

Guarding the Truth: Enhancing Fake Headline Detection using Transformer-Based Encoding and Deep Learning Methods

Mohammed Alghobiri

Abstract-Identifying fake news headlines is important in combating misinformation and remains an active research domain in Natural Language Processing (NLP). Traditional text encodings like CountVectorization and Term Frequency Times Inverse Document Frequency (TF-IDF) have limitations in capturing context and semantic information, leading to suboptimal performance in complex NLP tasks. This research introduces an approach utilizing sentence transformers to produce sentence embeddings that preserve both the semantic meaning and contextual information within the text. We aim to identify false news headlines by utilizing an array of deep learning models such as LSTM, BiLSTM, BERT, DistilBERT, and RoBERTa in conjunction with diverse embeddings including TF-IDF, GloVe, fastText, and sentence transformers, complemented by various machine learning algorithms like naïve Bayes, decision trees, random forest, and AdaBoost. The experiments, conducted on the Artificial Intelligence (AI) Open news headlines dataset, reveal that sentence transformers consistently outperform conventional encodings, demonstrating higher accuracy and F1 scores. Among the deep learning models, RoBERTa achieved the highest accuracy, reaching 95.48% with GloVe embeddings and 96.17% with sentence transformers. These empirical findings underline the superiority of our proposed approach over existing methods, offering valuable insights for effectively identifying fake headlines in news articles.

Keywords-natural language processing (NLP), fake headline detection, transformer-based encoding; sentence transformers; RoBERTa

I. INTRODUCTION

The rapid expansion of social media has revolutionized news accessibility, yet it has also given rise to a critical problem: the proliferation of fabricated news [1]. The repercussions of misinformation are profound, often spreading more swiftly than genuine news [2]. The term "fake news" encompasses deceptive or inaccurate information presented as legitimate news, commonly aimed at deceiving or manipulating readers. Fake news refers to the presentation of deceptive or inaccurate information as authentic news, often with the intent to mislead or manipulate readers. This dissemination of false information, akin to yellow journalism [3], involves intentionally distributing false news or hoaxes across various channels, including traditional print, social media platforms, and broadcast news media.

On the other hand, the spread of misinformation has broad implications across religious beliefs, social

connections, scientific understanding, and technology [4]. The dissemination of false information, resembling a form of asymmetric attack using social engineering tactics, has spurred a new interdisciplinary area of research within natural language processing (NLP), capturing the attention of computer science and other disciplines [5]. News encompasses various elements like text, audio, visuals, and videos, posing a significant challenge to create a systematic tool capable of accurately assessing the truthfulness of information due to its vast volume, diverse content, and rapid dissemination [6], [7].

In recent studies, researchers have extensively explored automated fake news detection using a mix of machine learning (ML) and deep learning (DL) techniques [8]. However, it is important to recognize potential challenges, such as dataset biases and varying performance when applied to news spanning different subjects [9]. Therefore, evaluating a range of models across diverse datasets is necessary to comprehensively assess their effectiveness [10]. Researchers aim to improve ML accuracy in identifying deceptive content by utilizing large and varied datasets for training [11]. These results emphasize the importance of robust training methods to create dependable approaches for detecting fake headlines.

Recent research has revealed the prevalence of sequence neural networks in encoding both news content and contextual social information [12]. Despite the widespread use of the convolutional neural network (CNN) DLD architecture for NLP tasks, its effectiveness faces challenges in capturing long-distance relationships and maintaining a coherent correlation between local and global characteristics due to limitations within its pooling layer. However, more recently, a fusion of word embedding techniques with CNN architecture has emerged. For instance, the bow-CNN model effectively replaces the convolution layer with the bag-of-words (BoW) approach, effectively integrating word order details within feature vectors [13]. Similarly, the enhanced word embedding (EWE) method, combines pre-trained word embeddings with various CNN modules to fortify the word embedding model. EWE strategically integrates syntactic, lexical, and positional features, addressing the challenge of assigning appropriate weights to different words while learning long-distance dependencies. Nevertheless, despite these improvements,

these models encounter difficulties in handling complex long-range relationships among words [14].

Furthermore, existing word embedding techniques such as fastText and Word2vec grapple with efficiently minimizing information duplication. Particularly for short text segments like headlines, word embedding techniques fall short compared to the capabilities of recurrent neural networks (RNNs) and CNNs in capturing local information effectively. Consequently, these state-of-the-art approaches excel at learning either local or global features but struggle to concurrently extract both. To address these challenges, this research aims to propose a novel architecture for headline classification, employing a blend of Machine Learning (ML) and Deep Learning (DL) methods in conjunction with diverse word embeddings. The key contributions of this study include:

- Proposing a unique framework for identifying fake headlines using a variety of DL algorithms in combination with GloVe word embeddings and sentence transformer-based encoding
- Applying various Natural Language Processing (NLP) feature engineering techniques to preprocess news headlines, enabling the extraction of both dense and sparse features from textual data
- Exploring the effectiveness of traditional encoding techniques compared to modern deep word embeddings and transformer encoding concerning ML and DL methods
- Conducting extensive empirical analysis utilizing a benchmark fake headline dataset to evaluate the performance of the proposed model, facilitating a comparative assessment against alternative methods

The subsequent sections describe the organization of this study: Section 2 provides an in-depth literature review, highlighting recent advancements in fake headline detection; Section 3 presents the architecture of the proposed research methodology; Section 4 comprehensively covers the dataset, experimental setup, and empirical findings, offering a comparative analysis against state-of-the-art approaches in fake headline detection; Finally, in section 5, we conclude based on the research findings from section 4 and presents potential avenues for future work.

II. RELATED WORK

Several studies have employed machine learning (ML) techniques to address the detection of fake news through various methodologies. For Instance, this study of Ahmed et al. [15] highlights the evolving field of fake news detection facing resource constraints by introducing a detection model that employs n-gram analysis and machine learning techniques. The results of the shows an

accuracy of 92% with Term Frequency-Inverted Document Frequency (TF-IDF) as the extraction method and Linear Support Vector Machine (LSVM) as the classifier. Similarly, Jain et al. [16] proposed a system that identifies fake news in addition to suggesting authentic articles. The study employs Naive Bayes classifier, SVM (Support Vector Machine), and NLP (Natural Language Processing) for combating misinformation and reveals that not all fake news originates from social media. Subsequently, Bali et al. [17] introduced a new set of features extracted from both headlines and content using gradient boosting and reaching an accuracy of 88%.

On the other hand, the enhanced transformer-based models, specifically BERT (Bidirectional Encoder Representations from Transformers), supplemented by an optimized BERT pretraining approach known as RoBERTa, significantly improved the effectiveness of BERT's deep-contextualization compared to older state-of-the-art models in this particular task [13]. Similarly, the study identifies challenges in misinformation detection, such as underutilization of multi-modal techniques, oversight in source verification, and the need to consider author credibility [18]. The results of the study demonstrate the impact of context learning methods, dataset size, and vocabulary dimension on transformer models' accuracy in misinformation detection.

