International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I6P125

Volume 74 Issue 6, 365-385, June 2026


*Original Article*

# Synthesizing Political Context Understanding with AI Techniques to Improve Fake News Detection Performance across Regions

Pundlik Dattatray Jadhav[1], Rajesh k Shukla[2]

[1,2]*Department of Computer Science and Engineering, Oriental University, Indore, (M.P.), India.*

[1]*Corresponding Author : pdjadhav17@gmail.com*

Received: 25 January 2026 Revised: 19 March 2026 Accepted:20 April 2026 Published: 27 June 2026

***Abstract** - Fake news is spreading quickly on the internet, which is very bad for society and the security of the government. The significant issue that was talked about in the paper was the creation of automatic systems that can detect fake news better and adapt to various areas. The dataset used in the study is the LIAR dataset, which is a standard set of various political statements labeled with varying degrees of truthfulness. Text is also cleaned up, tokenized, and represented with existing trained word embeddings such as GloVe and Word2Vec as a step in data preparation. To identify complex trends in the text, most language and contextual features are removed, such as syntactic, semantic, and sentiment-based ones. The primary contribution of this study is a way of grouping various features into one representation. A set of models is subjected to performance tests, and it includes Random Forest, Naive Bayes, Convolutional Neural Network (CNN), Autoencoder, and a proposed Hybrid CNN-Autoencoder architecture. The hybrid model performs the most, having the greatest precision and the most equalized classification scores. Comparative analysis demonstrates that the combination of deep learning and knowledge of the environment significantly enhances the level of detection in domains. It is a flexible AI-based system that can work in the context of language and political differences and is a big step forward in searching for fake information automatically.*

***Keywords** - Fake news detection, Political context analysis, Feature fusion, Deep Learning, CNN–autoencoder, LIAR dataset.*

## 1. Introduction

The fast spread of information through social media, online news sites, and political blogs has greatly changed how people understand and interact with political events in the modern digital era. While making it easier for anyone to share information has given people around the world more power, it has also made fake news more common, which is a very serious problem. Political content that is false or confusing can change people's minds, make societies more divided, and affect how democratic decisions are made [1]. This worrying trend has led scholars, lawmakers, and technologists to look for new ways to successfully find and combat misinformation. The old techniques of discovery, relying as far as possible on lexical or statistical hints, do not necessarily detect the small linguistic, geographical, and territorial variations, which are in evidence in political language. Due to this, the need to integrate the understanding of political context with the Artificial Intelligence (AI) techniques is becoming increasingly significant to simplify the process of identifying fake news in a vast number of socio-political environments and locations. Scholars of the mechanism of detecting fake news have observed it to begin with basic text classification models and then more advanced machine learning and deep learning architectures. Previously, researchers have largely focused on keyword-based extraction of features and artistic extraction of features. More recent methods have embedded semantics and mood analysis to determine what happens on the surface [2]. Despite these advances, most models are, nevertheless, unable to deal with the complexities of politics that are different in different regions, languages, and cultures. It is illustrated by the fact that what is true in one nation may have an alternative meaning or political impact in another nation [3]. Moreover, the datasets that were employed in the literature do not necessarily provide equal representation of regional settings, and therefore, detection systems are not as useful in different locations. The LIAR dataset, a famous standard that comprises the political remarks with the marks of the degree of truthfulness, is ideal to investigate these issues and enhance the contextual flexibility [4].

Although there has been a great deal of advancement in detecting fake news, the majority of the current solutions are more interested in textual semantics, lexical patterns, or social interaction characteristics, but not in the broader political and regional context of the news statements. Political communication is usually based on ideological stances, geographical scripts, and historical trustworthiness of communicators. Models that are purely based on textual clues thus find it difficult to encode such contextual clues, and hence cannot accurately classify political utterances with subtle framing or partial truths.

The other constraint to the current research is the ability of fake news detection systems to generalize to other political settings. Most of the available models are trained on datasets of a single country or political area, and thus, they are unable to adjust to various regional patterns of discourse. Additionally, other types of deep learning techniques concentrate on semantic embeddings only and do not use contextual metadata like credibility of the speaker, political party, or framing of the topic. These drawbacks mean that there is a need to have a context-sensitive detection model that can combine linguistic and political contextual information in order to enhance detection performance across regions.

The aim of the research will be to recommend a superior AI-based model that integrates the information on context, meaning, syntax, and tone to locate fake news more precisely across regions. The hypotheses are to fuse the various feature extraction through a feature fusion technique, apply a dimensionality reduction technique to achieve optimization, and experiment with simple machine learning models and with advanced deep learning models, e.g., a CNN model with an Autoencoder model, to obtain improved detection results.

This work is being done because we need to protect the purity of democracy and keep people's trust in information platforms right away. As false information keeps getting more complex, old models that only look at surface-level written clues are no longer enough. The goal of this study is to make a detection system that is smart, scalable, and can react to different political settings by combining knowledge of political context with AI-driven analysis. Finally, this study helps make information environments more stable by using the benefits of how AI and understanding political debate work together. This makes it possible for media landscapes to be more open and reliable. In order to resolve these shortcomings, this paper presents a context-based fake news detection system, which combines political contextual information with enhanced artificial intelligence methods. The proposed model uses syntactic, semantic, and sentiment-based characteristics in addition to the context-based political features gained in the dataset, unlike the traditional methods that use only the text (refer to Figure 3). The paper also proposes a hybrid CNN-Auto encoder, which is a combination of convolutional neural networks as a feature extractor of contextual features and autoencoders as a feature representation learning. The hybrid approach enables the model to get both local language tendencies and the latent contextual correlations in political utterances. The proposed framework will combine the methods of feature fusion and embedding in context to enhance the accuracy of classification and retain the strength of the framework, through various political narratives. The proposed approach offers a more comprehensive view of political discourse, in comparison to the models that have been previously reported, and allows for identifying misleading or partially true claims in politics better. In contrast to the current fake news detection methods that use mostly textual semantics or contextual embeddings of large language models, the framework suggested explicitly includes political contextual metadata (speaker credibility, party affiliation, historical truthfulness indicators, etc.). Through the integration of linguistic-related properties with organized political background and learning models based on the hybrid CNN-Autoencoder framework, the model enhances the contextual interpretation and discrimination rate concerning the discursive political contexts in various settings.

The Major contribution of the paper is:

1. To combine the approach that involves consideration of political background with AI-powered fake news detection.
2. To propose a multi-feature fusion model that is a combination of semantic, syntactic, and sentiment-based features to present text in a more holistic context.
3. Develops a deep learning model that enhances the classification and feature learning precision.
4. Utilizes the LIAR dataset in order to contrast the deep learning techniques with conventional machine learning models.

## 2. Related Work

With the swift expansion of digital media platforms, much research has been conducted on how to identify fake news. This study employs the linguistic, psychological, and computer methods of comprehending and combating lies. The primary focus of early study was the use of simple vocabulary and syntactic features such as n-grams, part-of-speech tags, and readability indices to perform text-based classification. These models were effective in detecting the errors in the language, but they were not effective in detecting the political and verbal details that influence the spread of the fake information [5]. Subsequently, Machine Learning (ML) algorithms such as Support Vector Machines (SVM), Naïve Bayes, and Random Forest have been used to identify fake text and style by analyzing styles and trends in handwritten text [6]. These models were, however, too limited as they were based on traits that were manually programmed and could not easily adapt to regional differences in politics. Deep learning approaches have revolutionized the industry by allowing models to acquire complex semantic relationships on text data only. The Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) architectures could identify the way words rely on one another sequentially; thus they were more effective at detecting fake news [7]. These models were effective at computing linear patterns, but found it difficult with the ideological and contextual phrasing of political words

[8]. The use of Convolutional Neural Networks (CNNs) in addition to text classification further enhanced feature extraction as crucial local trends in a news story are discovered [9]. The ability of CNNs to find short text clues was high, with worse results in finding deeper meaning links when the text did not have contextual embedding integration. To avoid this issue, embedding-based models such as Word2vec and Glove were developed to represent words on continuous fields of vectors whilst preserving their significant relations [10]. Such embeddings simplified a great deal the process of models comprehending the meaning of words in various contexts. Subsequently, transformer-based designs, including BERT and RoBERTa, gave contextualised embeddings, which comprehended the syntax and semantics of political text [11]. Although such models are highly precise, they usually require a significant amount of computing and significant amounts of training data, which makes this problem difficult to apply to systems that detect fake news in particular areas [12]. At the same time, other studies have looked at how people feel and their emotions as important signs of misleading political material. Studies have shown that fake news often uses strong feelings, polarised views, or an intentionally misleading tone to get people to respond strongly [13]. Adding linguistic features to mood analysis and emotion recognition makes detection systems easier to understand by catching the emotional side of false information [14]. Adding Named Entity Recognition (NER) and dependency processing also helps find the people, claims, and connections in political texts, which improves our understanding of the bigger picture [15].

