul. 17, 2024. [Online]. Available: [https://www.researchgate.net/publication/381659932\\_DISTILBERT\\_FOR\\_WEB\\_SECURITY\\_ENHANCED\\_DETECTION\\_OF\\_XSS\\_ATTACKS\\_USING\\_NLP\\_APPROACH](https://www.researchgate.net/publication/381659932_DISTILBERT_FOR_WEB_SECURITY_ENHANCED_DETECTION_OF_XSS_ATTACKS_USING_NLP_APPROACH)
- [17] R. Ghanem, H. Erbay, and K. Bakour, “Contents-Based Spam Detection on Social Networks Using RoBERTa Embedding and Stacked BLSTM,” *SN Comput Sci*, vol. 4, no. 4, pp. 1–15, Jul. 2023, doi: 10.1007/S42979-023-01798-X/METRICS.
- [18] R. Ghanem and H. Erbay, “Spam detection on social networks using deep contextualized word representation,” *Multimed Tools Appl*, vol. 82, no. 3, pp. 3697–3712, Jan. 2023, doi: 10.1007/S11042-022-13397-8/METRICS.
- [19] A. Makalesi, R. Article Rezan BAKIR, H. Erbay, and H. Bakir, “ALBERT4Spam: A Novel Approach for Spam Detection on Social Networks,” no. 2, p. 17, doi: 10.17671/gazibtd.1426230.
- [20] G. K. Koru and C. Uluyol, “Detection of Turkish Fake News from Tweets with BERT Models,” *IEEE Access*, vol. 12, pp. 14918–14931, 2024, doi: 10.1109/ACCESS.2024.3354165.
- [21] S. G. TAŞKIN, E. U. KÜÇÜKSİLLE, and K. TOPAL, “Twitter üzerinde Türkçe sahte haber tespiti,” *Balikesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, vol. 23, no. 1, pp. 151–172, Jan. 2021, doi: 10.25092/baunfbed.843909.
- [22] N. A. S. Abdullah, N. I. A. Rusli, and N. S. Yuslee, “Development of a machine learning algorithm for fake news detection,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 35, no. 3, pp. 1732–1743, Sep. 2024, doi: 10.11591/ijeecs.v35.i3.pp1732-1743.
- [23] R. Bakır and H. Bakır, “Swift Detection of XSS Attacks: Enhancing XSS Attack Detection by Leveraging Hybrid Semantic Embeddings and AI Techniques,” *Arab J Sci Eng*, pp. 1–17, Jun. 2024, doi: 10.1007/S13369-024-09140-0/TABLES/14.
- [24] H. Bakır and G. Tarihi, “Using Transfer Learning Technique as a Feature Extraction Phase for Diagnosis of Cataract Disease in the Eye,” *USBTU*, vol. 1, no. 1, p. 2022.
- [25] H. Bakır and K. Elmabruk, “Deep learning-based approach for detection of turbulence-induced distortions in free-space optical communication links,” *Phys Scr*, vol. 98, no. 6, p. 065521, May 2023, doi: 10.1088/1402-4896/ACD4FA.
- [26] U. Demircioğlu, A. Sayıl, and H. Bakır, “Detecting Cutout Shape and Predicting Its Location in Sandwich Structures Using Free Vibration Analysis and Tuned Machine-Learning Algorithms,” *Arab J Sci Eng*, vol. 49, no. 2, pp. 1611–1624, Feb. 2024, doi: 10.1007/S13369-023-07917-3/METRICS.
- [27] J. Bergstra, J. B. Ca, and Y. B. Ca, “Random Search for Hyper-Parameter Optimization Yoshua Bengio,” 2012. [Online]. Available: <http://scikit-learn.sourceforge.net>.
- [28] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” Oct. 2019, [Online]. Available: <http://arxiv.org/abs/1910.01108>
- [29] N. Seddari, A. Derhab, M. Belaoued, W. Halboob, J. Al-Muhtadi, and A. Bouras, “A Hybrid Linguistic and Knowledge-Based Analysis Approach for Fake News Detection on Social Media,” *IEEE Access*, vol. 10, pp.