Research identifies that reasoning plays an important role in differentiating between low and high quality political news [19]. Consequently, Umer et al. [20] utilized advanced deep learning models, including long short-term memory (LSTM) and bidirectional encoder representations from transformers (BERT), achieving an accuracy of 93%. Likewise, Wang et al. [8] employed a variety of algorithms, including LSTM (Long Short-Term Memory), deep belief network (DBN), and CNN (Convolutional Neural Network), for discerning fake news from textual data. The authors highlighted that misleading headlines constitute one of the primary reasons why individuals engage with news content in the first place. Nevertheless, Sepúlveda-Torres et al. [21] employed automatic news summaries to assess the alignment between a headline and its associated body text by proposing a two-stage method that employs summary techniques as inputs for classifiers, reducing the data processed while retaining critical information. The findings demonstrate that the utilization of automatic extractive summaries significantly supports in assessing the alignment of concise information, such as headlines or sentences, with their complete content and achieves and accuracy of 94.13%, surpassing current benchmarks.

Furthermore, in [4], the authors introduced an innovative system designed to identify fake news articles by leveraging content-based features and Extreme Gradient Boosting Tree (xgbTree) algorithm that is

optimized through the Whale Optimization Algorithm (WOA) to classify 91% of news articles based on the extracted features. Additionally, Kaliyar et al. [22] introduced FakeBERT, a BERT-based deep learning model that combines parallel blocks of single-layer Convolutional Neural Networks (CNN) with varying kernel sizes and filters, surpassing benchmarks with an accuracy of 98.90%. FakeBERT utilizes BERT as a sentence encoder, achieving precise context representation and automatically identifying optimal feature sets without manual engineering. Conversely, Hande et al. [23] introduced a specialized system for identifying COVID-19-related misinformation, leveraging verified knowledge. The proposed system uses pre-trained transformer-based models and explores different loss functions, including a novel one, showcasing that the approach, especially when paired with domain-specific models, yields the best results, reaching an accuracy of 98% in identifying fake news. Similarly, research identifies that LSTM models with attention mechanisms and CNN models incorporating global and local word contexts show promising performance [24]. Moreover, using custom-trained word embeddings, especially Word2Vec Skip-Gram models, significantly boosts accuracy by capturing local word context effectively.

Subsequently, Szczepański et al. [25] utilized a BERT-based model to classify fake news in short news headlines. The authors emphasize the necessity of an explainability approach for improved comprehension. Additionally, Bsoul et al. [26] curated a dataset aimed at distinguishing between clickbait and non-clickbait text. The study utilized sequential deep learning (DL)-based techniques for a binary classification task, focusing on headlines. Furthermore, Ali et al. [27] introduced a multi-class classifier employing sequential DL-based techniques, notably a multi-layer perceptron. The proposed model demonstrated significant enhancements over previously established state-of-the-art methods. Similarly, research achieved an accuracy of up to 92% in identifying fake news employing a multi-modal DL-based model [28]. In contrast, the authors employed CNN-based capsule networks along with BERT for a detailed analysis in their fake news detection architecture [29]. The proposed architecture integrates various DL models, including LSTM, CNN, and ResNet, with pre-trained word embedding models to effectively detect fake news. Besides, bidirectional LSTM showcased superior performance compared to other DL-based techniques in fake news detection.

Recently, the study of Fayyaz et al. [30] utilized a Random Forest (RF) classifier, extracting twenty-three textual features from the ISOT Fake News Dataset. Additionally, four feature selection techniques—Chi2, Univariate, information gain, and Feature importance—were utilized to select the fourteen best features. The proposed model, evaluated against benchmark techniques,

surpassed state-of-the-art machine learning models such as GBM, XGBoost, and AdaBoost Regression Model in terms of classification accuracy. Subsequently, research identifies that deep-learning models stand out as the most effective and accurate approach for identifying disinformation [31]. In addition, the proposed research systematically consolidates multiple contemporary strategies used to detect fake news, encompassing their methodologies, outcomes, limitations, and potential challenges. Similarly, Rai et al. [32] investigated the performance of various DL models, including LSTM, Bi-LSTM, BERT, DistilBERT, and RoBERTa, to provide a comprehensive evaluation of their effectiveness in detecting fake headlines. On the other hand, Truică et al. [33] employed two BiLSTM neural networks, coupled with sentence transformers, for assessing fake news authenticity. The study identifies the challenges in identifying similarities between English and German texts, leading to poor performance despite the theoretical advantages of cross-lingual transformers and transfer learning.

Furthermore, Truică and E. S. Apostol analyzed different BERT models and identified that BERT base and large models display minimal performance distinctions despite significant differences in training time [34]. Despite notable differences in runtime, BERT and DistilRoBERTa demonstrate high accuracy, yet MisRoBERTa excels in both performance and efficiency. Additionally, multilingual models like XLM do not surpass BERT base accuracy. In a similar study, the authors introduced Ember, a novel approach to fake news detection inspired by how readers verify news components [35]. Ember approaches the fake news problem by considering news from a component-level perspective. Specifically, we redefine the detection challenge as a fusion problem involving multiple components, aiming to extract both within-component and between-component features. Additionally, Ember can adapt to different datasets by adjusting the number of feature extractors for intra- and inter-component analysis. Likewise, Nadeem et al. [36] introduced a novel approach, the Stylometric and Semantic Similarity-oriented for Multimodal Fake News Detection (SSM). SSM consists of five modules: Firstly, a Hyperbolic Hierarchical Attention Network (Hype-HAN) extracts stylometric textual features. Secondly, it generates news content summaries and computes similarity measures between headlines and summaries. Thirdly, it calculates semantic similarities between visual and textual elements. Fourthly, it analyzes images for potential forgery. Finally, it fuses these extracted features for final classification.

On the other hand, in a detailed review by Agarwal et al. [37], presented the empirical analysis of various ML and ensemble approaches for fake news classification tasks. Among the ensemble techniques that combined multiple models to boost accuracy and robustness, AdaBoost showed promising results. Finally, Truică and

E. S. Apostol proposed DOCEMB (Document Embeddings) that combines TFIDF, WORD2VEC, FASTTEXT, GLOVE, BERT, ROBERTA, and BART to spot misinformation effectively [38]. The research shows that simpler machine learning models with DOCEMB outperform or match the performance of complex neural networks designed for fake news detection. In addition, the results highlight the importance of document encoding for accuracy over intricate classification architecture.

III. PROPOSED RESEARCH METHODOLOGY

Figure 1 presents the research framework proposed for detecting fake headlines. The first step involves text preprocessing of the dataset using NLP techniques such as tokenization, lemmatization, stop word removal, and part-of-speech (POS) tagging. Secondly, diverse encoding techniques are used, such as TF-IDF, GloVe, fastText, and sentence transformers. Subsequently, ML, ensemble-based, and DL models are trained to identify fake news headlines. Finally, the computational performance is assessed, and the classification results are measured.

Text preprocessing of raw headlines plays a pivotal role in our proposed research framework for fake headline detection. It involves using preprocessing techniques aimed at converting headlines into a more suitable format for subsequent encoding and modeling tasks.

B Tokenization

Tokenization involves breaking down of text into words, commonly referred to as tokens. In this work, we employ a regex-based tokenizer (RET), which uses regular expressions to identify and separate tokens based on specific patterns or rules. It allows for fine-grained control over the tokenization process, enabling us to handle complex linguistic structures and capture meaningful units of text accurately.

C Lemmatization

Lemmatization, a linguistic technique used in the context of fake headline text, aims to transform words into their base or canonical form, known as the lemma. It enhanced text analysis and comprehension of fake headlines. Unlike stemming, which solely removes prefixes and suffixes, lemmatization considers the context and part of speech of each word.