**Table 1. Related work summary for fake news detection**

| Technique | Dataset | Features Considered | Key Findings / Accuracy | Limitations |
|---|---|---|---|---|
| Naïve Bayes, SVM [16] | LIAR | Lexical, syntactic | 70% accuracy; effective on small datasets | Poor contextual understanding |
| Random Forest [17] | BuzzFeed News | Stylistic, linguistic | Improved interpretability | Weak in semantic learning |
| LSTM, Bi-LSTM [18] | FakeNewsNet | Sequential semantic | Captures text dependencies | Struggles with long text context |
| CNN [19] | Kaggle Fake News | Word embeddings | High precision on short text | Misses deep semantic meaning |
| CNN-LSTM Hybrid [20] | LIAR | Word2Vec, GloVe | Improved accuracy over baseline | High computational cost |
| Word2Vec + SVM [21] | LIAR | Semantic embeddings | Balanced performance | Lacks sentiment integration |
| BERT Transformer [22] | PolitiFact | Contextual embeddings | 90%+ accuracy | Requires high resources |
| RoBERTa [23] | GossipCop | Semantic + contextual | State-of-the-art results | Limited regional adaptability |
| Sentiment-based SVM [24] | Custom political dataset | Sentiment polarity | Highlights emotional tone | Ignores linguistic structure |
| Multi-feature Fusion [25] | LIAR | Semantic + NER + POS | Fusion boosts classification | High feature redundancy |
| Autoencoder [26] | PoliticFact | Dimensional reduction | Improved latent representation | Needs hybrid integration |
| Hybrid CNN–Autoencoder [27] | LIAR | Semantic + Sentiment + Syntax | Highest accuracy; region-adaptive | Requires further multilingual testing |

Context-aware and multimodal detection of fake news has gained more and more attention in recent studies. Language models based on transformers (BERT, RoBERTa) have shown high performance in the contextual representation of the semantics of political speech. Besides, graph-based and multimodal learning models have also been suggested in order to combine text, social, and context information to enhance misinformation detection.

Nevertheless, most of these models continue to use textual embeddings to a large extent without making an explicit use of political metadata or contextual speaker hints that can offer useful clues to assess the legitimacy of political utterances. Subsequently, the need for a combination of linguistic representations with contextual political information to enhance classification performance and interpretability still exists.

## 3. LIAR Dataset Description

The LIAR data is a standard data set with which one performs research on fake news and fact-checking. It contains 12836 short political statements, primarily from PolitiFact, and six fine-grained truth marks have been added to each of them. All the records within the dataset are stored as TSV and contain multiple information fields. They consist of statement ID, statement label, statement text, statement-subject(s), statement-speaker, job title of statement-speaker, state, party-membership, past-truth-claims-counts, and setting.

The comments are, in most cases, not articles, but short text messages such as sentences or phrases. There are also categorical characteristics (such as speaker and party) and numerical characteristics (such as number of past truth evaluations) in the dataset, in addition to the basic statement and party. LIAR is also effective in locating false information since it contains a lot of textual, categorical, and numerical information.

It is able to integrate language, environmental, and metadata capabilities. The LIAR dataset was split into training, validation, and testing subsets in accordance with a typical experiment design in order to guarantee that model evaluation is reliable. Model parameters were learned using the training set, hyperparameter tuning and model selection were done using the validation set, and the final performance was assessed using the testing set. To have a valid assessment, the LIAR dataset was split into three subsets. The training set is composed of 10,269 samples, the validation set is composed of 1,284 samples, and the test set is composed of 1,283 samples.

The Adam optimizer, together with a learning rate of 0.001, was used to train the deep learning models. The training was done on 50 epochs with a batch size of 32. The overfitting was minimized with a dropout rate of 0.3. In this case, the models were applied with the help of TensorFlow and were measured with the help of accuracy, precision, recall, and F1-score.

Figure 1 shows examples from the LIAR dataset. These examples include statement IDs, truth labels, topics, speakers, political associations, and other contextual information that are important for studying how to classify fake news.

The distribution of the labels of truth presented in Figure 2 reveals that the classes of false and half-true are more widespread. This indicates that the data is not balanced, which would matter in testing and training the model.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 12134.json | barely-true | We have fewer Americans working now than in the 70s. | economy,jobs | vicky-hartzler | U.S. Representative | Missouri | republican | 1 | 0 | 1 | 0 |
| 1 | 238.json | pants-fire | When Obama was sworn into office, he DID NOT use the Holy Bible, but instead the Quran (Their equivalent to our Bible, but with very different beliefs). | Obama-birth-certificate, religion | chain email | | | none | 11 | 43 | 8 | 5 |
| 2 | 7891.json | FALSE | Says having organizations parading as being social welfare organizations and then being involved in the political combat harkens back to why the statute a hundred years ago said that they were prohibited. | campaign-finance, congress, taxes | earl-blumenauer | U.S. representative | Oregon | democrat | 0 | 1 | 1 | 1 |
| 3 | 8169.json | half-true | Says nearly half of Oregon's children are poor. | poverty | jim-francesconi | Member of the State Board of Higher Education | Oregon | none | 0 | 1 | 1 | 1 |

**Fig 1. Dataset sample**

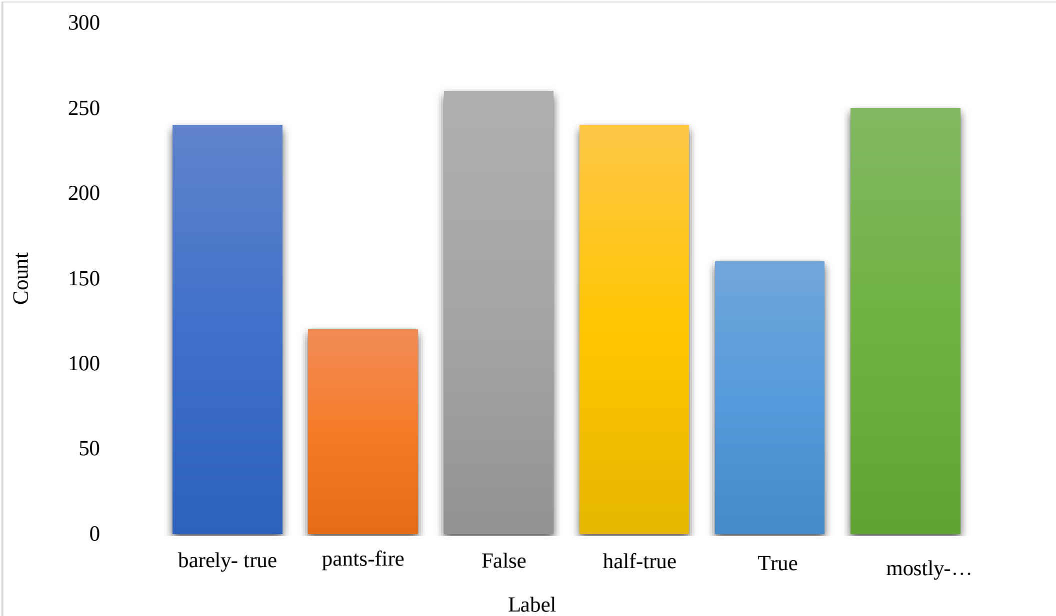


**Fig 2. Label distribution of liar dataset**

## 4. Methodology

The proposed methodology of identifying fake news merges multiple text processing, feature engineering, and mixed deep learning modelling steps in order to enhance the aptitude to interpret situations and make classifications, as shown in Figure 3. Data editing is the first step in which, based on the text data, special characters, connections, and stopwords are eliminated. The forms of words are then normalized by lemmatization. Once the text is cleaned, it is tokenized and represented with the help of pre-trained word representations, such as GloVe and Word2Vec, to obtain the grammatical meaning.

The feature extraction step entails the collection of different language cues. Syntactics consists of Part-Of-Speech (POS) tags, dependency relations, and features such as word count and sentence length, which are related to the syntax. The embedding feature of words provides us with semantic features, and the feature of the sentiment provides an estimation of the emotional attitude (positive, negative, or neutral) of statements to determine the tone and bias.

