Iraqi Journal for Computer Science and Mathematics

Volume 5 | Issue 4 Article 2

2024

# Automated Fake News Detection System

Saja A. Al-obaidi
*Department of Computer Engineering, Gazi University, Ankara, Turky AND Department of Scientific Affairs, Iraqi University, Baghdad, Iraq*, derasat20192020@gmail.com

Tuba Çağlıkantar
*Department of Computer Engineering, Gazi University, Ankara, Turky*

IRAQI JOURNAL FOR COMPUTER SCIENCE AND MATHEMATICS 2024;5:12–26



RESEARCH ARTICLE

# Automated Fake News Detection System

**Saja A. Al-obaidi [a,b,*], Tuba Çağlıkantar [a]**

[a] Department of Computer Engineering, Gazi University, Ankara, Turkey
[b] Department of Scientific Affairs, Iraqi University, Baghdad, Iraq

**ABSTRACT**

Online news has been the majority of people's information source in recent decades. However, a lot of the information that is accessible online is fake and sometimes even designed to mislead. It might be difficult for individuals to distinguish between certain false newspaper items and the real ones since they are so similar. Deep learning (DL) and machine learning (ML) models, among other automated false news detection (FND) techniques, are quickly becoming essential. A comparative study was conducted to analyze the performance of five prominent deep learning models across four distinct datasets, namely ISOT, FakeNewsNet, Dataset1, and Dataset2. Results indicated that while LSTM achieved the highest accuracy on the ISOT dataset (99.95%), CNN-GRU stood out with an exceptional 99.97% accuracy on the FakeNewsNet dataset. Both CNN-LSTM and LSTM exhibited almost perfect accuracies on Dataset1. On Dataset2, LSTM led with 98.64%. However, LSTM AutoEncoder consistently demonstrated lower efficiency, with accuracies spanning between 49% and 62%. The study underscores the critical role of dataset-specific model selection in optimizing deep learning outcomes.

**Keywords:** Fake news, Fake news detection, ISOT dataset, FakeNewsNet, Machine learning, Deep learning

## 1. Introduction

An astonishing 59.4% of the world's population, or 4.76 billion people, were predicted to be social media users worldwide as of January 2023, according to a digital global report [1]. With 137 million new users joining the platform in the previous year, social media continues its steady growth, at an average of more than 4 new users each second, or a 3% yearly growth rate. With more than 90% of internet users utilizing social media monthly, the digital landscape has made information globally accessible [2]. However, this digital democratization comes with significant drawbacks, chief among them being the propagation of false news [3]. Defining fake news as material that can deceive readers into believing its truthfulness is essential in addressing this problem [2]. Fake news, spread through various channels such as social media sites, emails, websites, and radio service, is a malicious form of propaganda aiming to influence public opinion [4]. Though false and often simple to verify, fake news is designed to mislead. Its rapid online spread can lead to societal consequences like decreased public trust in the news ecosystem, reputational harm, increased fear and skepticism, and potential societal instability [4]. The internet's lax regulation amplifies these challenges, allowing false information to sway opinions and disrupt the ecosystem's authenticity balance. Fake News Detection (FND) encompasses three subcategories: Fabrication, Hoax, and Satire. Fabrication relies on sensationalism, Hoax uses complex deception, while Satire intends humor but may be misconstrued as fact. The need for an automated system to accurately distinguish between false and truthful news is thus apparent. Despite previous efforts, further exploration is required to curb the spread of rumors and false information and enhance the evaluation of news source reliability. The digital era has witnessed an unprecedented surge in the accessibility and consumption of online information. With this rise, however, comes the alarming growth of fake news – a phenomenon

Received 26 August 2023; accepted 10 May 2024.
Available online 25 November 2024

* Corresponding author.
E-mail address: derasat20192020@gmail.com (S. A. Al-obaidi).

https://doi.org/10.52866/2788-7421.1200

that not only undermines journalistic integrity but also poses a significant threat to the very fabric of society [5]. Fake news, distinctly characterized by its intent to deceive, often serves as a vessel for propagating false beliefs, manipulating public opinion, and even instigating real-world consequences [6]. This is further exacerbated when one considers the overlap and occasional interchangeability of terms like "fake news" and "misinformation" [7]. For clarity, "fake news" typically refers to completely false information presented as news, while "misinformation" is a broader term, encompassing any false or misleading information, regardless of intent [8]. The proliferation of such false narratives, driven largely by social media, can lead to a myriad of societal issues, ranging from mere misinformation to large-scale public panic or even political upheavals. Previous attempts at combating this menace, although commendable, have often been limited in their scope, either focusing solely on binary classifications or lacking the nuance required to address the subtle shades of untruths. There remains a palpable gap in devising a comprehensive solution that not only identifies fake news but also discerns the varying degrees of its authenticity [9]. Our research seeks to bridge this gap by introducing a multi-dimensional approach to fake news detection, thus underscoring the urgency of this issue and its profound implications on society.

In this context, the current study introduces a groundbreaking approach to FND. By departing from traditional binary classification of 'Real' and 'Fake,' we propose a multi-layered approach that incrementally introduces new classes such as 'Half-True' and 'Barely True.' This nuanced classification, implemented through eight Machine Learning (ML) and three Deep Learning (DL) algorithms, provides a more precise detection of various shades of misinformation. Utilizing the public ISOT dataset, our method adds depth to the understanding of fake news and offers a sophisticated tool for combating its spread.

In light of the ever-evolving digital landscape and the challenges it poses, our research primarily aims to set itself apart by introducing a layered approach to Fake News Detection, moving beyond the rudimentary binary classifications predominantly observed in prior studies. Our primary contributions include the conception of an incremental multiclassification system which embraces categories like 'HalfTrue' and 'Barely True' – providing a holistic perspective to the subtleties of misinformation. This approach is supported by a comprehensive combination of eight Machine Learning algorithms and three advanced Deep Learning models, finetuned for accuracy. Leveraging the public ISOT dataset, we've ensured that our methodologies are adaptable and can be validated against real-world data. Additionally, our proposed framework outshines its predecessors by offering a nuanced understanding of misinformation shades, an element often overlooked in other solutions. While we acknowledge the robustness of our solution, we're also aware of its potential limitations, primarily concerning the adaptability of our models across varied datasets. As the abstract highlighted, the performance of models like LSTM AutoEncoder may not be consistent across different datasets. Going forward, our research endeavors will focus on refining these inconsistencies and expanding our model's versatility to cater to a broader spectrum of fake news categories. The rest of this study is structured as follows: Section 2 presents recent researches about FND published in the literature. Section 3 includes a detailed description of our proposed FND architecture, emphasizing the binary and multi-class approaches. Section 4 highlights the results, underscoring the effectiveness of our innovative contribution. Finally, the conclusion and future work are included in Section 5.

### *1.1. Literature review*

A collection of ensemble ML models for classifying news articles is cited in this research [10]. This research examines a variety of textual traits that can be utilized to differentiate between real and fake content. They then evaluate the performance of those algorithms using four real-world datasets. These properties are used to train a range of ML algorithms using a number of ensemble methodologies. Results show that the suggested ensemble learner technique performs better than individual learner approaches. Using the ISOT dataset, the random forest classifier achieves the greatest accuracy of 98%. In a research [11], lexical, sentiment, unigram, and bigram approaches with term frequency-inverse document frequency (TF-IDF) and Glove and character embedding were used to examine a number of ML and DL algorithms including: SVM, LR, DT, LSTM, convolutional HAN, and character level CLSTM). Results showed that the LSTM had the best outcome of 94% accuracy was produced by NB with bigram TF-IDF on the combined corpus dataset. In addition, in this work [12] a FND system was proposed using ML techniques. The TF-IDF of bag of words and n-grams is used as a feature extraction approach during the pre-processing stage. The SVM classifier is trained using a proposed dataset. Another study extracted features using the TF-IDF method [13]. Three ML models are implemented including SVM, Naïve Bayes (NB), and Passive Aggressive Classifier. The SVM classifier achieve the best accuracy of 95.05%. An ensemble-based-DL model for FND was put out in this study utilizing the LIAR dataset [3]. The dataset being natural led to the deployment of two DL models. A dense model was utilised for all the features except the "statement" features, they employed a bi-LSTM dense model. Studies revealed that the suggested approach has an accuracy of 89 percent when using just the statement attribute. Detecting fake news in social media is the purpose of this research [14]. They used the SVM and RF classifiers with and without 10-cross validation method. The two classifiers achieved an accuracy of 83.5% after using the cross validation technique. Using datasets from LIAR and PolitiFact, a research combines DL, NLP, and semantics to develop a hybrid methodology [15]. The study assessed the performance of DL models including Bi-LSTM GRU, and CapsNet as well as traditional ML models like MNB, SGD, LR, DT, and SVM. CapsNet prformed the best on the LIAR dataset with an accuracy of 64%. According to the study, including semantic data such NER feelings helped the classification model perform better. The researchers in this work have provided an ensemble classification model for accurately recognizing bogus news. The DT, RF, and Extra Tree Classifier ensemble of three well-known ML models is used by the model to extract key characteristics from datasets of false news and classify them. The Liar dataset shows that testing accuracy was 44.15 percent while training accuracy was 99.8 percent. Training and testing accuracy were both 100 percent on the ISOT dataset [16]. The authors of this study assessed the performance of ML and three DL models on two datasets of false and true news with varying sizes. Hold-out cross validation was used for the evaluation. The authors represented the text data using word frequency, TD-IDF, and embedding methods for DL models. Testing accuracy for the novel stacking model developed by the authors on the ISOT and KDnugget datasets was 99.94% and 96.05%, respectively.