A Text Preprocessing

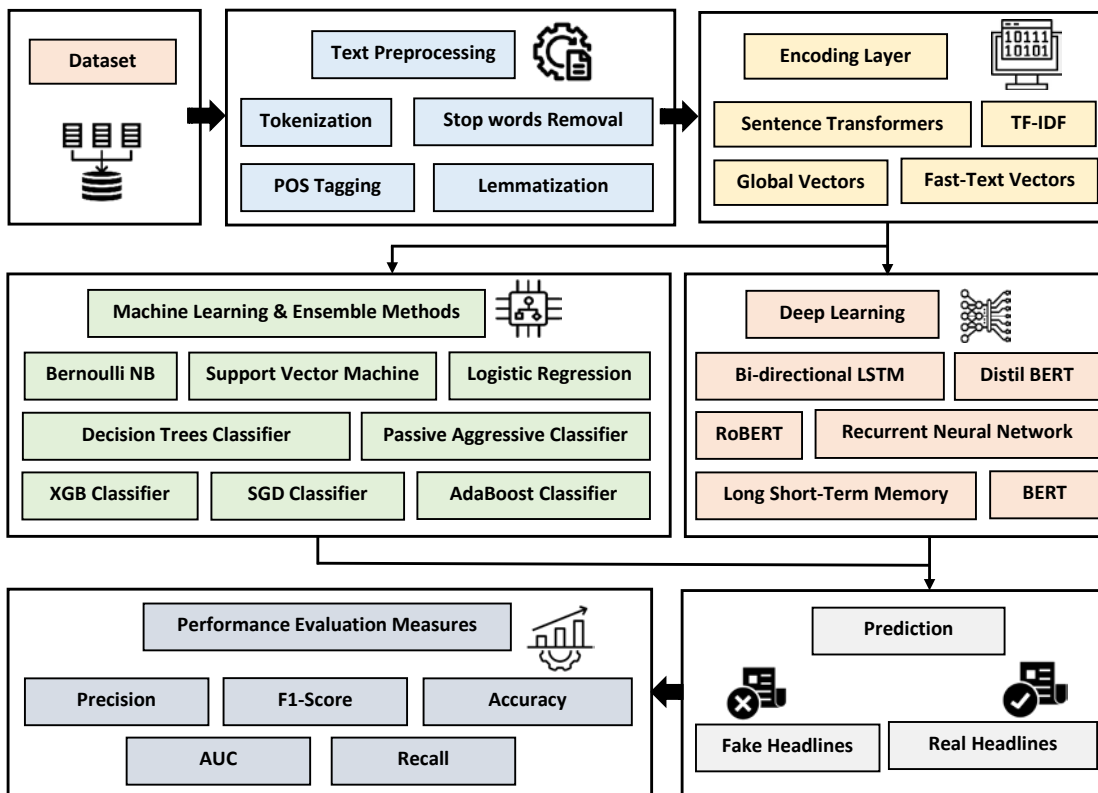


Figure 1. Framework of our Proposed Research Methodology for Fake Headline Detection

D Stop Word Removal

Fake headlines often contain common words that carry little meaningful information do not contribute significantly to the identification of fake news. Stop word removal aids in reducing noise and irrelevant features, thereby improving the efficiency and effectiveness of subsequent analysis and classification algorithms used to detect fake headlines.

E Part-of-Speech Tagging (POS)

POS tagging enables the identification of the syntactic role of each word, such as nouns, verbs, adjectives, or adverbs. As part of the detection process, it helps analyze the linguistic characteristics and structures of headlines, which enables us to distinguish between genuine and fake news.

F Encoding Layer

Text encoding plays a significant role in identification of fake headlines as it converts textual data into numerical representations. This research study involves the following conventional and deep encoding approaches.

G Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a lower-level representation used to assess the significance of a term within a collection of documents by considering its frequency in the specific document compared to its frequency across the entire document collection. It reflects two essential factors: the term frequency (TF) within the document and the term's rarity across the entire document collection (IDF). The following equation (1) is used to compute the TF-IDF encoding matrix.

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

Where, $TF(t, d)$ denotes the term frequency of t in document d , measuring its occurrence within the document. However, $IDF(t)$ indicates the inverse document frequency of term t , measuring the term's rarity across the document collection. The IDF component is computed using equation (2).

$$IDF(t) = \log\left(\frac{N}{DF(t)}\right) \quad (2)$$

Where, the term N represents the document count and $DF(t)$ denotes the frequency of term t , measuring the number of documents containing t .

H Global Vectors (GloVe)

GloVe is a commonly used encoding layer introduced by Pennington et al. [39], which aims to learn the semantics of the words in a high-dimensional vector space. Unlike traditional word embedding methods, GloVe uses a global co-occurrence statistics-based approach to learn word representations. It leverages the word co-occurrence probabilities in a large corpus with the intuition that words appearing together in similar contexts are likely to share semantic relationships. To calculate the encoding, the model constructs a co-occurrence matrix in which each element denotes the frequency of two words co-occurring in each context. Then, it seeks to learn word embeddings that satisfy a particular relationship between these co-occurrence probabilities. Let us consider two words i and j with associated word vectors, denoted as w_i and w_j , and their corresponding co-occurrence probability as X_{ij} . The relationship that GloVe aims to capture is as follows:

$$w_i \cdot w_j = \log(X_{ij}) \quad (3)$$

where \cdot indicates the dot product between the word vectors. To achieve this, GloVe formulates an objective function, quantifying the difference between the dot product of word vectors and the logarithm of co-occurrence probabilities. The model then minimizes this objective function using gradient-based optimization techniques to learn the word embeddings that capture semantic relationships effectively.

I fastText Vectors

fastText has gained significant popularity in NLP tasks due to its ability to efficiently handle out-of-vocabulary words. The basic concept of fastText lies in sub-word embeddings, which enables it to correspond words as a bag or collection of character n-grams. This approach captures morphological information and allows the model to understand the compositionality of words. Mathematically, the fastText model is trained to maximize the likelihood of predicting a word based on its surrounding context words within a specified window. fastText involves maximizing the likelihood of predicting a target word given its surrounding context words within a specified window. The aim of this encoding involves maximizing the log likelihood (LL) over all the target-context pairs in the training data. LL is calculated using equation (4).

$$LL = \sum_{[w \in V]} \sum_{[c \in Context(w)]} \sum \log P\left(\frac{c}{w}\right) \quad (4)$$

Where, V indicates the vocabulary, consisting of all unique words in the training data and w is a target word from the vocabulary. Moreover, c is a context word, i.e., a word that appears within a certain window of the target

word w . The term n is the length of character n-grams considered during sub-word representation and z is the average sub-word embedding of a word. To model the conditional probability, fastText uses the Softmax activation computed as in equation (5):

$$P\left(\frac{c}{w}\right) = \frac{\exp(z_c \cdot z_w)}{\sum_{[c' \in V]} \exp(z_{c'} \cdot z_w)} \quad (5)$$

Representations of news headlines, in the process of generating sentence embeddings, this model incorporates attention mechanisms to capture the key parts of the input sentences while encoding them into dense vectors. This attention mechanism calculates the importance of each word/token in the sentence concerning the others, capturing both local and global dependencies. The encoding process involves several steps such as tokenization and sub-word encoding, followed by positional encoding to preserve word order information. Formally, given a sentence X with n tokens, the sentence

Where, z_c is the representation of vector for context word c and z_w indicates the target word w vector representation.