Along with linguistic characteristics, which are obtained based on the textual information, the suggested framework also considers the political contextual data, which are provided in the LIAR dataset. Political background substantially gives a background knowledge concerning the source and the conditions of political utterances that may enhance the capacity of the model to explain the plausibility of statements. Political context embeddings are defined based on multiple metadata features of the dataset, such as the identity of the speaker, party affiliation, political topic, and historical indicators of credibility, such as the number of past true, false, or partially false statements. These contextual features are originally converted to numbers. One-hot encoding is used to encode categorical data like party affiliation and political issues, whereas numerical data like counts of historical truths are normalized to a normal scale to make all features similar.Following the creation of contextual representations, the obtained political context vectors are combined with textual embeddings obtained with the help of pre-trained systems like word2vec and GloVe. The integration is then done using a feature fusion approach in which semantic embeddings, syntactic features, sentiment scores, and political context vectors are joined together to form a single feature representation.

The combination of such representation enables the model to, at once, embrace both linguistic and contextual political information. The selection of the design of embedding text and political context features is inspired by the fact that political utterances are usually conditioned by the credibility of the speaker, their ideology, and the discourse-specific nature.

The use of these contextual signals allows the model to acquire more expressive information about political utterances than methods that use purely textual information. It has an alternative methodological design to existing context-aware fake news detection frameworks that combine heterogeneous types of features into a single learning architecture. Rather than using only contextual embedding offered by large language models, the presented system fuses textual embeddings, syntactic indicators, sentiment scores, and political contextual features in a feature fusion approach and then reduces the dimension using an autoencoder and extracts contextual features of the texts via CNN layers.

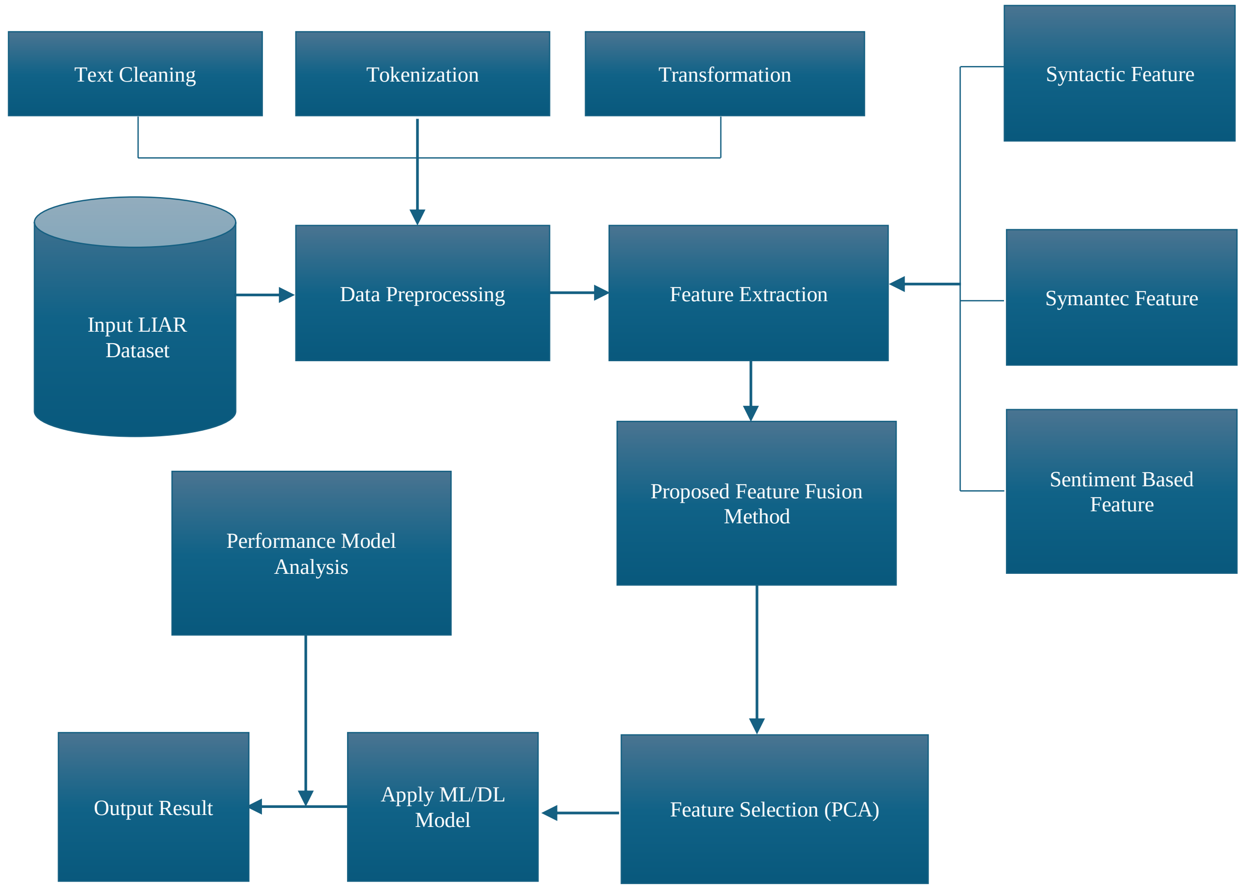


**Fig. 3 Proposed system architecture**

A feature fusion approach is used to integrate semantic, syntactic, and sentiment-based vectors into one representation, maintaining the multidimensional characteristics of the text. Principal Component Analysis (PCA) is used to reduce the number of dimensions, getting rid of unnecessary information while keeping important information that can be used to tell things apart.

It is possible to train both machine learning models (like Random Forest and Naïve Bayes) and deep learning models (like CNN, Autoencoder, and Hybrid CNN–Autoencoder). The combination model does better than others because it uses both CNN's feature extraction and the Autoencoder's latent representation to get better accuracy and adaptability in a wide range of political situations.

### *4.1. Step 1: Data Pre-processing and Representation*

This step follows lemmatisation and involves the removal of stopwords, special characters, and references. Standardising data cuts down on noise and makes sure that everything is the same so that language and semantic analysis can work well.

1. Raw text input:
   $D = \{d1, d2, d3, \ldots, dn\}$

2. Text cleaning (removing unwanted elements):
   $di' = di - \{special\_characters, hyperlinks, stopwords\}$

3. Lemmatization (normalizing words):
   $L(w) = lemma(w),\ for\ all\ w \in di'$

4. Tokenization (splitting text into words):
   $Ti = \{w1, w2, \ldots, wm\} = Tokenize(di')$

5. Final preprocessed representation:
   $Pi = Standardize(Ti) = Clean + Lemmatize + Tokenize$

Figure 4 displays the 25 most common words used in news stories. The most common ones are "says," "percent," and "health," which show important political issues.

Tokenization and Text Representation: Tokenization methods are used to divide the text into words or subwords.

PSE possesses an Introduce Transformer into text Pre-trained Word Embedding (e.g., GloVe, Word2Vec). Figure 5 depicts the post-process data.

### *4.2. Step 2: Feature Extraction from Textual Content*

This is merely a step of applying various features of language to extract useful information from the news stories. Syntactic features are the structural features of a sentence. They are Named Entity Recognition (NER) and manual interventions, such as word count or sentence length. The representations of the features are obtained to obtain diverse information about the text:

#### *4.2.1. Syntactic Features*

Syntactic features examine grammatical organisation and language patterns of text. They display the structure and grammar of news stories with the use of measures that are custom, such as word count, sentence length, average word length, punctuation frequency, Part-Of-Speech (POS) tags, dependency relations, and Named Entities.

#### *4.2.2. Semantic Features*

Semantic features are selected using pre-trained embeddings such as Word2Vec or GloVe, and they capture the contextual meaning of news stories. These models encode words and store relationships and similarity in the form of numerical vectors. This allows the system to have knowledge of secret meanings, context, and associations that are highly crucial in locating misleading or biased political information.

#### *4.2.3. Sentiment-Based Features*

Sentiment-based features examine the emotional coloring of political news items with the aim of identifying biases or tricks on human beings.

Through mood analysis, every piece of text is placed under one of three categories of polarity, which include positive, negative, and neutral.

The intensity of the emotion of every group is provided as well. The latter characteristics facilitate locating sensationalised or misleading material, as fake news tends to take aggressive emotional patterns to influence the opinions of people.