## 2. Methodology

As show Fig. 1, the proposed architecture for FND includes the collection of the ISOT dataset, followed by data preprocessing and the proposed classifiers implementation. The models are then trained and tested using the rafined dataset. Finally, the data is evaluated and compared to determine the effectiveness of the models.

### 2.1. Datasets

The detection and categorization of fake news necessitate a robust dataset that can capture the multifaceted nature of truthfulness in news reporting. Our approach, therefore, incorporates four distinctive datasets, enabling a more intricate analysis.

The datasets chosen for this study have been meticulously selected to encompass a comprehensive range of news dynamics, ensuring a robust and exhaustive exploration of fake news detection. The ISOT corpus was an ideal starting point due to its dichotomous nature, separating news into clear 'True' and

'Fake' categories. Its extensive database of articles from both reputable and less reliable sources provided a vast landscape of real-world news, making it a suitable foundational dataset. On the other hand, the FakeNewsNet dataset offered a unique dimension to our analysis by encapsulating the propagation dynamics of fake news through social media, evidenced by its inclusion of tweet IDs. Recognizing that fake news is not just an issue of binary truth or falsehood, but exists on a spectrum, it became imperative to incorporate the LIARPLUS dataset. LIAR-PLUS, with its nuanced categories such as "Half-True" and "Barely True", enabled the introduction of gradient truth classifications, effectively addressing the subtleties and gray areas of misinformation. Additionally, the evidence sentences in LIAR-PLUS provided an extra layer of verification, reinforcing the model's ability to ascertain the truthfulness of a claim. In essence, the amalgamation of these datasets aimed to cater to the complexities of fake news across different platforms and degrees of truthfulness, ensuring our model's efficacy across varied scenarios.

1) *The ISOT Corpus:* For binary classification, we utilize the ISOT corpus, fully sourced from real-world sources [17]. This dataset consists of both true and false news articles, drawing from reputable sources like Reuters.com and less reliable sites recognized by fact-checkers like Politifact and Wikipedia. The articles cover various subjects, emphasizing politics and global events, and are divided into two files, "True.csv" and "Fake.csv," with more than 12,600 articles each. The "False.csv" file aggregates articles from several fake news sites, while the "True.csv" file strictly consists of stories from Reuters.com. These articles, primarily from 2016 to 2017, include details such as title, text, type, and publication date, and any existing errors or punctuation in the fake news articles have been retained.
2) *FakeNewsNet Dataset:* The FakeNewsNet dataset is a unique compilation that serves researchers with data collected based on ground truths from two distinctive sources, namely Politifact and Gossipcop [18–20]. With an aggregate shape of

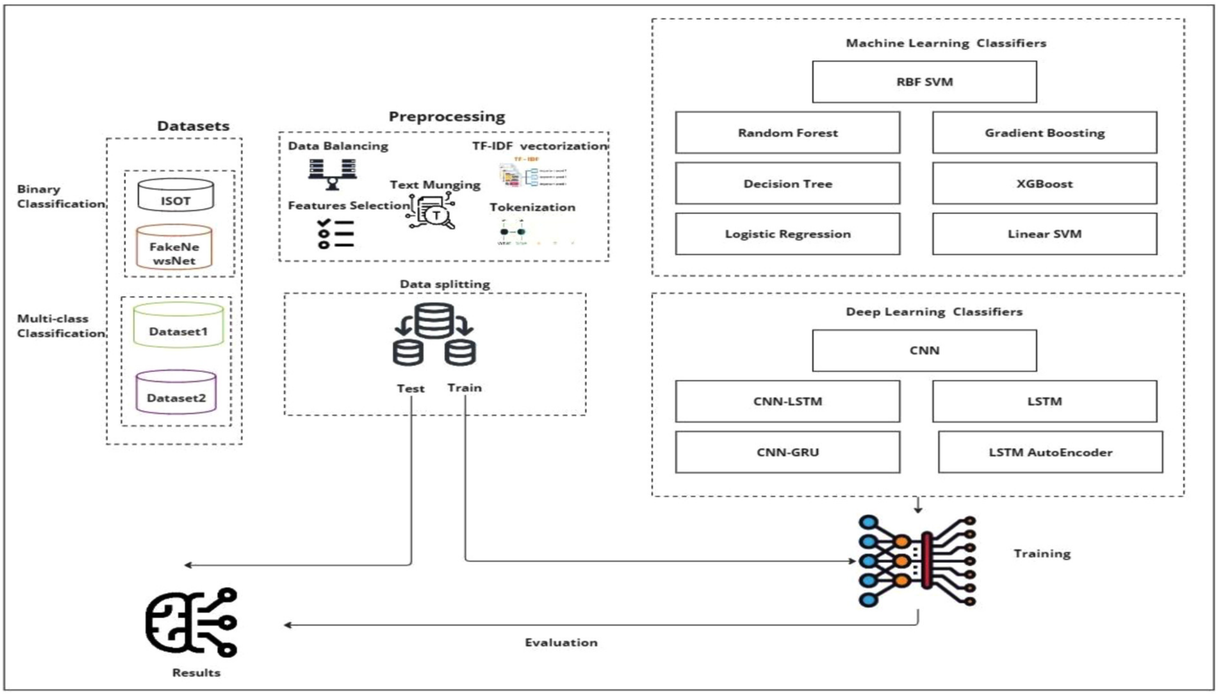


Fig. 1. FND proposed framework.

23,196 rows and 5 columns, it's structured to offer a comprehensive exploration of fake news dynamics. The dataset offers a structured insight into fake news, organized around five key columns:

- id: A unique identifier for each entry, facilitating precise data referencing and operations.
- news url: Direct URLs to the original news sources, ensuring that researchers can trace back the news' primary origin for context and verification.
- title: The headline or title of the news article, capturing the essence of the content and serving as the initial point of engagement for readers.
- tweet ids: A list of Twitter ID references that correspond to tweets disseminating the news article. This is particularly valuable for tracing the social media impact and propagation dynamics of the news article.
- label: A binary classification indicating the veracity of the news article, divided into 'Fake' and 'Real'. This is the linchpin for supervised learning tasks or comparative analyses in the domain of fake news detection.

3) *The LIAR-PLUS Dataset:* To deepen our understanding and classification of fake news, we further extend our approach by integrating classes from the LIAR-PLUS dataset, released in the paper [21]. This extension includes evidence sentences extracted automatically from full-text verdict reports by Politifact journalists. The LIAR-PLUS dataset is structured into 15 columns, including the ID of the statement, label, statement, subjects, speaker's details, state information, party affiliation, credit history counts (including "barely true," "false," "half true," "mostly true," and "pants on fire"), context, and the extracted justification. The inclusion of these evidence sentences aims to improve the benchmark for evidence retrieval, demonstrating that the addition of evidence information invariably enhances the performance of any fake news detection method.