J Sentence Transformers

Sentence transformers are used to generate dense and fixed length encoding transformer can be represented as follows:

$$X = \{w_1, w_2, \dots, w_n\} \quad (6)$$

Where, each token w_1 in X , the transformer, generates a hidden vector representation H_i , which is the result of multiple self-attention and feed-forward layers. These layers perform computations according to the transformer architecture using trainable parameters i.e., biases and weights for tuning the model during the training process.

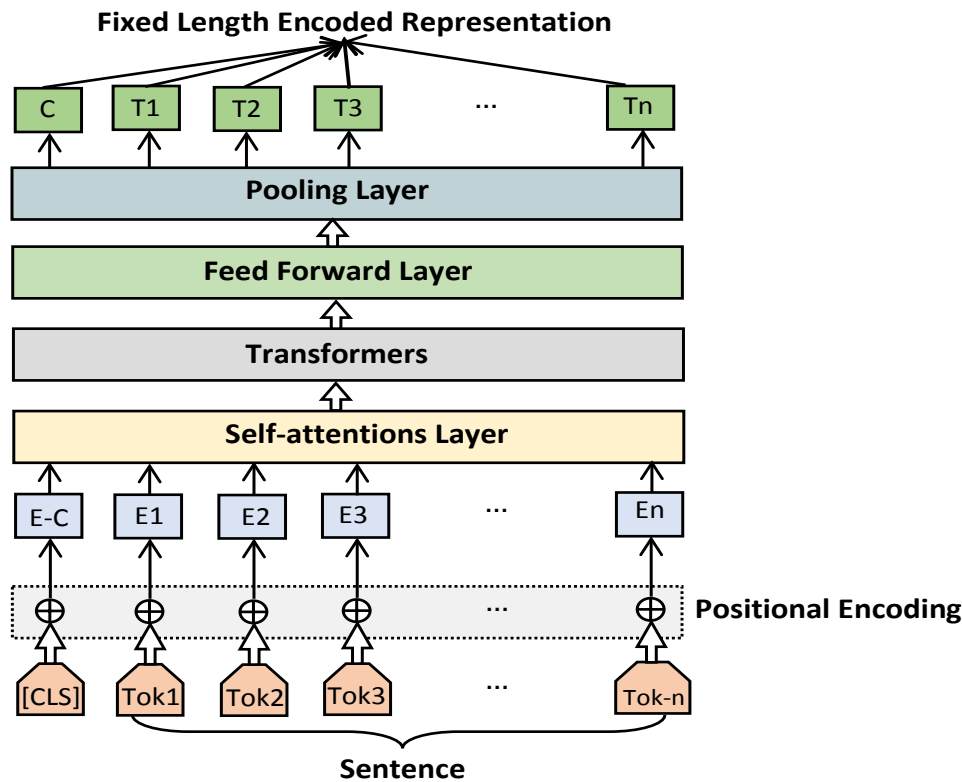


Figure 2. Architecture of Sentence Transformers for Fake Headline Detection

The self-attention mechanism includes the computation of three matrices: Q (query), K (key), and V (value) matrices, respectively. These matrices are obtained by projecting the input embeddings H_i onto lower-dimensional spaces. The dot product between Q and K is used to calculate attention scores, which are further scaled and passed through a Softmax activation to get attention weights. These weights are further used to weight the

values V and obtain the final contextual embeddings for each token. After computation of self-attention layers, the model performs feed-forward computations on each token's embeddings, passing through fully connected layers along with activation method to refine the embeddings and generate sentence representations. Then, the embeddings are pooled to obtain a fixed-length representation, which is further used for news headline classification.

K Machine Learning (ML) Methods

This research study involves diverse computational algorithms and statistical ML-based models to automatically identify and distinguish between genuine and deceptive headlines in textual content. The details of applied ML and ensemble-based models are as follows:

L Naïve Bayes

Naïve Bayes ML model relies on the principles of Bayes' theorem and assumes independence among features. The algorithm assumes that words or tokens in the text are independent from class labels. In the context of fake headlines classification, naïve Bayes computes the probability of an input headline belonging to a particular class, such as "real" or "fake," by analyzing the probabilities of individual words occurring in that headline for each class. The naïve Bayes can be computed as follows:

$$P(C|X) = \frac{(P(C) * P(X|C))}{P(X)} \quad (7)$$

Where $P(C|X)$ is the class C posterior probability for the observed features X in the text. However, $P(C)$ is the class C prior probability. Similarly, $P(X|C)$ is the likelihood probability of observing the features X given class C (probability of finding the features in the text given the class) and $P(X)$ is the probability of observing the features X in the text (probability of the given features appearing in any class).

M Support Vector Machine (SVM)

SVM is a supervised learning model that discovers the most suitable hyperplane, effectively separating data points that represent distinct classes within a high-dimensional space. Mathematically, SVM's objective is to discover the weight vector w and bias term b that satisfy the following equation (8) for all training examples, denoted by x_i for input data and y_i for the corresponding class label.

$$y_i \times (w \cdot x_i + b) \geq 1 \quad (8)$$

where y_i indicates the class label of the i -th training example, and $(w \cdot x_i + b)$ represents the decision function of the SVM. Thus, SVM maximizes the distance between the data points closest to the decision boundary associated with different classes while minimizing the classification error.

N Decision Tree (DT) Classifier

DTs can efficiently handle large feature spaces and

learn complex decision boundaries. DTs recursively split the features into subsets based on the most informative and significant features. Given a dataset with N samples and M features, the structure of the tree is built by selecting the best feature F and its corresponding threshold T to split the data such that it maximizes the information gain or minimizes the entropy. The entropy is calculated using equation (9).

$$E(p) = \sum_{i=1}^k p_i \log_2(p_i) \quad (9)$$

Where, p_i represents the probability of an instance belonging to class i in the dataset. The entropy ranges from 0 to $\log_2(K)$, where 0 indicates a pure dataset (all instances belong to the same class) and $\log_2(K)$ indicates a dataset that is entirely impure with an equal number of instances in each class. The process of computing entropy is iterated for each subset until a specified stopping criterion is reached. This iterative procedure leads to the formation of a tree-like structure, wherein every leaf node corresponds to a class label.

O Random Forest (RF) Classifier

An ensemble-based method RF is referred to as a forest of multiple DTs that work together to make predictions. After the encoding layer, the RF algorithm combines the predictions from all the individual DTs through a voting mechanism, where each tree "votes" for the most likely class label. The ultimate prediction is ascertained based on the class label that garners the highest number of votes. Let N be the number of DTs in the forest, and $F(X)$ represent the prediction of each tree for the input feature vector X . The ensemble prediction $Y(X)$ of the RF can be defined as:

$$Y(X) = \text{Max_votes}(F(X)_1, F(X)_1, F(X)_1, \dots, F(X)_1) \quad (10)$$

Where, Max_votes determines the class label with the highest occurrence among the predictions of each individual DT. This aggregation of predictions from multiple trees leads to improved accuracy and robustness in classification.

P AdaBoost Classifier

AdaBoost, short for Adaptive Boosting, aims to to enhance the precision of weak classifiers by amalgamating them into a potent classifier. The process begins with assigning equal weights to each training sample. During each iteration, a weak classifier on the data is trained and subsequently its performance is evaluated. The weight of misclassified samples has increased, making them more important for the subsequent classifier. At the end of each iteration, the

algorithm updates the sample weights based on their classification results. The iterative process persists until either a predetermined count of weak classifiers is attained or the desired accuracy is achieved. Subsequently, the final strong classifier is acquired by amalgamating the individual weak classifiers, where each one's contribution is weighted according to its performance.