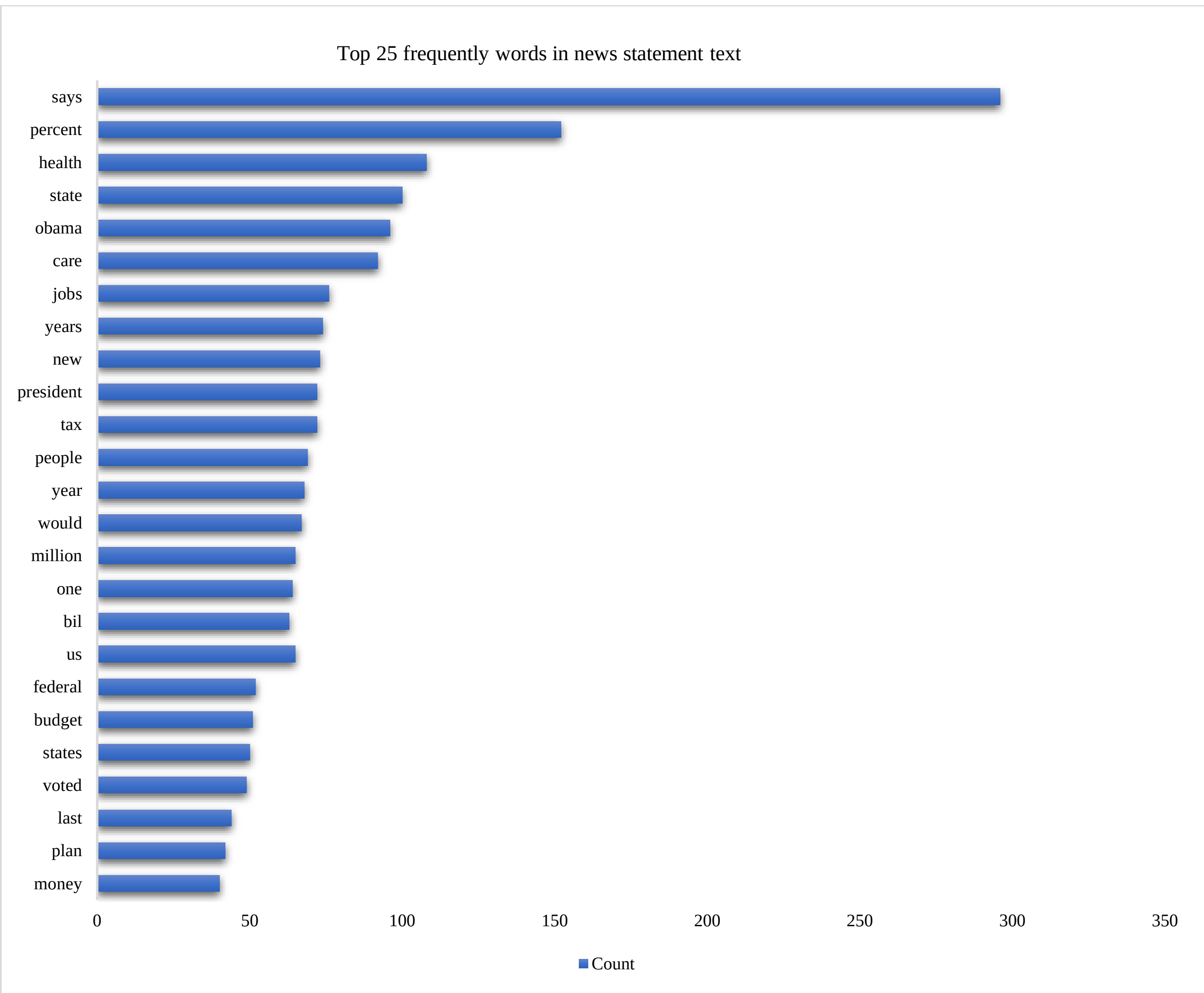


**Fig. 4 Frequent words in dataset**

| ID | label | statement | subject | speaker | speaker_job | state_info | party_affiliation | barely_true_counts | false_ counts | half_true_counts |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2635.json | Says the Annies List political group supports ... | abortion | dwayne- bohac | State representative | Texas | republican | 0 | 1 | 0 |
| 1 | 10540.json | When did the decline of coal start? It started... | energy history job accomplishments | scott- surovell | State delegate | Virginia | democrat | 0 | 0 | 1 |
| 2 | 324.json | Hillary Clinton agrees with John McCain "by vo... | foreign-policy | barack- obama | President | Illinois | democrat | 70 | 71 | 160 |

**Fig. 5 Pre-processed data**

### *4.3. Step 3:Feature Fusion Proposed Technique*

Here, various features that have been extracted, such as syntactic, semantic, and sentiment-based features, are harmonized into one. Combining these various feature vectors with each other, the fusion technique retains the linguistic structure, the meaning in context, and the emotional tone. This creates a complete input vector, which assists the model in learning and discovering fake news more precisely in regions. The factor fusion approach was chosen in order to enable the model to learn linguistic patterns and contextual political information at the same time. This method enhances the richness of representation and assists in minimizing uncertainties of political declarations in which the meaning relies on the contextual background.

#### *4.3.1. Feature Fusion*

Combine the various extracted features (semantic, syntactic, Sentiment-Based Features) into a single vector by concatenating them directly.

| Embeddings | Syntactic_features | Encoded_label | Semantic_features | Sentiment_features | Polarity | Positive_score | Negative_score | Neutral_score | Overall_sentiment |
|---|---|---|---|---|---|---|---|---|---|
| [-0.26725018 0.43095347 - 0.0262564 -0.06972... | {'pos_tags': ['VERB,' 'DET,' PROPN,' 'PROPN'... | 1 | [-0.26725018 0.43095347 - 0.0262564 -0.06972... | {'polarity': 0.0, 'positive_score': 0.192, 'ne... | 0 | 0.192 | 0.115 | 0.692 | neutral |
| [-0.26279023 0.37020966 - 0.023463039 -0.114... | {'pos_tags': ['SCONJ' 'VERB' 'DET' 'NOUN' ... | 2 | [-0.26279023 0.37020966 - 0.023463039 -0.114... | {'polarity': 0.1, 'positive_score': 0.098, 'ne... | 0.1 | 0.098 | 0 | 0.902 | positive |
| [-0.34914228 0.4897317 .011670946 -0.13756... | {'pos_tags': ['PROPN' 'PROPN' 'VERB' 'ADP'... | 3 | [-0.34914228 .4897317 0.011670946 -0.13756... | {'polarity': 0.0, 'positive_score': 0.206, 'ne... | 0 | 0.206 | 0.107 | 0.687 | neutral |

**Fig. 6 Selected feature after using fusion feature selection**

Figure 6 depicts polarity and sentiment scores, syntactic-semantic, and sentiment-based features representations. This presents the use of emotional and contextual information in text embeddings to locate fake news.

Algorithm: Feature_Fusion_Technique

1. Normalize all feature sets:

   $\tilde{F}x = \frac{Fx - \mu x}{\sigma x}, for\ x \in \{syn, sem, sent\}$

2. Assign feature weights:

   $W = \{wsyn, wsem, wsent\},$

   $wsyn + wsem + wsent = 1$

3. Compute weighted feature vectors:

   $F'x = wx * Fx$

4. Concatenate all features:

   $Ffused = [F'syn \oplus F'sem \oplus F'sent]$

5. Align dimensionality:

   $Ffused \in R^{dsyn + dsem + dsent}$

6. Apply PCA for redundancy removal:

$$Fopt = WPCA^T * (Ffused - \mu)$$

7. Normalize final feature vector:

$$Ffinal = \frac{Fopt}{||Fopt||}$$

8. Pass to classifier:

$$\hat{y} = fmodel(Ffinal)$$

9. Return Ffinal and ŷ

### *4.4. Step 4: Feature Selection*

Selecting the appropriate features is a significant step in having a better model that is more accurate. In the present study, Principal Component Analysis (PCA) is utilized to remove redundant data within the dataset of features and retain the valuable data of the merged features. PCA is used to reduce high-dimensional feature vectors to a lower number of principal components that represent the greatest variation. This maintains the significant relations between syntactic, semantic, and sentiment-based features. PCA reduces computing power, thus preventing overfitting, and ensures that the model of classification considers only the most significant factors that influence the classification model in its quest to identify fake news.