4) *Combining the datasets:* Our research contributes by thoughtfully combining the ISOT corpus with the LIARPLUS dataset to create a nuanced and multidimensional approach to fake news detection. This combination facilitates a step-by-step progression from binary classification to the inclusion of intermediate classes like "Half-True" and "Barely True". Specifically, dataset1 combines the ISOT corpus with the "Half-True" feature from the LIAR-PLUS dataset. Dataset2 builds upon this by further integrating the "Barely True" feature, essentially combining the ISOT corpus, the "Half-True" feature, and the "Barely True" feature from the LIAR-PLUS dataset, creating a progressive extension from dataset1. By merging the specificities of

both datasets in this structured manner, we cater to the complexities of fake news, allowing our model to discern not only outright falsehoods but also shades of misinformation. This innovative method underscores the importance of considering the gradations of truth and falsehood in the era of digital information and contributes a sophisticated tool in the fight against the propagation of fake news.

This version captures the progression you described in your datasets, highlighting the incremental approach that adds layers of complexity to your classification. It should reflect the novel contribution of your research accurately.

### 2.2. Pre-processing

Data preprocessing is an indispensable phase in the machine learning pipeline, ensuring that models receive high-quality input for generating reliable predictions. Given the diverse origins and nature of our datasets, meticulous preprocessing became essential to harmonize discrepancies and elevate input data quality. Here's a detailed breakdown of our approach:

1) Feature Selection and Aggregation: Owing to the rich features present in our datasets, we deemed it crucial to focus on the most pertinent ones. We singled out the 'title' and 'text' attributes, as they encompassed the crux of the news articles. By amalgamating these features into one column, we fashioned a coherent text body, streamlining subsequent preprocessing steps.
2) Balancing the Dataset: Within the FakeNewsNet dataset, there was a discernible imbalance between 'real' and 'fake' news entries. To ensure a balanced model training, we undertook downsampling of the 'real' news entries, ensuring parity with the 'fake' news count.
3) Text Cleaning using Regular Expressions: Initiating the cleaning process, it was pivotal to cleanse the data of extraneous and potentially confounding elements. Utilizing regular expressions, we efficiently eradicated HTML tags, ubiquitous in web-scraped data. Additionally, superfluous punctuations, which might introduce noise, were purged. A uniform text landscape was achieved by converting all text to lowercase, ensuring words like 'The' and 'the' weren't distinctively treated.
4) Tokenization: Post cleaning, tokenization was our next endeavor. This entailed fragmenting the cleaned text into discrete words or tokens. This step's significance is monumental, as tokenized text facilitates algorithms to discern patterns and meanings. Each token is treated as a discrete entity, ripe for model analysis.
5) Term Frequency-Inverse Document Frequency (TFIDF): While tokenization parsed the text, not all tokens (or words) are of equal relevance. Certain ubiquitous words might not offer unique insights. The TF-IDF technique was thus employed to assign weight to each token based on its relevance [16]. Essentially, a word recurrent in a specific document but sparse across the corpus garners a higher score, underscoring its importance.
6) Data Splitting: Upon concluding preprocessing, it became imperative to segregate the data into training and testing subsets. This bifurcation ascertained that our model was trained on a substantial data portion (70%), while retaining an untapped segment (30%) for model validation, assessing overfitting, and gauging real-world efficacy.

Through this stringent preprocessing regimen, our models were equipped with data that was purified, balanced, tokenized, weighted for significance, and judiciously partitioned, laying the groundwork for potent and efficient machine learning.

### 2.3. Proposed classifiers

#### 2.3.1. Machine learning classifiers

In the study, seven ML models were used to detect false news from the ISOT dataset. The models were thoroughly discussed.

1- Decision Tree A Decision Tree (DT) is a supervised ML algorithm that identifies relationships between variables in a dataset to predict outcomes [22]. It uses a topdown tree-like structure where nodes are either class labels or decision points that determine the outcome. The algorithm is simple to understand and its decision-making process is transparent. However, it may not produce accurate results with small datasets due to being a weak learner. The crucial step in DT learning is choosing the most relevant attribute, which various DT algorithms approach differently, such as using Information Gain (ID3 algorithm) or Gain Ratio (C4.5 algorithm).The gain ratio and information gain may be determined using an attribute with $n$ distinct values and a training dataset $Gt$ as follows:

$$Gain(A; Gt) = \frac{Entropy\ (Gt) - \sum_{j=1}^{n} \frac{|Gt_j|}{|Gt|}\ Entropy\ (Gt_j)}{6} \tag{1}$$

$$GainRatio\ \ (I; Gt) = \frac{Gain\ (Gt; I)}{IV\ (I)} \tag{2}$$

where the intrinsic value of the characteristic $I$ may be calculated as follows:

$$\mathrm{IV}\ (\mathrm{I}) = -\sum_{\mathrm{j}=1}^{\mathrm{n}} \frac{|\mathrm{G_j}|}{|\mathrm{Gt}|} log_2 \frac{|Dt_j|}{|Gt|} \tag{3}$$

2- Random Forest is an ensemble approach that blends GT bagging with a random feature pick at each split [23]. Every tree in the forest makes a forecast, and the guess that receives the most votes wins out. Since no one method is always the most accurate due to the No Free Lunch principle, Random Forest outperforms individual classifiers in terms of accuracy and robustness. The final output is the mean prediction of all the trees, represented as

$$Y_f\,(X) = \frac{1}{T}\sum_{t=1}^{T} Y_t\,(X) \tag{4}$$

where $\mathrm{Y_t(X)}$ is the output of the individual tree t, and T is the number of trees in the forest.

3- Gradient Boosting is a ML technique used to build predictive models [24]. It works by combining multiple weak models to create a strong model that can make accurate predictions. This is achieved by iteratively adding models to the ensemble, each designed to correct the errors made by the previous models. The prediction of a Gradient Boosting model is the sum of the predictions made by each individual weak learner, and the choice of loss function and base learner can impact the model's performance. The hyperparameters of the Gradient Boosting model can also be tuned to optimize its performance. The process is mathematically represented in the additive model expression:

$$F\,(x) = \sum_{i=1}^{M} \alpha_i h_i\,(x) \tag{5}$$

where F(x) is the final predictive model, M is the number of trees, $\alpha_{\mathrm{i}}$ is the weight given to the i-th tree, $\mathrm{h_i(x)}$ is the prediction of the i-th tree

4- Logistic Regression is a commonly used ML classification algorithm for binary classification problems [25]. The aim is to predict the value of the predictive variable y, which can either be 0 or 1 representing the negative and positive classes, respectively. To classify these two classes, a hypothesis $h() = {}^{T} X$ is designed, and the threshold for the classifier's output is set to 0.5. If the value of $h() \geq 0.5$, it predicts y = 1 (real news), and if the value of $h() < 0.5$, it predicts y = 0 (fake news). The prediction in logistic regression is done under the condition that 0 ¡= h() ¡= 1. The logistic regression sigmoid function can be expressed as:

$$h\,(\theta) = \frac{1}{1 + \mathrm{e}^{-\theta^{\mathrm{T}}\,\mathrm{X}}} \tag{6}$$

And the cost function for logistic regression can be written as:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} cost\ (h(\theta(i)), y(i)) \tag{7}$$

where $m$ is the number of training samples, cost () is a function that calculates the error between the predicted and actual values.

5- XGBoost is a powerful and efficient ML algorithm that combines multiple weak models to form a strong model. It is well-suited for working with large datasets and is designed to be parallelizable for fast training. XGBoost includes many user-friendly features such as automatic handling of missing values, built-in cross-validation, and a comprehensive set of hyper-parameters for control of the learning process [26]. the objective function can be defined as:

$$\mathrm{Obj}\,(\Theta) = \sum_{i=1}^{n} l\,(y_i, \widehat{y_i}) + \sum_{i=1}^{t} \Omega\,(f_i) \tag{8}$$

where $\mathrm{Obj}(\Theta)$ is the objective function, $\mathrm{l(y_i, \widehat{y_i})}$ is the loss function, $\Omega(\mathrm{f_i})$ is the regularization term, $\mathrm{y_i}$ is the true label for instance I, $\widehat{\mathrm{y_i}}$ is the predicted label for instance I, $\mathrm{f_i}$ is the i-th tree, t is the number of trees.