Q Deep Learning (DL) Methods

ML algorithms rely on handcrafted features to learn the significant patterns from the headlines. In contrast, DL techniques possess the ability to autonomously acquire hierarchical representations of textual data, thereby resulting in enhanced and resilient feature extraction. The details of applied DL models for fake headline detection are as follows:

Long Short-Term Memory (LSTM)

LSTM is an advanced variant of RNN that accounts for the vanishing gradient problem that plagues traditional RNNs, allowing it to effectively learn long-range dependencies in sequential data like text. The LSTM cell consists of three main gates: the input gate i_t , the forget gate f_t and the output gate o_t . The gates within the cell regulate information flow and empower the LSTM to selectively retain or discard information at various time steps. LSTM gates are computed using the following equations (11-13).

$$i_t = \text{sigmoid}(W_i \times [h_{t-1}, x_t] + b_i) \quad (11)$$

$$f_t = \text{sigmoid}(W_f \times [h_{t-1}, x_t] + b_f) \quad (12)$$

$$o_t = \text{sigmoid}(W_o \times [h_{t-1}, x_t] + b_o) \quad (13)$$

Where h_{t-1} represents the previous hidden state, x_t is the input at the current time step. The W and b terms represent weight matrices and bias vectors, respectively. The above three gates enable the LSTM model to learn and retain essential information over long sequences.

Bi-directional LSTM (BiLSTM)

The BiLSTM improves upon the LSTM's capacity to capture contextual information by concurrently processing input sequences in both forward and backward directions, thus enhancing its ability to understand context. This bidirectional processing enables the model to consider not only the past information but also the future context of each word or token in the input sequence. The architecture of BiLSTM comprises two LSTM layers: one processes the input sequence from the beginning to the end, while the other processes it in reverse. This bidirectional nature enables the network to

comprehend long-range dependencies and semantic structures more effectively. The final representation of each word is obtained by concatenating the hidden states from both LSTM layers. This representation is then forwarded to the subsequent layers for classification. Given an input sequence $X = \{x_1, x_2, \dots, x_T\}$, the forward LSTM computations for each time step t are as follows: Firstly, the input to the LSTM cell at time step t is computed using the equation (14)

$$a_t = W_a \times [h_{t-1}, x_t] + b_a \quad (14)$$

where, W_a and b_a indicates the weight matrix, and bias vector respectively. Secondly, the forget gate, input gate, and candidate cell state are computed using the above equations (11-13). Next, candidate cell and update cell state are calculated using the following equations (15-16). After the calculation of candidate cells, the hidden state is computed using equation (17).

$$g_t = \text{tanh}(W_g * a_t + b_a) \quad (15)$$

$$c_t = f_t \times c_{t-1} + i_t \times g_t \quad (16)$$

$$h_t = o_t \times \text{tanh}(c_t) \quad (17)$$

Furthermore, for reverse LSTM, similar equations but with different weights and biases were used i.e., $X = \{X_T, x_{T-1}, \dots, x_1\}$. Finally, BiLSTM hidden state at time step t is obtained by concatenating the hidden states from both the forward and reverse LSTMs using equation (18).

$$h_{t_{final}} = [h_t, h_{t_{reverse}}] \quad (18)$$

Bidirectional Encoder Representations from Transformers (BERT)

BERT captures contextualized word representations in a bidirectional manner, considering both the left and right contexts of each word. It incorporates the transformer architecture and uses self-attention mechanisms to evaluate the significance of individual words in the input sequence while creating their respective representations. This process helps generate highly informative word embeddings that effectively capture the semantics and relationships within the text. After encoding, BERT employs the [CLS] token, which represents the entire collective information of the input sequence, to perform classification tasks.

The architecture of BERT involves self-attention, embedding transformations, and classification layers.

Given an input sequence of length L with hidden

representations denoted by the matrix H of size $(L \times d)$, where d is the dimension of the word embeddings. For each of the three matrices (Q , K , and V), BERT uses learnable weight matrices that are initialized randomly and then updated during the training process calculated using equations (19-21).

$$Q = H \times W_q \quad (19)$$

$$K = H \times W_k \quad (20)$$

$$V = H \times W_v \quad (21)$$

These weight matrices are denoted as W_q , W_k , and W_v respectively. These scores are further passed through a softmax function to obtain the attention weights A matrix of size $L \times L$ computed using equation (22).

$$A = \text{softmax} \frac{Q \times K^T}{\sqrt{d}} \quad (22)$$

The final step in the self-attention mechanism involves calculating the output matrix O of dimensions $L \times d$ by performing a weighted sum of the Value V matrix using the attention weights A . This computation is accomplished using equation (23).

$$O = A \times V \quad (23)$$

Subsequently, self-attention BERT architecture involves a feed-forward layer that is composed of a couple of fully connected layers to each word's representation in the sequence, followed by an activation function like ReLU. This adds more non-linearity to the model and helps capture complex patterns. This study also involves the application of two further variations of the original BERT: Distillation of BERT (DistilBERT) and robustly optimized BERT pretraining approach (x). DistilBERT distill knowledge from the larger BERT model into a smaller one without significantly sacrificing its performance. It achieves this by knowledge distillation, where the knowledge from a larger model is transferred to a smaller model during training. However, RoBERTa is designed to address some of the limitations of BERT and improve its performance through modifications in its pretraining approach. RoBERTa, in contrast to BERT, uses a larger batch size and longer training duration. It also removes the next sentence prediction (NSP) task that BERT uses during pretraining. This modification allows RoBERTa to have more training data and, therefore, leads to better performance.

IV. DATASET DESCRIPTION

In this research study, we incorporated the AI Open news headlines dataset for empirical analysis of proposed models. The utilized data is a pivotal contribution to the field of fake headline detection, thoughtfully curated to overcome the limitations prevalent in existing datasets. The comprehensive collection of data from The Onion, a renowned source of sarcastic versions of current events. To enhance our research, authentic and non-sarcastic news headlines were collected from the HuffPost, a prestigious American online news media company, using their news archive page. This unique dataset offers numerous advantages over conventional sarcasm datasets. The news headlines are crafted by professionals in a formal manner, ensuring they are free from spelling errors and informal language commonly seen in social media-based datasets. The detail of the dataset is given in Table I.

Table I. Detailed Description of Fake Headlines Dataset

Detail	Value
No. of classes	2
Total headlines	26709
No. of real headlines	14985
No. of fake headlines	11724
Average word length of headlines	5.38
Max word length of headlines	13.33
Min word length of headlines	2.33

As a result, the dataset exhibits reduced vocabulary sparsity and enhanced performance with the use of pretrained embeddings. Secondly, The Onion's exclusive dedication to producing sarcastic news ensures a wealth of high-quality labels in substantial quantities, effectively controlling label accuracy and dataset scalability. Furthermore, the self-contained nature of the dataset mitigates issues arising from sarcastic posts referencing external content, thereby enabling more precise isolation and identification of authentic sarcastic elements within the corpus.