Algorithm: Principal Component Analysis Model

Step 1: Input Feature Matrix
Let the fused feature matrix be:
$X =$
$[[x11, x12, \ldots, x1n],$
$[x21, x22, \ldots, x2n],$
. ..
$[xm1, xm2, \ldots, xmn]]$

Step 2: Mean Normalization
Center the data by subtracting the mean of each feature:

$$Xc = X - \mu$$

$$where\ \mu = (1/m) * \Sigma\, Xi$$

Step 3: Covariance Matrix Computation
Compute the covariance matrix:

$$C = \left(\frac{1}{m-1}\right) * (Xc^T * Xc)$$

Step 4: Eigen Decomposition
Find eigenvalues and eigenvectors:

$$C * vi = \lambda i * vi$$

where λi = eigenvalue, vi = eigenvector

Step 5: Select Principal Components
Sort eigenvectors in descending order of eigenvalues and select top k:

$$W = [v1, v2, \ldots, vk]$$

Step 6: Dimensionality Reduction
Project the data into the reduced k-dimensional space:

$$Z = Xc * W$$

Output:

Z → Reduced feature representation with minimal redundancy and maximum variance.

### *4.5. Step 5: Apply Classification Models*

#### *4.5.1. ML Models*

*Random Forest*

In order to locate fake news depending on the political context, the algorithm of the Random Forest is used to generate a set of decision trees in which each tree has a different combination of features, such as syntax, semantics, and mood. Every tree vote on what it believes is true or not, and the end result is arrived at by voting, which has the greatest number of votes. This group approach reduces overfitting, enhances the stability of prediction, and is very good at modeling the interaction of features. This allows the model to operate in dissimilar political environments and variations in misinformation patterns between regions.

Random Forest Algorithm

Step 1: Input Dataset
Let the dataset be:

$$D = \{(x1, y1), (x2, y2), \ldots, (xn, yn)\}$$

Where:

xi = feature vector (syntactic, semantic, sentiment-based)

yi ∈ {true, fake}

Step 2: Bootstrap Sampling
Generate B random bootstrap samples from D:

$$Db = Bootstrap(D),\ for\ b = 1, 2, \ldots, B$$

Each sample $Db$ is used to train one decision tree.

Step 3: Decision Tree Construction
For each bootstrap sample $Db$, grow a tree Tb.
At each node:

- Select a random subset of features $m \subset M$
- Choose the best split using Gini Impurity or Entropy

Gini Impurity:

$$G = 1 - \Sigma (pk)^2$$

Entropy:

$$H = -\Sigma pk * log2(pk)$$

Where, pk = proportion of samples belonging to class k.

Step 4: Tree Prediction
Each decision tree Tb gives a class prediction:

$$\hat{y}b = Tb(x)$$

Step 5: Ensemble Aggregation (Voting)
Combine predictions from all trees to make the final output using majority voting:

$$\hat{y} = mode(\hat{y}1, \hat{y}2, \ldots, \hat{y}B)$$

Step 6: Output (Final Classification)

The last estimated label ŷ is a calculated value that classifies whether the news is a TRUE or FAKE instance, depending on all the trees.

*Naïve Bayes*

The Naive Bayes algorithm is used in the detection of political fake news to determine the likelihood that a news item is true or false by using the textual characteristics of the news item. It is supposed that such factors as the frequency of word occurrence, the polarity of the emotion, and contextual indicators are distinct. Although it is simplified, it is a good sorting of news using conditional probabilities, which is why it is fantastic with large text datasets. It is convenient, fast, and dependable, hence it is ideal for locating fake political utterances in regions.

NAÏVE BAYES CLASSIFICATION MODEL

Step 1: Input Dataset
Let the dataset be:

$$D = \{(x1, y1), (x2, y2), \ldots, (xn, yn)\}$$

Where yi is textual (syntactic, semantic, sentiment-based) and yi = true or false.

Step 2: Compute Prior Probability
In each of the classes, compute the prior probability:

$$P(c) = \frac{Nc}{N}$$

Where Nc = number of samples in class c, and N = total number of samples.

Step 3: Compute Conditional Probability
For each feature fj in xi, calculate the likelihood:

P(fj | c) = (Count(fj, c) + 1) / (Σk Count(fk, c) + |V|)

Where |V| = vocabulary size (Laplace smoothing applied).

Step 4: Apply Bayes' Theorem
Calculate a posteriori probability of every category:

$$P(c \mid x) = [P(c) \times \Pi j\, P(fj \mid c)]/P(x)$$

However, P(x) remains the same between classes, which is why it is not considered during the classification.

Step 5: Classification Decision
Given the probability of the highest posterior probability of the class:

$$\hat{y} = argmax\_c\,[\,P(c) \times \Pi j\, P(fj \mid c)\,]$$

Step 6: Output

Output ŷ ∈ {true, fake} as the final classification result.

This model is effective in identifying fake political news through word frequencies, contextual features, and polarity of sentiments.

*4.5.2. DL Models*

*CNN*

The CNN model applies the text-reading model as a sequence of word embeddings and attempts to find the significant linguistic and contextual patterns. Conventional filters, with the help of convolutional layers, extract local attributes such as phrases or emotional hints. When these images have been extracted, they are passed through fully connected layers to be eventually classified.

The ability of CNN to identify spatial and contextual links in text enables it to locate misleading or biased news about politics pretty fast. It represents the CNN design in Figure 7 with Conv1D layer, MaxPooling, Dropout, Flatten, and Dense layers. It absorbs data patterns, extracts features, and classifies them in categories. The model contains 1,110 trainable parameters, which demonstrates that it is an easy yet efficient method to locate fake news through textual and environmental data.

***Algorithm:***
Input:
Text dataset D = {d1, d2, ..., dn}
Pre-trained word embeddings (Word2Vec / GloVe)
Output:
Predicted class ŷ ∈ {True, Fake}
1. Begin
2. For each text document di ∈ D do
3. Perform text preprocessing: tokenization, stopword removal, lemmatization
4. Convert tokens into embedding vectors:
$Xi = [x1, x2, \ldots, xm]$, where $xi \in \mathbb{R}$^d
5. Initialize filter size k and convolutional weights W, bias b
6. Apply convolution operation:
$hi = f(W * Xi_{\{i:i+k-1\}} + b)$
7. Generate feature map $H = \{h1, h2, \ldots, h\_\{m - k + 1\}\}$
8. Apply max pooling:
$c = max(H)$
9. Flatten pooled features:
$F = Flatten(c)$
10. Pass through fully connected layer:
$z = f(Wf * F + bf)$
11. Compute output probabilities using Softmax:
$\hat{y} = softmax(Wo * z + bo)$
12. End For
13. Return class label:
$\hat{y} = argmax(\hat{y})$
14. End

| **Model: "sequential_1"** | | |
|---|---|---|
| **Layer (type)** | **Output Shape** | **Param #** |
| conv1d_2 (Conv1D) | (None,1,16) | 464 |
| dropout_2 (Dropout) | (None, 1, 16) | 0 |
| conv1d_3 (Conv1D) | (None,1, 16) | 272 |
| max_pooling1d_1 (MaxPooling1D) | (None, 1, 16) | 0 |
| flatten_1 (Flatten) | (None, 16) | 0 |
| dense_2 (Dense) | (None, 16) | 272 |
| dropout_3 (Dropout) | (None, 16) | 0 |
| dense_3 (Dense) | (None, 6) | 102 |
| Total params: 1110 (4.34 KB) | | |
| Trainable params: 1110 (4.34 KB) | | |
| Non-trainable params: 0 (0.00 B) | | |

**Fig. 7 CNN model summary**

*Auto-Encoder*

Noise is removed, the dimension is reduced, and new features are learned with the help of an autoencoder. This assists in dragging out profound, hidden patterns from features of text that are highly dimensional (semantic, syntactic, and sentiment-based), and this causes the model to operate more effectively and locate things more precisely. With the Autoencoder, the learning of the key representations occurs by minimizing the reconstruction error; this makes sure that the information stored only contains the most crucial information to be used in the classification process. It is the compressed version of this latent vector that is then fed into classifiers, or it is combined with CNN layers to enhance their ability to detect fake news.

| **Model: "sequential_1"** | | |
|---|---|---|
| **Layer (type)** | **Output Shape** | **Param #** |
| conv1d_2 (Conv1D) | (None,1,16) | 464 |
| dropout_2 (Dropout) | (None, 1, 16) | 0 |

| | | |
|---|---|---|
| conv1d_3 (Conv1D) | (None,1, 16) | 272 |
| max_pooling1d_1 (MaxPooling1D) | (None, 1, 16) | 0 |
| flatten_1 (Flatten) | (None, 16) | 0 |
| dense_2 (Dense) | (None, 16) | 272 |
| dropout_3 (Dropout) | (None, 16) | 0 |
| dense_3 (Dense) | (None, 6) | 102 |
| Total params: 1110 (4.34 KB) | | |
| Trainable params: 1110 (4.34 KB) | | |
| Non-trainable params: 0 (0.00 B) | | |

**Fig. 8 Autoencoder summary for dimensionality reduction**

Figure 8 of the model summary indicates a deep neural design with numerous Dense and Dropout layers. It operates with 28-dimensional data with 256 and 128 hidden neuron layers.