6- Support Vector Machine A well-liked ML approach called the SVM may be used for both classification and regression [27]. SVM determines the maximum margin border, which is the ideal boundary between several classes. SVM may be constructed using either a linear or radial basis function kernel (RBF). While the RBF SVM uses a non-linear border to divide the classes, the linear SVM does it using a linear boundary. The SVM may differentiate between classes that are not linearly separable in the original space thanks to the boundary being generated based on a mathematical equation that translates the input data into a higher dimensional space. Mathematically, the SVM seeks to solve an optimization problem defined by the following primal formulation:

$$\begin{array}{ll} minimize & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ subject\ to & y_i(w \cdot x_i + b) \geq 1 - \xi_i,\ i = 1, \ldots, n \\ & \xi_i \geq 0, i = 1, \ldots, n \end{array} \tag{9}$$

where: $w$ is the weight vector perpendicular to the hyperplane, b is the bias term, C is the regularization parameter that controls the trade-off between maximizing the margin and minimizing the classification error, $x_i$ represents individual data points, $y_i$ is the label associated with $x_i$, $\xi_i$ are the slack variables that allow for soft-margin classification, facilitating the handling of non-linearly separable data and avoiding overfitting.

#### *2.3.2. Deep learning classifiers*

Convolutions neural networks (CNN) and Recurrent neural network (RNN) are two sub fields of DL. For DL models are proposed in our research CNN, CNN-GRU, LSTM, and CNN-LSTM.

*2.3.2.1. Convolution neural network.* The CNN model is a sequential model consisting of an embedding layer followed by dropout layers, three convolutional layers with different kernel sizes (3, 4, and 5), a global max pooling layer, dense layers, and dropout layers. The embedding layer maps each word in the input sequence to a 128-dimensional vector representation. Dropout layers help prevent overfitting, and the convolutional layers with 128 filters capture local patterns in the input sequence. The global max pooling layer reduces dimensionality, and the dense layers capture complex relationships. The final dense layer with the number of classes as units and sigmoid activation produces classification probabilities.

*2.3.2.2. CNN-LSTM.* The sequential model begins with an Embedding layer, which maps each word in the input sequence to a dense vector representation. The input dimension is determined by the length of the tokenizer's word index plus one, and the output dimension is set to the specified embedding size. The input length is defined as the maximum sequence length. Convolutional layers are then added to capture local patterns in the input sequence. The first Conv1D layer has 64 filters with a kernel size of 3 and uses the ReLU activation function. Max pooling with a pool size of 2 is applied to reduce the dimensionality of the features. The second Conv1D layer has 128 filters with a kernel size of 3 and also uses the ReLU activation function. Again, max pooling is applied. Next, an LSTM (Long Short-Term Memory) layer is added. The LSTM layer has 64 units and incorporates dropout with a rate of 0.2 to prevent overfitting. The LSTM layer is well-suited for capturing long-range dependencies and sequential patterns in the input sequence. A dense layer with 64 units and the ReLU activation function follows the LSTM layer to capture complex relationships in the data. Finally, a dense layer with 2 units and the sigmoid activation function is added for binary classification, with each unit representing a class (e.g., true or false for fake news detection).

*2.3.2.3. CNN-GRU.* Gated Recurrent Unit (GRU) is a powerful variation of the standard RNN that incorporates a combined gating mechanism, similar to LSTM, to address short-term memory limitations. Within the GRU architecture, a set of gates controls and regulates the flow of information, enabling the model to determine which information is crucial to retain or discard [28]. This gating mechanism facilitates effective learning and prediction by selectively passing on important information. In this research, we implement a hybrid model that combine the CNN and GRU models. This model combines convolutional layers for capturing local patterns and a GRU layer for capturing temporal dependencies. The model has 64 filters in the first convolutional layer and 128 filters in the second convolutional layer. Max pooling is applied after each convolutional layer with a pool size of 2. The GRU layer has 64 units and utilizes dropout with a rate of 0.2 to prevent overfitting. The model is flattened before passing through a dense layer with 64 units and ReLU activation. The final dense layer has 2 units, representing the number of classes, and utilizes the sigmoid activation function.

## 3. Optimization and evaluation metrics justification

For our deep learning models, optimization was a pivotal aspect to ensure enhanced performance and prediction accuracy. The optimization process was multifaceted and included various steps, as elaborated below:

### *3.1. Model optimization*

1) *Hyperparameter Tuning:* Using techniques such as grid search and random search, we systematically assessed various hyperparameters like learning rate, dropout rate, and batch size. This empirical approach aided in identifying the optimal hyperparameter set that would boost the model's performance.
2) *Regularization:* To counter overfitting, we incorporated dropout layers in the model. By randomly setting a fraction of input units to 0 during training, the model became less sensitive to specific weights, fostering more generalization.
3) *Learning Rate Scheduling:* Adapting the learning rate during training, either by reducing it when a plateau was detected or using a decay rate, we ensured that the model converged faster and more robustly.
4) *Early Stopping:* We constantly monitored the validation loss during training. If it ceased

to decrease (indicating potential overfitting), training was halted to prevent wastage of computational resources and further overfitting.

### 3.2. Justification for evaluation metrics

The choice of evaluation metrics was rooted in the specific nature of the problem – detecting false news. Given the possible severe implications of false negatives and false positives in this context, it was crucial to consider metrics beyond mere accuracy.

1) *Precision:* Given the dire repercussions of misclassifying real news as fake, a high precision was desired to ensure fewer false positives. It gauged the model's ability to correctly identify actual fake news from all flagged instances.
2) *Recall:* This metric ascertained the model's ability to identify all the fake news from the dataset. A high recall is crucial to minimize the number of fake news instances that slip through undetected.
3) *F1-Score:* Given the trade-offs between precision and recall, F1-score provided a harmonic mean of both, offering a holistic view of the model's performance.
4) *Area Under the ROC Curve (AUC-ROC):* Given the binary classification nature of the problem, AUC-ROC was a robust metric to determine the model's discriminative power between real and fake news, regardless of the threshold setting.

By considering these metrics, we ensured a comprehensive assessment of our models. The emphasis was on achieving a delicate balance between ensuring genuine articles aren't incorrectly flagged (precision) and capturing as many fake instances as possible (recall). This approach aimed to ensure that the proposed models are both reliable and efficient for real-world applications.

## 4. Results and discussion

### 4.1. Evaluation metrics

Several measures were used to evaluate the performance of the implemented classifiers. The equations of these metrics have been presented including ACC (11), PREC (10), REC (9), and F1 (12).

$$REC = \frac{TP}{FN + TP} \tag{10}$$

$$PREC = \frac{TP}{TP + FP} \tag{11}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$F1 = \frac{2 \times REC \times PREC}{REC + PREC} \tag{13}$$

- TP represents the instances when the model correctly identified a positive news article.
- TN indicates the instances when the model accurately classified a negative news article.
- FP shows the instances when the model misclassified a negative news article as positive.
- FN reflects the cases where the model incorrectly classified a positive news article as negative.

### 4.2. The training parameters

The deep models used in this analysis were trained with the following parameters. The loss function employed was categorical cross-entropy. The optimizer chosen was Adam, which is a popular optimization algorithm known for its efficiency and effectiveness in training deep neural networks. The training process was conducted over 5 epochs, with each epoch representing a complete pass through the training data. A batch size of 128 was utilized, indicating that the model parameters were updated after processing 128 samples at a time. The models were designed to classify data into two, three and four classes, and the k-fold cross-validation technique with a k-value of 5 was employed to assess their performance on multiple subsets of the data. This technique involves dividing the dataset into five equal-sized subsets or folds. Each fold is then treated as a separate validation set, while the remaining four folds are used for training the model. This process is repeated five times, with each fold serving as the validation set once. By doing so, we ensure that the model is trained and evaluated on different subsets of the data, providing a more robust assessment of its performance. For each fold, the model underwent a training process consisting of multiple epochs. In each epoch, the model's loss and accuracy were calculated based on the training data, while the validation loss and accuracy were computed using the validation data. This allowed us to monitor the model's performance and assess its ability to generalize to unseen data.