V. PERFORMANCE EVALUATION MEASURES

To validate the performance of the proposed ML-, DL-, and transformer-based models, we employed four widely used evaluation measures including accuracy, precision, recall, and F1 score. The details of each measure are as follows:

Accuracy (ACC)

Accuracy evaluates the overall correctness of a model's predictions by calculating the ratio of correctly predicted instances (including true positives and true negatives) to the total number of instances. Accuracy can be calculated

using equation (24).

$$ACC = \frac{TP + TN}{Total\ Instances} \quad (24)$$

Where, TP represents the true positives, indicating the number of news headlines correctly predicted as fake headlines by the model, while TN indicates the true negatives, representing the number of headlines correctly predicted as real headlines.

Precision (PRC)

Precision assesses the ratio of true positive predictions to all positive predictions generated by the model, thereby indicating the model's capacity to minimize false positives. It is computed using equation (25).

$$PRC = \frac{TP}{TP + FP} \quad (25)$$

Where, FP represents false positives, which is the portion of the headlines that are incorrectly predicted as fake headlines but are real headlines.

Recall (RC)

Recall, alternatively referred to as sensitivity or true positive rate, quantifies the ratio of correct positive predictions among all the actual positive instances. It signifies the effectiveness of the model in capturing positive instances accurately. It is defined as:

$$RC = \frac{TP}{TP + FN} \quad (26)$$

Where, FN indicates false negative, which represents the incorrectly predicted fake headlines misclassified as real.

F1 Score

The F1 score represents the harmonic mean of precision and recall, offering a well-balanced evaluation of a model's performance by considering both these metrics. The F1 score is defined as:

$$F1 = 2 \times \frac{PRC \times RC}{PRC + RC} \quad (27)$$

Where, PRC and RC indicate the computed precision and recall, respectively.

VI. RESULTS AND DISCUSSION

In this study, we performed experiments by varying the diverse encoding layers after preprocessing and the results were computed using the AI Open news headlines dataset as the benchmark for fake headline classification.

A Results for Machine Learning & Ensemble Models

Table II presents the performance of diverse ML models using two different text encoding techniques: TF-IDF and sentence transformer encoding. Firstly, when analyzing the impact of encoding, it is evident that sentence transformer encoding consistently outperforms TF-IDF across all models. Sentence transformers generate sentence embeddings that capture more semantic information, enabling the models to better comprehend the underlying patterns in the text data. This richer representation is particularly beneficial in complex NLP tasks, resulting in higher accuracy, precision, recall, and F1 scores. Naïve Bayes, for example, achieved an accuracy of 78.42% with TF-IDF and 83.14% with sentence transformers. Similarly, DTs improve accuracy from 69.97% to 84.16%, showcasing the significant impact of the encoding technique.

According to Figure 3, the best performing model overall is AdaBoost when coupled with sentence transformer encoding. This combination achieves an impressive accuracy of 86.41% along with high precision, recall, and F1 scores. AdaBoost, as an ensemble learning method, effectively combines weaker models (DTs in this case) to create a strong learner, which, in conjunction with sentence transformers, results in better predictive capabilities. However, it is essential to consider the computational resources required for AdaBoost and the complexity of the task at hand, as simpler models like naïve Bayes with sentence transformers can offer competitive performance at a lower computational cost.

Table II. Performance Comparison of Diverse ML Model Over Fake Headline Dataset using TF-IDF and Sentence Transformer Encoding

Model	Encoding Layer	ACC	PRC	RC	F1
Naïve Bayes	TF-IDF	78.42	77.15	79.51	78.64
SVM		71.53	69.58	71.52	70.58
Decision Tree		69.97	70.05	68.50	69.48
Random Forest		73.64	73.80	70.00	71.86
AdaBoost		75.05	74.16	71.63	74.37
Naïve Bayes	Sentence Transformers	83.14	84.10	82.02	83.40
SVM		81.28	80.15	79.50	80.64
Decision Tree		84.16	80.58	83.50	82.10
Random Forest		80.69	81.50	78.58	79.50
AdaBoost		86.41	84.59	83.20	84.59



Figure 3. Accuracy Comparison of Diverse ML Models using TF-IDF and Sentence Transformer Encoding

B Results for Deep Learning & Transformer-Based Models

Table III. presents the results obtained from diverse DL models using different encoding techniques such as TF-IDF, GloVe, fastText, and sentence transformers. The models encoded with TF-IDF yielded a reasonably good

performance. The LSTM model achieved an accuracy of 85.65% and a balanced F1 score of 84.74%, indicating a fair trade-off between precision and recall. The BiLSTM variant performed slightly lower with an accuracy of 83.48% and an F1score of 83.39%. BERT, on the other hand, showed promising results with an accuracy of 86.52% and an F1 score of 84.47%. The DistilBERT model achieved an accuracy of 82.74%, though its F1

score was relatively lower at 81.17%. Among the TF-IDF encoded models, RoBERTa demonstrated the best

performance, achieving an accuracy of 88.82% and an F1 score of 86.07%.

Table III. Performance Comparison of DL and Transformer-based Models Over Fake Headline Dataset using TF-IDF, GloVe, fastText, and Sentence Transformer Encoding

Model	Encoding Layer	ACC	PRC	RC	F1
LSTM	TF-IDF	85.65	86.51	84.39	84.74
BiLSTM		83.48	82.28	81.74	83.39
BERT		86.52	83.96	85.39	84.47
DistilBERT		82.74	82.47	80.14	81.17
RoBERTa		88.82	86.36	85.85	86.07
LSTM	GloVe Embeddings	93.18	94.45	92.47	93.74
BiLSTM		91.96	90.74	89.69	90.31
BERT		94.74	90.93	93.00	92.36
DistilBERT		90.28	91.17	88.09	89.63
RoBERTa		95.48	94.39	93.74	94.29
LSTM	fastText Embeddings	91.69	92.25	90.53	91.72
BiLSTM		89.17	88.42	87.45	88.31
BERT		92.63	88.13	91.56	90.26
DistilBERT		88.48	89.23	86.23	87.37
RoBERTa		94.52	92.36	91.52	92.43
LSTM	Sentence Transformers	93.74	94.81	93.33	94.07
BiLSTM		92.17	90.78	89.68	91.13
BERT		95.01	90.96	93.89	92.72
DistilBERT		90.99	92.11	88.97	90.33
RoBERTa		96.17	95.32	94.04	95.42

The use of GloVe embeddings significantly improved the performance of the models. The LSTM model achieved an impressive accuracy of 93.18% and an F1 score of 93.74%, showcasing the effectiveness of this encoding technique. The BiLSTM variant also showed a considerable improvement with an accuracy of 91.96% and an F1 score of 90.31%. BERT achieved remarkable results with GloVe embeddings, obtaining an accuracy of 94.74% and an F1 score of 92.36%. Similarly, DistilBERT achieved an accuracy of 90.28% and an F1

score of 89.63%. Among the GloVe encoded models, RoBERTa stood out with the highest accuracy of 96.87% and an impressive F1 score of 94.29%. fastText embeddings also proved to be effective in enhancing model performance. The LSTM model achieved an accuracy of 91.69% and an F1 score of 91.72%. The BiLSTM variant obtained an accuracy of 89.17% and an F1 score of 88.31%.