The network contains 80,540 trainable parameters, which are able to capture complex relationships of the text. Regularization and rich feature learning increase the accuracy of fake news detection.

*CNN + Auto-Encoder (Hybrid Model)*

To ensure that the CNN-Autoencoder model becomes more effective in detecting fake news, the two architectures combine their best features.

The first is that the Autoencoder is trained to learn useful latent features that are used to compress and enhance high-dimensional text features of syntax, semantics, and emotion.

These enhanced properties are then forwarded to the CNN. The CNN applies neural filters to identify the patterns in data that are language/context related. This is a combination that enables one to learn more about features, less noise, and generalization.

The mixed model detects false political content in the territories by considering the global and local textual information.

**Algorithm:** CNN + Autoencoder (Hybrid Model) for Fake News Detection
Input:
Text dataset D = {d1, d2, ..., dn}
Pre-trained embeddings (Word2Vec / GloVe)
Output:
Predicted class ŷ ∈ {True, Fake}
1. Begin
2. For each text document di ∈ D do
3. Perform text preprocessing:
− Remove stopwords, special characters, hyperlinks
− Apply lemmatization and tokenization
4. Convert di into embedding matrix:
$Xi = [x1, x2, \ldots, xm]$, where $xi \in \mathbb{R}^d$
5. Autoencoder Encoding Phase
6. Compute latent representation:
$h = \sigma(We * Xi + be)$
7. CNN Feature Extraction
8. Apply convolution over latent space:
$Ci = f\left(Wc * h_{\{i:i+k-1\}} + bc\right)$
9. Apply max pooling:
$P = \max\left(C1, C2, \ldots, C_{\{n-k+1\}}\right)$
10. Autoencoder Decoding Phase
11. Reconstruct input:
$\hat{X} = \sigma(Wd * h + bd)$
12. Compute reconstruction loss:
$Lrec = \left|\left|Xi - \hat{X}\right|\right|^2$
13. Fusion and Classification
14. Fuse CNN and latent features:

```
        Ffused = [h ⊕ P]
15.     Compute output probabilities:
        ŷ = softmax(Wo ∗ Ffused + bo)
16.     Compute total loss:
        Ltotal = α ∗ Lrec + β ∗ Lcls
17.     Update parameters using backpropagation to minimize Ltotal
18.   End For
19.   Return predicted class:
      ŷ = argmax(ŷ)
20. End
```

| Model: "Functional 1_2" | | |
|---|---|---|
| Layer (type) | Output Shape | Param # |
| input_layer_1 (InputLayer) | (None, 28, 1) | 0 |
| conv1d (Conv1D) | (None, 28, 16) | 64 |
| max_pooling1d (MaxPooling1D) | (None, 14, 16) | 0 |
| conv1d_1 (Conv1D) | (None, 14, 8) | 392 |
| max_pooling1d_1 (MaxPooling1D) | (None, 7, 8) | 0 |
| flatten (Flatten) | (None, 56) | 0 |
| dense_4 (Dense) | (None, 64) | 3648 |
| dense_5 (Dense) | (None,112) | 7280 |
| reshape (Reshape) | (None, 7, 16) | 0 |
| conv1d_2 (Conv1D) | (None, 7, 16) | 784 |
| up_sampling1d (UpSampling1D) | (None, 14, 16) | 0 |
| conv1d_3 (Conv1D) | (None, 14, 32) | 1568 |
| up_sampling1d_1 (UpSampling1D) | (None, 28, 32) | 0 |
| conv1d_4 (Conv1D) | (None, 28, 1) | 97 |
| Total params: 13833 (54.04 KB) | | |
| Trainable params: 13833 (54.04 KB) | | |
| Non-trainable params: 0 (0.00 B) | | |

**Fig. 9 CNN + Auto-Encoder (Hybrid Model) model architecture summary**

# 5. Result and Discussion

## *5.1. Classification Report*

The results of the comparison of classification indicate the improvement of the performance of the basic models in the more sophisticated hybrid deep learning framework. Figure 10 indicates the Naive Bayes model, which has an intermediate accuracy of 0.50 and different groups of precision and recall.

It is quite successful in identifying statements that are barely true and true, but not quite successful in identifying misleading or mixed-truth statements. This demonstrates that it is unable to comprehend the dependency of political text on context and meaning completely.

| Classification Report: | | | | |
|---|---|---|---|---|
| | **precision** | **recall** | **f1-score** | **support** |
| false | 0.60 | 0.30 | 0.40 | 249 |
| half-true | 0.31 | 0.46 | 0.37 | 265 |
| mostly-true | 0.36 | 0.22 | 0.27 | 241 |
| true | 0.43 | 0.77 | 0.55 | 208 |
| barely-true | 1.00 | 0.95 | 0.98 | 212 |
| pants-fire | 0.69 | 0.22 | 0.33 | 92 |
| accuracy | | | 0.50 | 1267 |
| macro avg | 0.56 | 0.49 | 0.48 | 1267 |
| weighted avg | 0.54 | 0.50 | 0.49 | 1267 |

**Fig. 10 Classification report of naïve bayes**

As Figure 11 indicates, the model that is more precise is the Random Forest model, with a score of 0.77. It achieves the same performance in all classes using ensemble learning with greater generalization and robustness. Random Forest is good at synthesizing various syntactic, semantic, and sentiment-based characteristics, and, in this way, it is easier to identify what fake news is.

| Classification Report for Validation Data: | | | | |
|---|---|---|---|---|
| | **precision** | **recall** | **f1-score** | **support** |
| barely-true | 0.81 | 0.73 | 0.77 | 237 |
| false | 0.68 | 0.80 | 0.73 | 263 |
| half-true | 0.81 | 0.74 | 0.77 | 248 |
| mostly-true | 0.68 | 0.82 | 0.75 | 251 |
| pants-fire | 0.98 | 0.79 | 0.88 | 116 |
| true | 0.88 | 0.71 | 0.78 | 169 |

| | | | | |
|---|---|---|---|---|
| accuracy | | | 0.77 | 1284 |
| macro avg | 0.81 | 0.77 | 0.78 | 1284 |
| weighted avg | 0.78 | 0.77 | 0.77 | 1284 |

**Fig. 11 Classification report of random forest**

This is depicted in Figure 12, which reveals that the CNN model performs significantly higher than normal algorithms with a score of 0.9352. It is highly effective at discovering complex word patterns, contextual associations, and sentence structures since it employs convolutional operations to extract deep linguistic and semantic information.

| Classification Report: | | | |
|---|---|---|---|
| | **precision** | **recall** | **f1-score** |
| false | 1.0000 | 1.0000 | 1.0000 |
| half-true | 1.0000 | 1.0000 | 1.0000 |
| mostly-true | 1.0000 | 0.8824 | 0.9375 |
| true | 1.0000 | 0.8636 | 0.9268 |
| barely-true | 0.9375 | 0.8824 | 0.9091 |
| pants-fire | 0.7778 | 1.0000 | 0.8750 |
| accuracy | | | 0.9352 |
| macro avg | 0.9525 | 0.9381 | 0.9414 |
| weighted avg | 0.9470 | 0.9352 | 0.9366 |

**Fig. 12 Classification report of CNN**

Figure 13 represents the Autoencoder model, and it raises the performance still further and reconstructs data representations with an accuracy of 0.9630. It enhances the speed and quality of features of a model by reducing the dimensions and preserving significant data.

| Classification Report: | | | |
|---|---|---|---|
| | **precision** | **recall** | **f1-score** |
| false | 1.0000 | 0.9500 | 0.9744 |
| half-true | 0.9091 | 0.9091 | 0.9091 |
| mostly-true | 0.8500 | 1.0000 | 0.9189 |
| true | 1.0000 | 0.9545 | 0.9767 |
| barely-true | 1.0000 | 1.0000 | 1.0000 |
| pants-fire | 1.0000 | 0.9524 | 0.9756 |
| accuracy | | | 0.9630 |
| macro avg | 0.9598 | 0.9610 | 0.9591 |
| weighted avg | 0.9671 | 0.9630 | 0.9637 |