### 4.3. Binary classification using the ISOT dataset

*1) Machine Learning Classifiers Results:* We employed various ML models to classify fake and real news articles, as shown in Table 1, with the main criterion

Table 1. The machine learning classifiers results.

| Classifier | ACC | Class | PREC | REC | F-S |
|---|---|---|---|---|---|
| Random Forest | 83.66% | Fake | 88% | 40% | 55% |
| | | Real | 83% | 98% | 90% |
| Gradient Boosting | 82.24% | Fake | 100% | 29% | 45% |
| | | Real | 81% | 100% | 89% |
| XGBoost | 86.89% | Fake | 89% | 55% | 68% |
| | | Real | 87% | 98% | 92% |
| Linear SVM | 34.15% | Fake | 25% | 79% | 38% |
| | | Real | 73% | 19% | 30% |
| RBF SVM | 74.89% | Fake | 0% | 0% | 0% |
| | | Real | 75% | 100% | 86% |
| Logistic Regression | 74.82% | Fake | 29% | 0% | 0% |
| | | Real | 75% | 100% | 86% |
| Decision Tree | 99.73% | Fake | 100% | 100% | 100% |
| | | Real | 100% | 100% | 100% |

Table 2. The deep learning classifiers results.

| Classifier | ACC | Class | PREC | REC | F-S |
|---|---|---|---|---|---|
| CNN | 98.59% | Fake | 96% | 99% | 97% |
| | | Real | 100% | 99% | 99% |
| CNN-LSTM | 98.03% | Fake | 99% | 93% | 96% |
| | | Real | 98% | 100% | 99% |
| CNN-GRU | 97.99% | Fake | 98% | 94% | 96% |
| | | Real | 98% | 99% | 99% |
| LSTM | 99.95% | Fake | 100% | 100% | 100% |
| | | Real | 100% | 100% | 100% |
| LSTM AutoEncoder | 52% | Fake | 100% | 52% | 69% |
| | | Real | 0% | 0% | 0% |

for comparison being accuracy. Among the models, the top two performers were the Random Forest and XGBoost models. The Random Forest model achieved an accuracy of 83.66%, with a precision of 88% and recall of 40% for the fake class, and a precision of 83% and recall of 98% for the real class. On the other hand, the XGBoost model achieved an accuracy of 86.89%, with a precision of 89% and recall of 55% for the fake class, and a precision of 87% and recall of 0.98% for the real class. These results highlight the effectiveness of ensemble-based models, specifically Random Forest and XGBoost, in accurately distinguishing between fake and real news articles. The rest of models achieved varying levels of accuracy in classifying fake and real news articles. While Gradient Boosting performed relatively well with an accuracy of 82.24%, Logistic Regression, Decision Tree, Linear SVM, and RBF SVM showed lower accuracies ranging from 34.15% to 74.89%.

*2) Deep Learning Classifiers Results:* Table 2 presents the results of the deep models (CNN, CNN-LSTM, and CNNGRU) in classifying fake and real news articles. The models achieved high accuracy, with the CNN model achieving an accuracy of 98.59%. For the fake class, the CNN model achieved a precision of 96%, indicating that it correctly identified 96% of the fake articles. The recall (or sensitivity) for the fake class was 99%, meaning it correctly classified 99% of the actual fake articles. The F1-score, which considers both precision and recall, was 97%, indicating a good balance between the two metrics. The sensitivity and specificity for the fake class were 98.52% and 98.62%, respectively. Similar high performance was observed for the real class, with a precision of 100%, recall of 99%, and F1-score of 99%. The sensitivity and specificity for the real class were 98.62% and 98.52%, respectively. For the CNN-LSTM model, an accuracy of 98.03% was achieved. The precision, recall, and F1-score for the fake class were 99%, 93%, and 96%, respectively. The sensitivity and specificity for the fake class were 93.39% and 99.56%, respectively. For the real class, the precision, recall, and F1-score were 98%, 100%, and 99%, respectively. The sensitivity and specificity for the real class were 99.56% and 93.39%, respectively. The CNN-GRU model achieved an accuracy of 97.99%. The precision, recall, and F1-score for the fake class were 98%, 94%, and 96%, respectively. The sensitivity and specificity for the fake class were 93.45% and 99.45%, respectively. For the real class, the precision, recall, and F1-score were 98%, 99%, and 99%, respectively. The sensitivity and specificity for the real class were 99.94% and 93.57%, respectively.

Among the three deep models, the CNN model achieved the highest accuracy of 98.59% and exhibited excellent performance in classifying both fake and real news articles. With a precision of 96% and recall of 99% for the fake class, the CNN model demonstrated its ability to accurately identify fake articles while minimizing false negatives. Similarly, for the real class, it achieved a precision of 100% and recall of 99%, indicating its proficiency in correctly classifying real news articles.

### 4.4. Binary classification using the fake news net dataset

*1) Machine Learning Classifiers Results:* Table 3 illustrates the performance of several machine learning classifiers on the FakeNewsNet dataset. Among them, XGBoost outperforms the others with an accuracy of 99.81%, achieving perfect scores in precision, recall, and F-score for both Fake and Real classes. Following closely, both Linear SVM and RBF SVM achieve an accuracy of 99.67% and 99.58%, respectively, with perfect scores in most metrics, except for a slight reduction in precision and F-score for the Real class in RBF SVM. Gradient Boosting also exhibits strong performance, with an accuracy of 99.54% and high scores across all metrics, notably achieving perfect F-Scores for both classes. Conversely, Random Forest

**Table 3.** The machine learning classifiers results.

| Classifier | ACC | Class | PREC | REC | F-S |
|---|---|---|---|---|---|
| Random Forest | 99.14% | Fake | 99% | 99% | 99% |
| | | Real | 99% | 99% | 99% |
| Gradient Boosting | 99.54% | Fake | 100% | 99% | 100% |
| | | Real | 99% | 100% | 100% |
| XGBoost | 99.81% | Fake | 100% | 100% | 100% |
| | | Real | 100% | 100% | 100% |
| Linear SVM | 99.67% | Fake | 100% | 100% | 100% |
| | | Real | 100% | 100% | 100% |
| RBF SVM | 99.58% | Fake | 100% | 100% | 100% |
| | | Real | 99% | 100% | 99% |
| Logistic Regression | 99.04% | Fake | 99% | 99% | 99% |
| | | Real | 99% | 99% | 99% |
| Decision Tree | 77.84% | Fake | 56% | 57% | 56% |
| | | Real | 85% | 85% | 85% |

**Table 4.** The deep learning classifiers results.

| Classifier | ACC | Class | PREC | REC | F-S |
|---|---|---|---|---|---|
| CNN | 69.07% | Fake | 68% | 72% | 70% |
| | | Real | 71% | 67% | 69% |
| CNN-LSTM | 97.99% | Fake | 98% | 94% | 96% |
| | | Real | 98% | 99% | 99% |
| CNN-GRU | 99.97% | Fake | 100% | 100% | 100% |
| | | Real | 100% | 100% | 100% |
| LSTM | 93.74% | Fake | 100% | 100% | 100% |
| | | Real | 100% | 100% | 100% |
| LSTM AutoEncoder | 52% | Fake | 100% | 52% | 69% |
| | | Real | 0% | 0% | 0% |

and Logistic Regression, while still demonstrating high performance, present the lowest accuracies at 99.14% and 99.04%, respectively. They achieve a balanced but slightly lower performance across both classes.

*2) Deep Learning Classifiers Results:* Table 4 showcases the performance of various deep learning classifiers, presenting a wide range of results. CNN-GRU tops the list with an extraordinary accuracy of 99.97%, achieving perfect scores across precision, recall, and F-score for both the Fake and Real classes. CNN-LSTM also demonstrates strong performance, with an accuracy of 97.99%, and high scores across all metrics, particularly excelling in precision and recall for the Real class. In contrast, the CNN model shows a significantly lower accuracy of 69.07%, with a substantial gap in precision and recall between the Fake and Real classes. LSTM exhibits a notable result with an accuracy of 93.74%, achieving perfect scores across all metrics, but still falling behind the more advanced CNN-GRU model. The LSTM AutoEncoder represents a unique case, as it presents an accuracy of only 52%, with significant disparities between classes. While the model performs well for the Fake class in precision and recall, it fails completely for the Real class, showing 0% in all three metrics. This range of results illustrates the capabilities and limitations of various deep learning architectures in this specific context. The superiority of CNN-GRU underscores the effectiveness of combining convolutional layers with recurrent gates for this task. At the same time, the dramatic difference in LSTM AutoEncoder's performance between classes reveals potential challenges in model tuning or data representation.