BERT continued to perform well with fastText embeddings, achieving an accuracy of 92.63% and an F1

score of 90.26%. DistilBERT achieved an accuracy of 88.48% and an F1 score of 87.37%. RoBERTa maintained its excellence, achieving an accuracy of 94.52% and an F1 score of 92.43%. Among all the encoding techniques, sentence transformers demonstrated the most significant performance boost. The LSTM model achieved an accuracy of 93.74% and an F1 score of 94.07%. The BiLSTM variant obtained an accuracy of 92.17% and an F1 score of 91.13%. BERT continued to show impressive results with sentence transformer encoding, achieving an accuracy of 95.01% and an F1 score of 92.72%. DistilBERT achieved an accuracy of 90.99% and an F1 score of 90.33%. RoBERTa remained the top performer, achieving an accuracy of 96.17% and

an F1 score of 95.42%.

VII. CONCLUSION

This research study presents an investigation into the detection of fake headlines using text classification. The empirical analysis involves various ML, DL, and transformer-based models, and two different encoding techniques using the AI Open news headlines dataset as the benchmark. Results show that sentence transformer encoding consistently outperformed TF-IDF across all ML models.

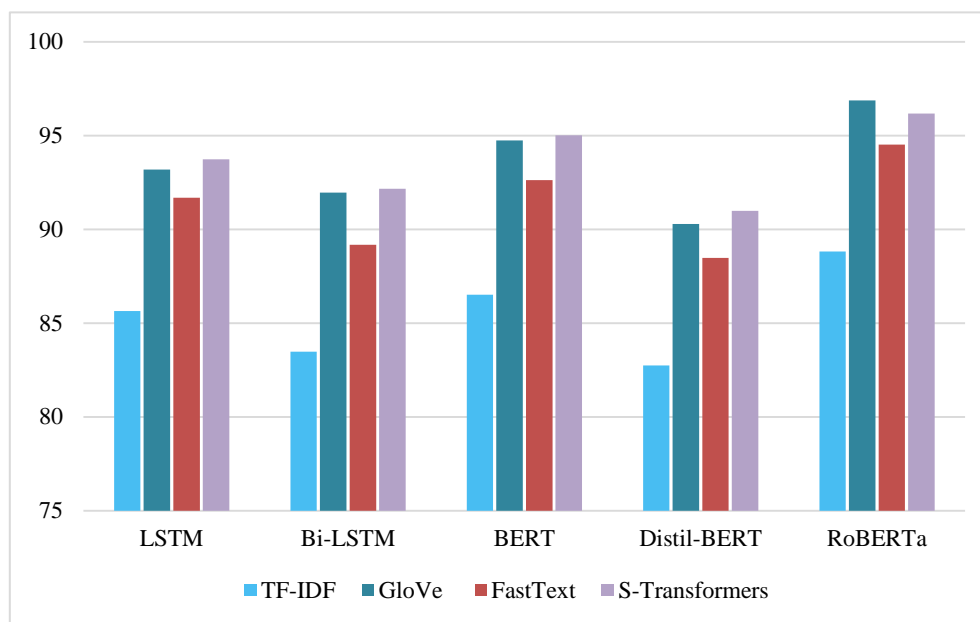


Figure 4. Accuracy Comparison of Various DL and Transformer-based Models using TF-IDF, GloVe, fastText and Sentence Transformer Encoding

The richer semantic information captured by sentence embeddings enables the models to better comprehend the underlying patterns in the text data, resulting in higher accuracy, precision, recall, and F1 scores. For instance, naïve Bayes achieved an accuracy of 78.42% with TF-IDF, which significantly improved to 83.14% with sentence transformers. Similarly, DTs saw a substantial improvement in accuracy from 69.97% to 84.16% with the use of sentence transformer encoding. The results of various DL models using different encoding techniques show that models using GloVe, fastText, and sentence transformers consistently outperformed those using TF-IDF. RoBERTa demonstrated superior performance across all encoding methods, achieving the highest accuracy of 96.87% with GloVe embeddings and 96.17% with sentence transformers. This study highlights the importance of choosing appropriate text encoding techniques when building text classification models. Future work on detecting fake headlines using text classification could explore advanced encoding

techniques like contextual embeddings, investigate ensemble methods for model combination, and focus on real-time monitoring and cross-lingual detection to enhance model performance and applicability.

VIII. REFERENCES

- [1] O. Stitini, S. Kaloun, and O. Bencharef, "Towards the Detection of Fake News on Social Networks Contributing to the Improvement of Trust and Transparency in Recommendation Systems: Trends and Challenges," *Inf.*, vol. 13, no. 3, p. 128, 2022, doi: 10.3390/info13030128.
- [2] M. Luo, J. T. Hancock, and D. M. Markowitz, "Credibility Perceptions and Detection Accuracy of Fake News Headlines on Social Media: Effects of Truth-Bias and Endorsement Cues," *Communic. Res.*, vol. 49, no. 2, pp. 171–195, 2022, doi: 10.1177/0093650220921321.
- [3] E. Shushkevich, M. Alexandrov, and J. Cardiff, "BERT-based Classifiers for Fake News Detection on Short and Long Texts with Noisy Data: A Comparative Analysis," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, 2022, pp. 263–274. doi: 10.1007/978-3-031-16270-1_22.
- [4] S. Sheikhi, "An effective fake news detection