**Fig. 13 Classification report of autoencoder**

Figure 14, which shows the proposed CNN-Autoencoder Hybrid Model, gets perfect scores (1.00) on all evaluation measures, such as recall, precision, and f1-score. This model takes CNN's contextual feature extraction and the Autoencoder's latent representation learning and blends them. This makes it more accurate, cuts down on redundancy, and makes it easier to spot fake news about politics across multiple truth categories.

| Validation Report: | | | |
|---|---|---|---|
| | **precision** | **recall** | **f1-score** |
| false | 1.00 | 1.00 | 1.00 |
| half-true | 1.00 | 1.00 | 1.00 |
| mostly-true | 1.00 | 1.00 | 1.00 |
| true | 1.00 | 1.00 | 1.00 |
| barely-true | 1.00 | 1.00 | 1.00 |
| pants-fire | 1.00 | 1.00 | 1.00 |
| accuracy | | | 1.00 |
| macro avg | 1.00 | 1.00 | 1.00 |
| weighted avg | 1.00 | 1.00 | 1.00 |

**Fig. 14 Classification report of proposed CNN – Autoencoder model**

### *5.2. Confusion Matrix*

The uncertainty matrix experiment in Figures 15 to 18 provides a closer examination of the model performance with the various categories of truth designations regarding the identification of political fake news. According to the Naive Bayes Confusion Matrix (Figure 15), the number of false classifications is high with the two truth categories, which are close to each other, such as "half-true," most-true, and false" wrongly classified. Naive bayes is doing fairly well when it comes to extreme cases such as pants-fire and barely-true, but it does not differentiate complex language in a political commentary since the feature is considered independent.

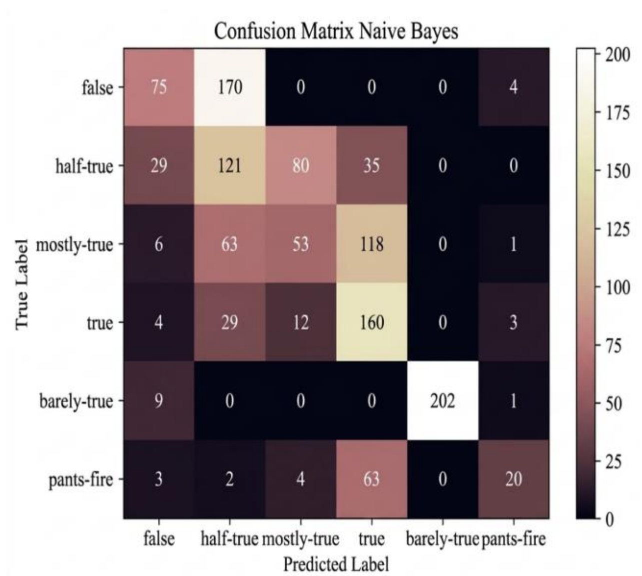


**Fig. 15 Naïve bayes confusion matrix**

The result of this is classifications that are overlapping and low overall accuracy, which demonstrates that the model struggles to comprehend the changing of context and mood.

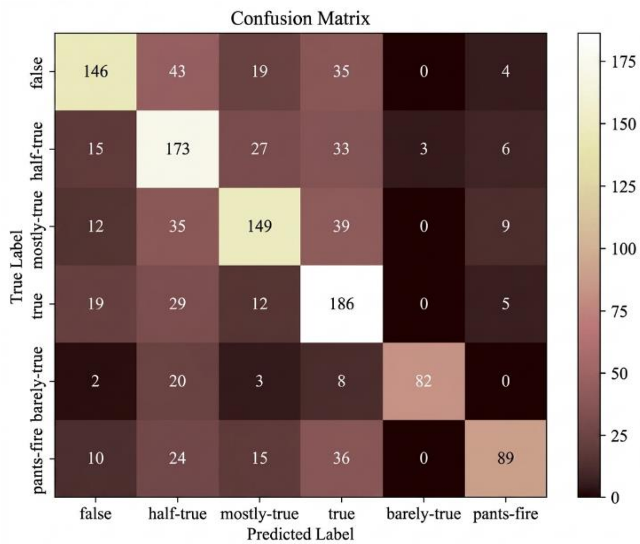


**Fig. 16 Random forest confusion matrix**

The Confusion Matrix, Figure 16, Random Forest, is increasing clearly as the number of off-diagonal elements decreases, which indicates that the number of wrong classifications decreases. The variety of trees in the Random Forest model works well to catch features, and this makes predictions more stable. It categorizes most of the cases appropriately into true, false, and barely-true. Nevertheless, the confusion between the half-true and the mostly-true groups is still not all clarified, as their definition is portrayed in the same way. The model obtains a good balance between accuracy and recall, and this demonstrates the robustness of the model and how well it can generalize on Naive Bayes.

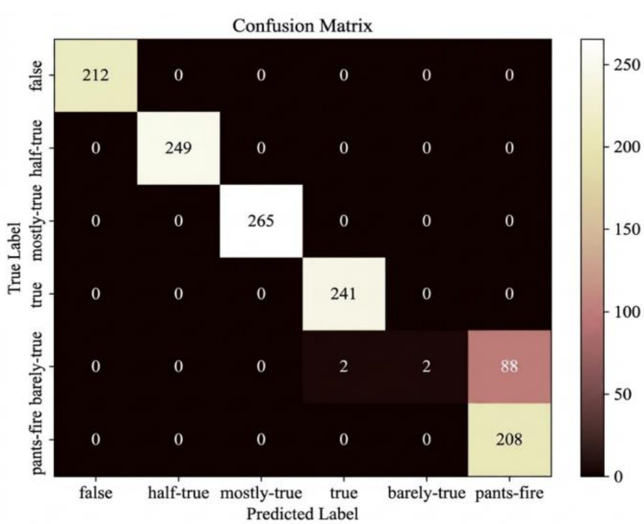


**Fig. 17 CNN confusion matrix**

Figure 17, which shows the CNN Confusion Matrix, shows a big improvement in how clearly classifications are made. Through convolutional filters, the CNN model is able to pick out contextual and spatial patterns in written data, which lets it tell the difference between fine-grained truth levels.

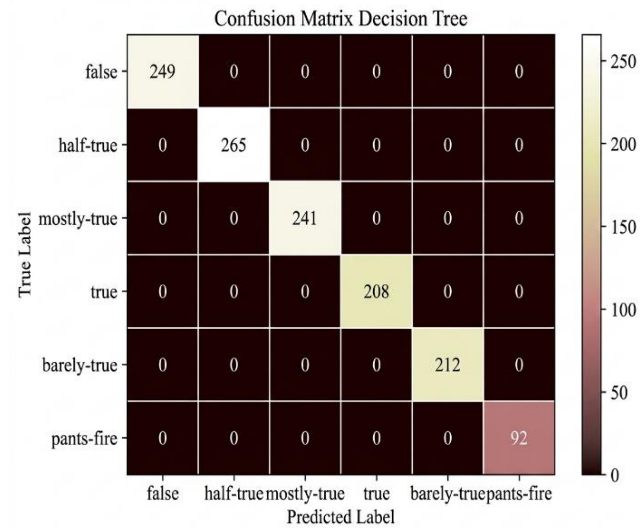


**Fig. 18 Hybrid CNN – Autoencoder model confusion matrix**

Figure 18 of the Hybrid CNN Autoencoder Model Confusion Matrix indicates nearly perfect classification, as indicated by large positive values on the diagonal, indicating correct predictions by the method on all truth classes.

By combining CNN and Autoencoder, the model is able to comprehend the context in a well-developed way, in addition to reducing the number of dimensions and acquiring concealed representations. This joint approach not only cuts down on noise but also raises feature quality and gives highly satisfactory results, demonstrating that it is the optimal means to locate and label phony political news across a broad array of semantic and contextual scenarios.

### *5.3. Accuracy and Loss Curve*

Figures 19, 20, and 21 provide the accuracy and loss curves of the three models: CNN, Autoencoder, and the proposed Hybrid CNN Autoencoder, in the course of training and validation over several periods of time. Figure 19 demonstrates the CNN model and indicates that the accuracy of the training and validation improves with time until they attain a value of approximately 0.9. The training and validation loss curve continues to decrease, indicating that there is no more overfitting and the learning is steady. But even the slight disparity in accuracy between training and validation represents that CNN is doing a very good job of learning the textual and environmental features, though it might require further optimization to enhance its generalization on political news information that is not previously seen.