### 4.5. Results using dataset1

*1) Machine Learning Classifiers Results:* Our investigation into the multiclass classification as shown in Table 5 of fake news detection utilizing dataset1, which consists of three classes - fake (0), true (Eq. (1)), and half-true (Eq. (2)), has yielded insightful results. Among the machine learning algorithms applied, Random Forest achieved an accuracy of 93.28%, with a precision, recall, and F1-score of 0.93, 0.94, and 0.93 for class 0, and slightly higher values for class 1 and class 2. The Gradient Boosting model exhibited a significant improvement with an accuracy of 98.97%, demonstrating near-perfect precision and recall for classes 0 and 1, while achieving a 0.93 precision and 0.97 recall for class 2. The Logistic Regression model achieved an even higher accuracy at 98.99%, with almost uniform precision, recall, and F1-score across all three classes. XGBoost emerged as the top performer with an impressive accuracy of 99.56%, reaching perfection in precision and recall for classes 0 and 1, and a 0.98 score for class 2. The Decision Tree (D-tree) model closely followed, with an accuracy of 99.27% and similar distribution in precision, recall, and F1-score. Finally, the SVM with RBF kernel and Linear SVM models had accuracies of 92.99% and 92.22%, respectively, with comparable performance across the three classes, although slightly lower than other models. The high accuracy, precision, and recall scores across most models, particularly XGBoost, suggest a successful implementation of the multiclass classification system. The results highlight the efficacy of complex ensemble methods like Gradient Boosting and XGBoost, demonstrating their superior capability in handling nuanced classifications such as discerning outright falsehoods, absolute truths, and the shades of misinformation in between. However, simpler models like Random Forest and SVM also performed commendably, showcasing the robustness and versatility of the approach.

*2) Deep Learning Classifiers Results:* Our exploration into deep learning techniques for the multiclass classification of fake news detection (consisting of classes: fake (0), true (Eq. (1)), and half-true (Eq. (2))) revealed remarkable findings. Among the models analyzed, LSTM demonstrated extraordinary proficiency

**Table 5.** The machine learning classifiers results using dataset1.

| Classifier | ACC | Class | PREC | REC | F-S |
|---|---|---|---|---|---|
| Random Forest | 93.28% | Fake | 93% | 94% | 93% |
| | | Real | 94% | 93% | 93% |
| | | half-true | 95% | 88% | 91% |
| Gradient Boosting | 98.97% | Fake | 99% | 98% | 99% |
| | | Real | 99% | 100% | 100% |
| | | half-true | 93% | 97% | 95% |
| XGBoost | 99.56% | Fake | 100% | 99% | 100% |
| | | Real | 100% | 100% | 100% |
| | | half-true | 98% | 98% | 98% |
| Linear SVM | 92.22% | Fake | 92% | 93% | 92% |
| | | Real | 92% | 92% | 92% |
| | | half-true | 96% | 89% | 92% |
| RBF SVM | 92.99% | Fake | 92% | 94% | 93% |
| | | Real | 94% | 92% | 93% |
| | | half-true | 94% | 86% | 90% |
| Logistic Regression | 98.99% | Fake | 99% | 99% | 99% |
| | | Real | 99% | 99% | 99% |
| | | half-true | 99% | 95% | 97% |
| Decision Tree | 99.27% | Fake | 99% | 99% | 99% |
| | | Real | 100% | 100% | 100% |
| | | half-true | 97% | 96% | 96% |

**Table 6.** The deep learning classifiers results using dataset1.

| Classifier | ACC | Class | PREC | REC | F-S |
|---|---|---|---|---|---|
| CNN | 59.34% | Fake | 45% | 62% | 52% |
| | | Real | 72% | 56% | 63% |
| | | half-true | 86% | 73% | 79% |
| CNN-LSTM | 99.90% | Fake | 100% | 100% | 100% |
| | | Real | 100% | 100% | 100% |
| | | half-true | 99% | 99% | 99% |
| CNN-GRU | 99.96% | Fake | 100% | 100% | 100% |
| | | Real | 100% | 100% | 100% |
| | | half-true | 100% | 99% | 100% |
| LSTM | 99.90% | Fake | 100% | 100% | 100% |
| | | Real | 100% | 100% | 100% |
| | | half-true | 100% | 100% | 100% |
| LSTM AutoEncoder | 49% | Fake | 100% | 49% | 66% |
| | | Real | 0% | 0% | 0% |
| | | half-true | 0% | 0% | 0% |

with an accuracy of 99.90%, achieving perfect precision, recall, and F1-score across all three classes. The CNNLSTM model also yielded an accuracy of 99.90%, with perfection in precision, recall, and F1-score for classes 0 and 1, and near-perfect scores for class 2. The CNN-GRU model topped the performance with an outstanding accuracy of 99.96%, reaching a perfect 1.00 in nearly all metrics, save for a 0.99 recall for class 2. In stark contrast, the standalone CNN model lagged behind significantly, with an accuracy of only 59.34%, exhibiting a marked discrepancy in performance across classes; the precision ranged from 0.45 to 0.86, and the recall varied from 0.56 to 0.73.

### 4.6. Results using dataset2

*1) Machine Learning Classifiers Results:* In our comprehensive evaluation using dataset2, as shown in Tables 6 and 7, the introduction of the "barely-true" class added a nuanced layer to the fake news detection task, generating varied performances across the models. Random Forest reported an accuracy of 88.25%, displaying strong results for classes 0 and 1 but a considerable decline in precision and recall for the newly introduced classes 2 and 3. Gradient Boosting marked a significant increase in accuracy to 94.89%, with near-perfect scores for the first two classes, while the results for the half-true and barely-true categories remained less optimal. Similarly, Logistic Regression achieved an accuracy of 94.91%, with a slight improvement in precision and recall for classes 2 and 3. The XGBoost model further advanced the accuracy to 95.19%, maintaining outstanding performance in classes 0 and 1, with moderate results in the new categories. The Decision Tree model closely followed with an accuracy of 94.87%. RBF SVM and Linear SVM reported accuracies of 89.26% and 91.08% respectively, again showing strong results for classes 0 and 1, with a more varied performance for classes 2 and 3. The consistent excellence in identifying outright fake and

**Table 7.** The machine learning classifiers results.

| Classifier | ACC | Class | PREC | REC | F-S |
|---|---|---|---|---|---|
| Random Forest | 88.25% | Fake | 90% | 94% | 92% |
| | | Real | 94% | 91% | 92% |
| | | half-true | 54% | 59% | 56% |
| | | barely-true | 45% | 30% | 36% |
| Gradient Boosting | 94.89% | Fake | 100% | 98% | 99% |
| | | Real | 99% | 100% | 100% |
| | | half-true | 55% | 76% | 64% |
| | | barely-true | 49% | 29% | 36% |
| XGBoost | 95.19% | Fake | 99% | 99% | 99% |
| | | Real | 100% | 100% | 100% |
| | | half-true | 57% | 64% | 60% |
| | | barely-true | 48% | 42% | 45% |
| Linear SVM | 91.08% | Fake | 93% | 96% | 95% |
| | | Real | 96% | 95% | 96% |
| | | half-true | 55% | 65% | 59% |
| | | barely-true | 45% | 26% | 33% |
| RBF SVM | 89.26% | Fake | 91% | 95% | 93% |
| | | Real | 94% | 92% | 93% |
| | | half-true | 56% | 63% | 59% |
| | | barely-true | 47% | 30% | 36% |
| Logistic Regression | 94.91% | Fake | 99% | 99% | 99% |
| | | Real | 99% | 99% | 99% |
| | | half-true | 60% | 70% | 64% |
| | | barely-true | 52% | 37% | 43% |
| Decision Tree | 94.87% | Fake | 99% | 99% | 99% |
| | | Real | 100% | 100% | 100% |
| | | half-true | 55% | 60% | 57% |
| | | barely-true | 47% | 43% | 45% |

**Table 8.** The deep learning classifiers results using dataset2.

| Classifier | ACC | Class | PREC | REC | F-S |
|---|---|---|---|---|---|
| CNN | 61.88% | Fake | 64% | 65% | 64% |
| | | Real | 62% | 64% | 63% |
| | | half-true | 94% | 44% | 60% |
| | | barely-true | 0% | 100% | 0% |
| CNN-LSTM | 96.6% | Fake | 100% | 100% | 100% |
| | | Real | 100% | 100% | 100% |
| | | half-true | 97% | 59% | 73% |
| | | barely-true | 16% | 68% | 26% |
| CNN-GRU | 95.77% | Fake | 100% | 100% | 100% |
| | | Real | 100% | 100% | 100% |
| | | half-true | 100% | 56% | 71% |
| | | barely-true | 0.1% | 83% | 0.2% |
| LSTM | 98.64% | Fake | 100% | 100% | 100% |
| | | Real | 100% | 100% | 100% |
| | | half-true | 93% | 83% | 88% |
| | | barely-true | 77% | 92% | 84% |
| LSTM AutoEncoder | 62% | Fake | 64% | 65% | 64% |
| | | Real | 62% | 64% | 63% |
| | | half-true | 94% | 44% | 60% |
| | | barely-true | 0% | 100% | 0% |

true news contrasts with the relative challenge in classifying the more nuanced categories of “half-true” and “barely-true.” These results emphasize the complexity of distinguishing finer shades of truth, where models like XGBoost and Gradient Boosting outperformed others, while also illustrating the limitations in capturing the subtleties of misinformation.