- 62097–62109, 2022, doi: 10.1109/ACCESS.2022.3181184.
- [30] S. R. Sahoo and B. B. Gupta, “Multiple features based approach for automatic fake news detection on social networks using deep learning,” *Appl Soft Comput*, vol. 100, p. 106983, Mar. 2021, doi: 10.1016/J.ASOC.2020.106983.
- [31] A. Jarrahi and L. Safari, “FR-Detect: A Multi-Modal Framework for Early Fake News Detection on Social Media Using Publishers Features,” Sep. 2021, Accessed: Sep. 24, 2024. [Online]. Available: <https://arxiv.org/abs/2109.04835v1>
- [32] Y. Wang, Y. Zhang, X. Li, and X. Yu, “COVID-19 Fake News Detection Using Bidirectional Encoder Representations from Transformers Based Models,” Sep. 2021, Accessed: Sep. 24, 2024. [Online]. Available: <https://arxiv.org/abs/2109.14816v2>
- [33] S. Ni, J. Li, and H. Y. Kao, “MVAN: Multi-View Attention Networks for Fake News Detection on Social Media,” *IEEE Access*, vol. 9, pp. 106907–106917, 2021, doi: 10.1109/ACCESS.2021.3100245.
- [34] Y. J. Lu and C. Te Li, “GCAN: Graph-aware co-attention networks for explainable fake news detection on social media,” *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 505–514, 2020, doi: 10.18653/V1/2020.ACL-MAIN.48.
- [35] “(3) (PDF) Fake News Early Detection: A Theory-driven Model.” Accessed: Sep. 24, 2024. [Online]. Available: [https://www.researchgate.net/publication/332726212\\_Fake\\_News\\_Early\\_Detection\\_A\\_Theory-driven\\_Model](https://www.researchgate.net/publication/332726212_Fake_News_Early_Detection_A_Theory-driven_Model)
- [36] K. Shu, S. Wang, H. Liu, and D. Mahudeswaran, “Hierarchical Propagation Networks for Fake News Detection: Investigation and Exploitation”, doi: 10.48550/arXiv.1903.09196.
- [37] A. Kesarwani, S. S. Chauhan, and A. R. Nair, “Fake News Detection on Social Media using K-Nearest Neighbor Classifier,” *Proceedings of the 2020 International Conference on Advances in Computing and Communication Engineering, ICACCE 2020*, Jun. 2020, doi: 10.1109/ICACCE49060.2020.9154997.
- [38] S. Yang, K. Shu, S. Wang, R. Gu, F. Wu, and H. Liu, “Un-supervised fake news detection on social media: A generative approach,” *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pp. 5644–5651, 2019, doi: 10.1609/AAAI.V33I01.33015644.
- [39] K. Shu, D. Mahudeswaran, and H. Liu, “FakeNewsTracker: a tool for fake news collection, detection, and visualization,” *Comput Math Organ Theory*, vol. 25, no. 1, pp. 60–71, Mar. 2019, doi: 10.1007/s10588-018-09280-3.
- [40] T. Traylor, J. Straub, Gurmeet, and N. Snell, “Classifying Fake News Articles Using Natural Language Processing to Identify In-Article Attribution as a Supervised Learning Estimator,” *Proceedings - 13th IEEE International Conference on Semantic Computing, ICSC 2019*, pp. 445–449, Mar. 2019, doi: 10.1109/ICOSC.2019.8665593.
- [41] T. Rasool, W. H. Butt, A. Shaukat, and M. U. Akram, “Multi-label fake news detection using multi-layered supervised learning,” *ACM International Conference Proceeding Series*, pp. 73–77, Feb. 2019, doi: 10.1145/3313991.3314008.
- [42] M. KAYAKUŞ and F. YİĞİT AÇIKGÖZ, “Twitter’da Makine Öğrenmesi Yöntemleriyle Sahte Haber Tespiti,” *Abant Sosyal Bilimler Dergisi*, vol. 23, no. 2, pp. 1017–1027, Jul. 2023, doi: 10.11616/asbi.1266179.