- method using WOA-xgbTree algorithm and content-based features,” *Appl. Soft Comput.*, vol. 109, p. 107559, 2021, doi: 10.1016/j.asoc.2021.107559.
- [5] 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, 2021, doi: 10.1016/j.asoc.2020.106983.
- [6] M. Amjad, G. Sidorov, A. Zhila, H. Gómez-Adorno, I. Voronkov, and A. Gelbukh, “‘Bend the truth’: Benchmark dataset for fake news detection in Urdu language and its evaluation,” *J. Intell. Fuzzy Syst.*, vol. 39, no. 2, pp. 2457–2469, 2020, doi: 10.3233/JIFS-179905.
- [7] F. Fifita, J. Smith, M. B. Hanzsek-Brill, X. Li, and M. Zhou, “Machine Learning-Based Identifications of COVID-19 Fake News Using Biomedical Information Extraction,” *Big Data Cogn. Comput.*, vol. 7, no. 1, p. 46, 2023, doi: 10.3390/bdcc7010046.
- [8] X. Wang, P. Zhao, and X. Chen, “Fake news and misinformation detection on headlines of COVID-19 using deep learning algorithms,” *Int. J. Data Sci.*, vol. 5, no. 4, p. 316, 2020, doi: 10.1504/ijds.2020.115873.
- [9] T. Felber, “Constraint 2021: Machine Learning Models for COVID-19 Fake News Detection Shared Task,” *arXiv Prepr. arXiv2101.03717*, 2021, [Online]. Available: <http://arxiv.org/abs/2101.03717>
- [10] B. Wang, Y. Feng, X. cai Xiong, Y. heng Wang, and B. hua Qiang, “Multi-modal transformer using two-level visual features for fake news detection,” *Appl. Intell.*, vol. 53, no. 9, pp. 10429–10443, 2023, doi: 10.1007/s10489-022-04055-5.
- [11] L. Ying, H. Yu, J. Wang, Y. Ji, and S. Qian, “Multi-Level Multi-Modal Cross-Attention Network for Fake News Detection,” *IEEE Access*, vol. 9, pp. 132363–132373, 2021, doi: 10.1109/ACCESS.2021.3114093.
- [12] N. Capuano, G. Fenza, V. Loia, and F. D. Nota, “Content-Based Fake News Detection With Machine and Deep Learning: a Systematic Review,” *Neurocomputing*, vol. 530, pp. 91–103, 2023, doi: 10.1016/j.neucom.2023.02.005.
- [13] H. Jwa, D. Oh, K. Park, J. M. Kang, and H. Lim, “exBAKE: Automatic fake news detection model based on Bidirectional Encoder Representations from Transformers (BERT),” *Appl. Sci.*, vol. 9, no. 19, p. 4062, 2019, doi: 10.3390/app9194062.
- [14] A. Kumar, S. Saumya, and J. P. Singh, “NITP-AI-NLP@UrduFake-FIRE2020: Multi-layer dense neural network for fake news detection in urdu news articles,” in *CEUR Workshop Proceedings*, 2020, pp. 458–463.
- [15] H. Ahmed, I. Traore, and S. Saad, “Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, 2017, pp. 127–138. doi: 10.1007/978-3-319-69155-8_9.
- [16] A. Jain, A. Shakya, H. Khatter, and A. K. Gupta, “A smart System for Fake News Detection Using Machine Learning,” in *IEEE International Conference on Issues and Challenges in Intelligent Computing Techniques, ICICT 2019, IEEE*, 2019, pp. 1–4. doi: 10.1109/ICICT46931.2019.8977659.
- [17] A. P. S. Bali, M. Fernandes, S. Choubey, and M. Goel, “Comparative Performance of Machine Learning Algorithms for Fake News Detection,” in *Communications in Computer and Information Science*, Springer, 2019, pp. 420–430. doi: 10.1007/978-981-13-9942-8_40.
- [18] R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha, “FNDNet – A deep convolutional neural network for fake news detection,” *Cogn. Syst. Res.*, vol. 61, pp. 32–44, 2020, doi: 10.1016/j.cogsys.2019.12.005.
- [19] A. Wani, I. Joshi, S. Khandve, V. Wagh, and R. Joshi, “Evaluating Deep Learning Approaches for Covid19 Fake News Detection,” in *Communications in Computer and Information Science*, Springer, 2021, pp. 153–163. doi: 10.1007/978-3-030-73696-5_15.
- [20] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B. W. On, “Fake news stance detection using deep learning architecture (CNN-LSTM),” *IEEE Access*, vol. 8, pp. 156695–156706, 2020, doi: 10.1109/ACCESS.2020.3019735.
- [21] R. Sepúlveda-Torres, M. Vicente, E. Saquete, E. Lloret, and M. Palomar, “Exploring Summarization to Enhance Headline Stance Detection,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, 2021, pp. 243–254. doi: 10.1007/978-3-030-80599-9_22.
- [22] 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, 2021, doi: 10.1007/s11042-020-10183-2.
- [23] A. Hande, K. Puranik, R. Priyadarshini, S. Thavareesan, and B. R. Chakravarthi, “Evaluating Pretrained Transformer-based Models for COVID-19 Fake News Detection,” in *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021, IEEE*, 2021, pp. 766–772. doi: 10.1109/ICCMC51019.2021.9418446.
- [24] V. I. Ilie, C. O. Truica, E. S. Apostol, and A. Paschke, “Context-Aware Misinformation Detection: A Benchmark of Deep Learning Architectures Using Word Embeddings,” *IEEE Access*, vol. 9, pp. 162122–162146, 2021, doi: 10.1109/ACCESS.2021.3132502.
- [25] M. Szczepański, M. Pawlicki, R. Kozik, and M. Choras, “New explainability method for BERT-based model in fake news detection,” *Sci. Rep.*, vol. 11, no. 1, p. 23705, 2021, doi: 10.1038/s41598-021-03100-6.
- [26] M. A. Bsoul, A. Qusef, and S. Abu-Soud, “Building an Optimal Dataset for Arabic Fake News Detection,” *Procedia Comput. Sci.*, vol. 201, no. C, pp. 665–672, 2022, doi: 10.1016/j.procs.2022.03.088.
- [27] 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, p. 6970, 2022, doi: 10.3390/s22186970.
- [28] B. Palani, S. Elango, and K. Vignesh Viswanathan, “CB-Fake: A multimodal deep learning framework for automatic fake news detection using capsule neural network and BERT,” *Multimed. Tools Appl.*, vol. 81, no. 4, pp. 5587–5620, 2022, doi: 10.1007/s11042-021-11782-3.
- [29] I. K. Sastrawan, I. P. A. Bayupati, and D. M. S. Arsa, “Detection of fake news using deep learning CNN-RNN based methods,” *ICT Express*, vol. 8, no. 3, pp. 396–408, 2022, doi: 10.1016/j.icte.2021.10.003.
- [30] M. Fayaz, A. Khan, M. Bilal, and S. U. Khan, “Machine learning for fake news classification with optimal feature selection,” *Soft Comput.*, vol. 26, no. 16, pp. 7763–7771, 2022, doi: 10.1007/s00500-022-06773-x.
- [31] L. Huang, “Deep Learning for Fake News Detection: Theories and Models,” *ACM Int. Conf. Proceeding Ser.*, pp. 1322–1326, 2022, doi: 10.1145/3573428.3573663.
- [32] N. Rai, D. Kumar, N. Kaushik, C. Raj, and A. Ali, “Fake News Classification using transformer based enhanced LSTM and BERT,” *Int. J. Cogn. Comput. Eng.*, vol. 3, pp. 98–105, 2022, doi: 10.1016/j.ijcce.2022.03.003.
- [33] C. O. Truică, E. S. Apostol, and A. Paschke, “Awakened at CheckThat! 2022: Fake News Detection using BiLSTM and Sentence Transformer,” *CEUR Workshop Proc.*, vol. 3180, pp. 749–757, 2022.
- [34] C. O. Truică and E. S. Apostol, “MisRoBÆRTa: Transformers versus Misinformation,” *Mathematics*, vol. 10, no. 4, p. 569, 2022, doi: 10.3390/math10040569.
- [35] J. Yin, M. Gao, K. Shu, Z. Zhao, Y. Huang, and J. Wang, “Emulating Reader Behaviors for Fake News Detection,” *arXiv Prepr. arXiv2306.15231*, 2023, [Online]. Available: <http://arxiv.org/abs/2306.15231>
- [36] M. I. Nadeem et al., “SSM: Stylometric and semantic similarity oriented multimodal fake news detection,”

J. King Saud Univ. - Comput. Inf. Sci., vol. 35, no. 5, p. 101559, 2023, doi: 10.1016/j.jksuci.2023.101559.

[37] A. Agarwal, S. Mishra, and S. Ahmad, "Fake News Detection Using Machine Learning," in Lecture Notes in Electrical Engineering, IEEE, 2023, pp. 51–59. doi: 10.1007/978-981-99-5358-5_4.

[38] C. O. Truică and E. S. Apostol, "It's All in the Embedding! Fake News Detection Using Document Embeddings," Mathematics, vol. 11, no. 3, p. 508, 2023, doi: 10.3390/math11030508.

[39] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2014,

pp. 1532–1543. doi: 10.3115/v1/d14-1162.

Mohammed Alghobiri is an associated prof. in the department of Informatics at King Khalid university in Saudi Arabia. He is experienced and highly involved in information systems development and Implementation, especially experimental and participative approaches. Part of his interest is Management and Evaluation of Systems Development, including Software Process Improvement Methods and ERP D&I. Data Mining, Decision Support Systems and Electronic Government Concepts are also within his concern. He has very good experience in team leading and IT centres management.

maalghobiri@kku.edu.sa

0000-0002-6414-739X