*2) Deep Learning Classifiers Results:* The investigation of deep learning algorithms on dataset2, as shown in Table 8, unveiled unique insights into fake news detection in a multiclass setting. The LSTM model demonstrated strong accuracy at 98.64%, achieving perfection in classes 0 and 1, but a noticeable drop in precision and recall for classes 2 and 3. Similarly, the CNN-LSTM model reached an accuracy of 96.06%, with flawless scores for the fake and true classes but a significant decline in the halftrue category, and an unexpected result in the barely-true class. The CNN model’s performance was distinctly modest with an accuracy of 61.88%, reflecting more balanced yet lower precision and recall across the four classes, and an anomalous result for the barely-true class with only one support instance. Lastly, the CNN-GRU model reported an accuracy of 95.77%, again showcasing impeccable results for classes 0 and 1, with a sharp reduction in performance for the newly introduced classes. The disparate outcomes in these models indicate the varying capabilities of deep learning techniques in discerning between fake, true, half-true, and barely-true news. While the LSTM and CNN-GRU models showed a more consistent and robust performance, the CNN-LSTM and CNN models seemed to struggle with the more nuanced classes, especially the barely-true category.

**Table 9.** Accuracy of machine learning classifiers across datasets.

| Classifier | ISOT | FakeNewsNet | Dataset1 | Dataset2 |
|---|---|---|---|---|
| Random Forest | 83.66% | 99.14% | 93.28% | 88.25% |
| Gradient Boosting | 82.24% | 99.54% | 98.97% | 94.89% |
| XGBoost | 86.89% | 99.81% | 99.56% | 95.19% |
| Linear SVM | 34.15% | 99.67% | 92.22% | 91.08% |
| RBF SVM | 74.89% | 99.58% | 92.99% | 89.26% |
| Logistic Regression | 74.82% | 99.04% | 98.99% | 91.75% |
| Decision Tree | 99.73% | 77.84% | 99.27% | 89.73% |

### *4.7. Comparing results*

As illustrated in Table 9 For the ISOT dataset, the Random Forest classifier achieved an accuracy of 83.66%. When tested on the FakeNewsNet dataset, its performance jumped remarkably to 99.14%. On Dataset1, the accuracy was 93.28%, and for Dataset2, it was 88.25%. Gradient Boosting delivered an accuracy of 82.24% on the ISOT dataset. Its performance was exceptional on the FakeNewsNet dataset with an accuracy of 99.54%. On Dataset1 and Dataset2, the accuracies were 98.97% and 94.89%, respectively. This model stood out particularly on the FakeNewsNet dataset, achieving an outstanding accuracy of 99.81%. On the ISOT dataset, it scored 86.89%. For Dataset1, the accuracy was 99.56% and for Dataset2, it was 95.19%. Linear SVM had a notable drop in performance on the ISOT dataset, where it managed only 34.15% accuracy. However, its accuracy on the FakeNewsNet dataset was 99.67%. On Dataset1, it scored 92.22% and on Dataset2, the accuracy was 91.08%. For the ISOT dataset, the RBF SVM achieved 74.89% accuracy. On the FakeNewsNet dataset, its accuracy was 99.58%. The model achieved 92.99% accuracy on Dataset1 and 89.26% on Dataset2. On the ISOT dataset, Logistic Regression recorded an accuracy of 74.82%. It performed relatively well on the FakeNewsNet dataset with an accuracy of 99.04%. On Dataset1, its accuracy was nearly perfect at 98.99%, while for Dataset2, it scored 91.75%. The Decision Tree classifier showcased a stellar performance on the ISOT dataset with an almost perfect accuracy of 99.73%. However, its performance slightly dipped on the FakeNewsNet dataset, securing an accuracy of 77.84%. For Dataset1, the accuracy was 99.27%, and on Dataset2, it was 89.73%. Most of the classifiers demonstrated exceptionally high accuracy rates on the FakeNewsNet and Dataset1, with XGBoost emerging as the most consistent in performance across all datasets. Linear SVM’s performance was notably variable, achieving high accuracy on some datasets while lagging significantly on others, particularly the ISOT dataset. Decision Tree’s almost perfect accuracy on the ISOT dataset was another standout result.

**Table 10.** Accuracy of deep learning models on different datasets.

| Model | ISOT | FakeNewsNet | Dataset1 | Dataset2 |
|---|---|---|---|---|
| CNN | 98.59% | 69.07% | 59.34% | 61.88% |
| CNN-LSTM | 98.03% | 97.99% | 99.90% | 96.6% |
| CNN-GRU | 97.99% | 99.97% | 99.96% | 95.77% |
| LSTM | 99.95% | 93.74% | 99.90% | 98.64% |
| LSTM AutoEncoder | 52% | 52% | 49% | 62% |

The Table 10 presents a comparative analysis of the performance of five deep learning models across four datasets. When evaluating the ISOT dataset, the LSTM model exhibited the highest accuracy at 99.95%, closely followed by the CNN with 98.59%. Interestingly, the FakeNewsNet dataset saw almost impeccable performance from the CNN-GRU model at 99.97%, and the CNN-LSTM wasn't far behind with 97.99%. For Dataset1, both the CNN-LSTM and LSTM achieved near-perfect scores, both registering an accuracy of 99.90%. Meanwhile, on Dataset2, the LSTM led the pack with an accuracy of 98.64%. A notable outlier in these results was the LSTM AutoEncoder, which consistently lagged behind its counterparts, with its performance ranging between 49% to 62% across the datasets. This comprehensive assessment reveals significant disparities in model efficiencies depending on the dataset used, emphasizing the importance of model selection based on the nature of the data at hand. Upon close observation of the data from Table 9, it is evident that there are substantial differences in the accuracy rates across different classifiers and datasets. One of the most surprising results is the performance of the Decision Tree classifier on the ISOT dataset with an astonishing accuracy of 99.73%. Given that Decision Trees tend to overfit when presented with a lot of features, it's likely that the ISOT dataset has distinct features that the Decision Tree could efficiently leverage, or it could also be a sign of overfitting. In contrast, its performance dipped significantly on the FakeNewsNet dataset. This suggests that the nature of data in the FakeNewsNet dataset is more complex, with possibly overlapping features, which usually challenges Decision Trees.

XGBoost, a gradient boosting algorithm, maintains consistently high accuracy across all datasets. Its efficiency is attributed to its ability to combine multiple decision trees and correct the errors of predecessor trees iteratively. Furthermore, the regularization parameters in XGBoost play a pivotal role in preventing overfitting, hence making it versatile across different datasets.

The Linear SVM has a drastic performance dip on the ISOT dataset. SVMs, especially the linear ones, rely on the data being linearly separable. The ISOT dataset might have a more convoluted decision boundary that a linear hyperplane cannot efficiently dissect.

In Table 10, Deep Learning models present intriguing results. The CNN, although often powerful, underperforms on the FakeNewsNet, Dataset1, and Dataset2. This suggests that the datasets might have long-range dependencies and sequential patterns that simple CNNs can't capture effectively. CNN-LSTM and CNN-GRU hybrids show robust performances, pointing to the significance of combining feature extraction capabilities of CNNs with the temporal pattern detection strengths of RNNs like LSTM and GRU. LSTM, with its memory cells, has an exceptional ability to remember past information which might be the reason for its consistent high accuracy. The LSTM AutoEncoder's mediocre performance suggests that the compression and decompression mechanism might be losing vital features necessary for accurate classification in these specific datasets.

In summary, the performance variations across different models accentuate the importance of understanding dataset intricacies and tailoring models that can best capture the underlying patterns in the data.

### *4.8. Comparative analysis of benchmark results*

The tabulated benchmarks in Table 11 offer a holistic perspective on the performance of various methodologies in the realm of fake news detection. [10] utilized an ensemble methodology, highlighting the strength of combining various base learners, and achieved an impressive 98% accuracy on the ISOT dataset with a Random Forest classifier. On another front, LSTM's prowess in handling sequential data is evident from [11] where it outperforms other models on a combined corpus. However, the innovative use of CapsNet in [15], which encapsulates hierarchical relationships in data, showcased a modest result, underscoring the challenges of the LIAR dataset or potential model-data misalignments. This study's novel stacking approach, combining both ML and DL techniques, has demonstrated its versatility with high accuracy rates on both the ISOT and KD-nugget datasets. In summary, while many methods

**Table 11.** Comparison of accuracy results from various studies.

| Reference | Best Model/Dataset | Achieved Accuracy |
|---|---|---|
| [10] | Random Forest/ISOT | 98% |
| [11] | LSTM/Combined Corpus | 94% |
| [13] | SVM | 95.05% |
| [3] | Bi-LSTM Dense/LIAR (Statement) | 89% |
| [14] | SVM, RF (with 10-fold CV) | 83.5% |
| [15] | CapsNet/LIAR | 64% |
| This Study | CNN-GRU/Dataset1 | 99.96% |

offer promising results, the choice of dataset and the inherent features play a pivotal role in determining the efficacy of a model, making it imperative to consider both the model and data intricacies when benchmarking and devising new strategies.

## 5. Conclusion and future work

Detecting and combatting fake news in today's digital age has ascended to paramount importance. The pervasive dissemination of misleading content not only manipulates individual perspectives but can also jeopardize societal harmony and undermine the very pillars of democracy. Consequently, the urgency to devise potent mechanisms to discern genuine news from fabrications is more pressing than ever. In our quest to address this burgeoning challenge, our research embarked on a comprehensive evaluation of an array of machine learning (ML) and deep learning (DL) models. Our methodology hinged on leveraging four distinct datasets, each tailored for binary and multi-class classification tasks. The diverse nature of these datasets allowed us to simulate various real-world scenarios, thereby enhancing the reliability and applicability of our findings. Among the algorithms we employed, seven hailed from traditional ML paradigms: Random Forest, XGBoost, Logistic Regression, Decision Tree, Linear SVM, and RBF SVM. Concurrently, we explored the capabilities of five advanced DL models, namely CNN, CNN-LSTM, CNN-GRU, LSTM, and LSTM autoencoder. Our meticulous approach ensured each model was rigorously trained and subsequently assessed on our chosen datasets. This holistic approach offered insights into the model's proficiency in differentiating between binary classifications of real and fake news, as well as multiclass classifications of fake, real, half-true, and in extended scenarios, barely-true news articles. What emerged from our investigation was a nuanced understanding of the strengths and limitations of each model in various contexts. Some models exhibited exemplary performance across all datasets, while others showcased niche specializations. It's imperative to highlight that no single model served as a panacea. Instead, the efficacy of a model was invariably contingent on the characteristics of the dataset and the specific classification task at hand. In conclusion, our research underscores the potential of both ML and DL in the fight against fake news. It suggests a multi-faceted approach, where a combination of models might be harnessed based on the nuances of the problem. As the digital realm continues to evolve, so will the sophistication of fake news strategies. Armed with the insights from our research, we are better poised to evolve our defense mechanisms in tandem, ensuring that truth and authenticity remain unassailable.

## Acknowledgement

Authors would like to thank Al_raqia for moral support.

## Funding

No funding received for this work.

## Conflicts of interest

The author declares no conflict of interest.

## References

1. D. Reportal, "Social media users," 2023. [Online]. Available: https://datareportal.com/social-media-users.
2. K. S. Adewole, T. Han, W. Wu, H. Song, and A. K. Sangaiah, "Twitter spam account detection based on clustering and classification methods," *The Journal of Supercomputing*, vol. 76, pp. 4802–4837, 2020.
3. N. Aslam, I. Ullah Khan, F. S. Alotaibi, L. A. Aldaej, and A. K. Aldubaikil, "Fake detect: A deep learning ensemble model for fake news detection," *complexity*, vol. 2021, pp. 1–8, 2021.
4. S. Mishra, P. Shukla, and R. Agarwal, "Analyzing machine learning enabled fake news detection techniques for diversified datasets," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–18, 2022.
5. N. A. Lafta, "A Comprehensive Analysis of Keras: Enhancing Deep Learning Applications in Network Engineering," *Babylonian Journal of Networking*, pp. 94–100, 2023. doi: https://doi.org/10.58496/BJN/2023/012.
6. M. R. Kondamudi, S. R. Sahoo, L. Chouhan, N. Yadav, "A comprehensive survey of fake news in social networks: Attributes, features, and detection approaches," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 6, p. 101571, 2023.
7. X. Zhang and A. A. Ghorbani, "An overview of online fake news: Characterization, detection, and discussion," *Information Processing & Management*, vol. 57, no. 2, p. 102025, 2020.
8. M. Burhanuddin, "Efficient Hardware Acceleration Techniques for Deep Learning on Edge Devices: A Comprehensive Performance Analysis", *KHWARIZMIA*, vol. 2023, pp. 1–10, 2023. doi: 10.70470/KHWARIZMIA/2023/010.
9. I. Ali, M. N. B. Ayub, P. Shivakumara, and N. F. B. M. Noor, "Fake news detection techniques on social media: A survey," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.
10. I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake news detection using machine learning ensemble methods," *Complexity*, vol. 2020, pp. 1–11, 2020.
11. J. Y. Khan, M. T. I. Khondaker, S. Afroz, G. Uddin, and A. Iqbal, "A benchmark study of machine learning models for online fake news detection," *Machine Learning with Applications*, vol. 4, p. 100032, 2021.
12. N. F. Baarir and A. Djeffal, "Fake news detection using machine learning," in *2020 2nd International Workshop on*

*Human-Centric Smart Environments for Health and Well-being (IHSH)*. IEEE, 2021, pp. 125–130.
13. J. Shaikh and R. Patil, "Fake news detection using machine learning," in *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*. IEEE, 2020, pp. 1–5.
14. W. E. Noori and A. S. Albahri, "Towards Trustworthy Myopia Detection: Integration Methodology of Deep Learning Approach, XAI Visualization, and User Interface System," *Applied Data Science and Analysis*, pp. 1–15, 2023. doi: https://doi.org/10.58496/ADSA/2023/001.
15. A. M. Braşoveanu and R. Andonie, "Integrating machine learning techniques in semantic fake news detection," *Neural Processing Letters*, vol. 53, no. 5, pp. 3055–3072, 2021.
16. T. Jiang, J. P. Li, A. U. Haq, A. Saboor, and A. Ali, "A novel stacking approach for accurate detection of fake news," *IEEE Access*, vol. 9, pp. 22626–22639, 2021.
17. F. A. Ozbay and B. Alatas, "Fake news detection within online social media using supervised artificial intelligence algorithms," *Physica A: statistical mechanics and its applications*, vol. 540, p. 123174, 2020.
18. K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media," *arXiv preprint arXiv:1809.01286*, 2018.
19. K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017.
20. A. A. Nafea, S. A. Alameri, R. R. Majeed, M. A. Khalaf, and M. M. AL-Ani, "A Short Review on Supervised Machine Learning and Deep Learning Techniques in Computer Vision," *Babylonian Journal of Machine Learning*, pp. 48–55, 2024. doi: https://doi.org/10.58496/BJML/2024/004.
21. T. Alhindi, S. Petridis, and S. Muresan, "Where is your evidence: Improving fact-checking by justification modeling," in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 2018, pp. 85–90.
22. X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu and S. Y. Philip et al., "Top 10 algorithms in data mining," *Knowledge and information systems*, vol. 14, no. 1, pp. 1–37, 2008.
23. T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE, 1995, pp. 278–282.
24. J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, pp. 1189–1232, 2001.
25. D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 20, no. 2, pp. 215–232, 1958.
26. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 785–794. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785.
27. C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
28. J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.