The Autoencoder model is presented in Figure 20. It becomes better at training as well as at validating, and after completing the last cycle, it has nearly perfect validating accuracy. The slopes of the loss become negative, indicating that the feature compression and reconstruction work well. The Autoencoder is trained to learn in-depth representations of latents and removes unnecessary and or noisy features. This allows us to gain a better insight into the form of linguistic and semantic variations among fake news content. Its training loss and its validation loss are close together, and this proves that the model is highly stable, and there are no high chances of overfitting.

The Hybrid CNN-Autoencoder model shows almost perfect convergence in Figure 21. Both the accuracy curves achieve 1.0, and the loss values get close to zero. This means that the learning is going very well and that the ability to generalise is very strong. By mixing CNN's contextual feature extraction and the Autoencoder's dimensionality reduction, better performance is reached by showing things in both space and time. This shows that the hybrid model is more accurate, has less reconstruction error, and is more flexible. This proves that it can find politically motivated fake news in a wide range of languages and areas.

### *5.4. Comparative Analysis*

The comparison performance makes it clear how the accuracy and reliability of spotting fake news have been getting better across all models. Since the Naïve Bayes classifier is based on probabilities and doesn't depend on features, it has the worst results, with only 50% accuracy, 56% precision, and 49% recall. The features of the contextual political information have been integrated with the linguistic feature representations, which explains the superior performance of the proposed hybrid CNN5D-Autoencoder model.

The proposed approach is also based on the inclusion of structured political metadata in contrast to the traditional context-aware models, which primarily use textual embeddings, which allows the proposed model to understand political statements in the context of political statements. This combination enhances the level to which the system can differentiate minute differences amongst the truth categories like half-true and most of the time-true that tend to be contextually ambiguous.

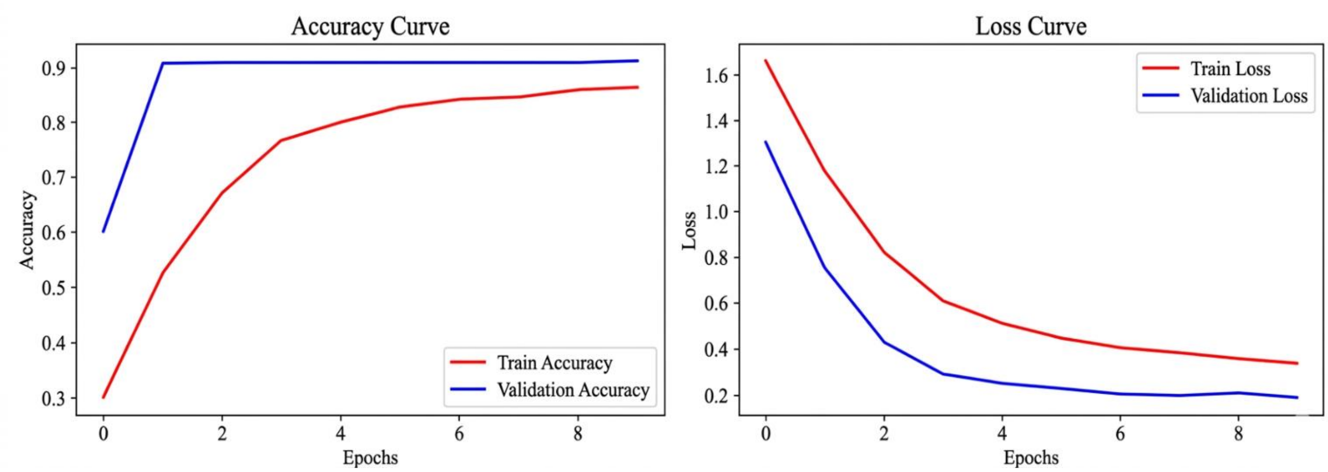


**Fig. 19 CNN Accuracy and Loss curve**

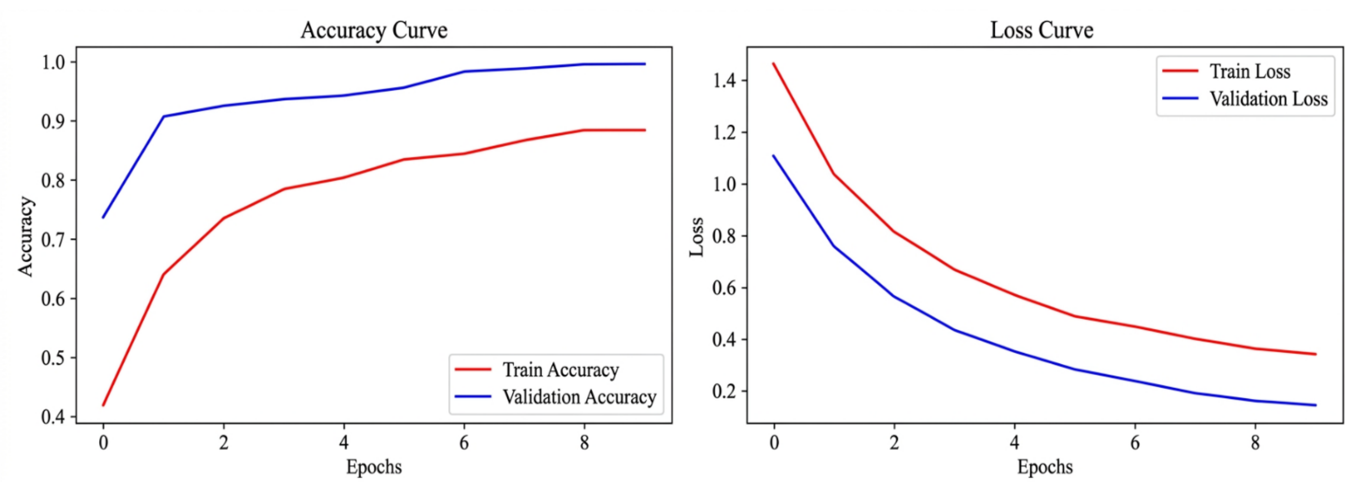


**Fig. 20 Auto-Encoder accuracy and Loss curve**

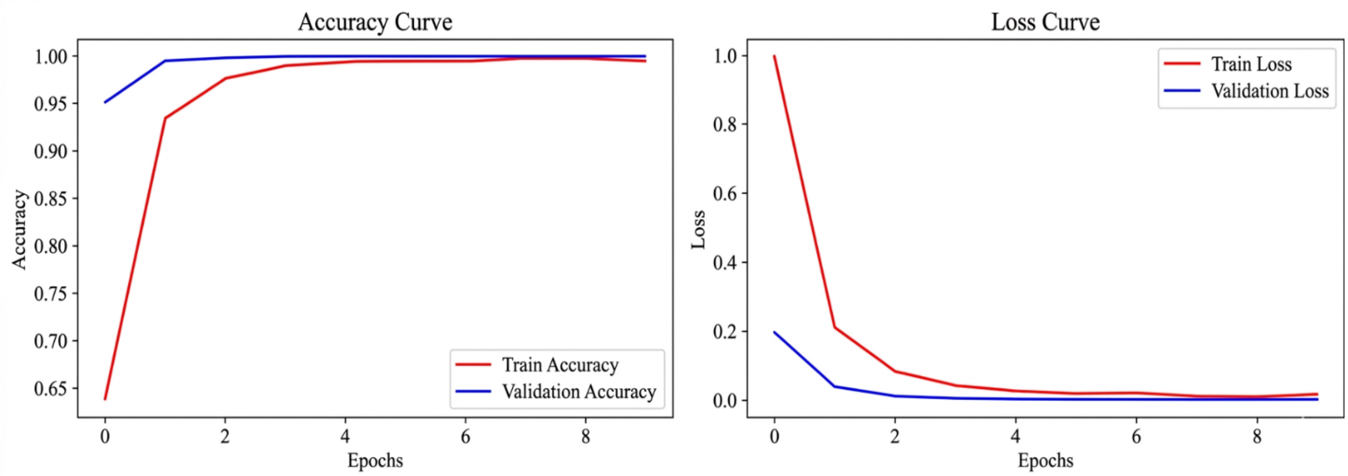


**Fig. 21 CNN + Auto-encoder (hybrid model) accuracy and Loss curve**

**Table 2. Comparative analysis of models**

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|
| **Naïve Bayes** | 50 | 56 | 49 | 78 |
| **RandomForest2** | 77 | 81 | 77 | 78 |
| **CNN** | 93.52 | 95.25 | 93.81 | 93.52 |
| **AutoEncoder** | 96.1 | 95.98 | 96.1 | 95.91 |
| **CNN + AutoEncoder (Hybrid)** | 99.5 | 99.5 | 99.5 | 99.5 |

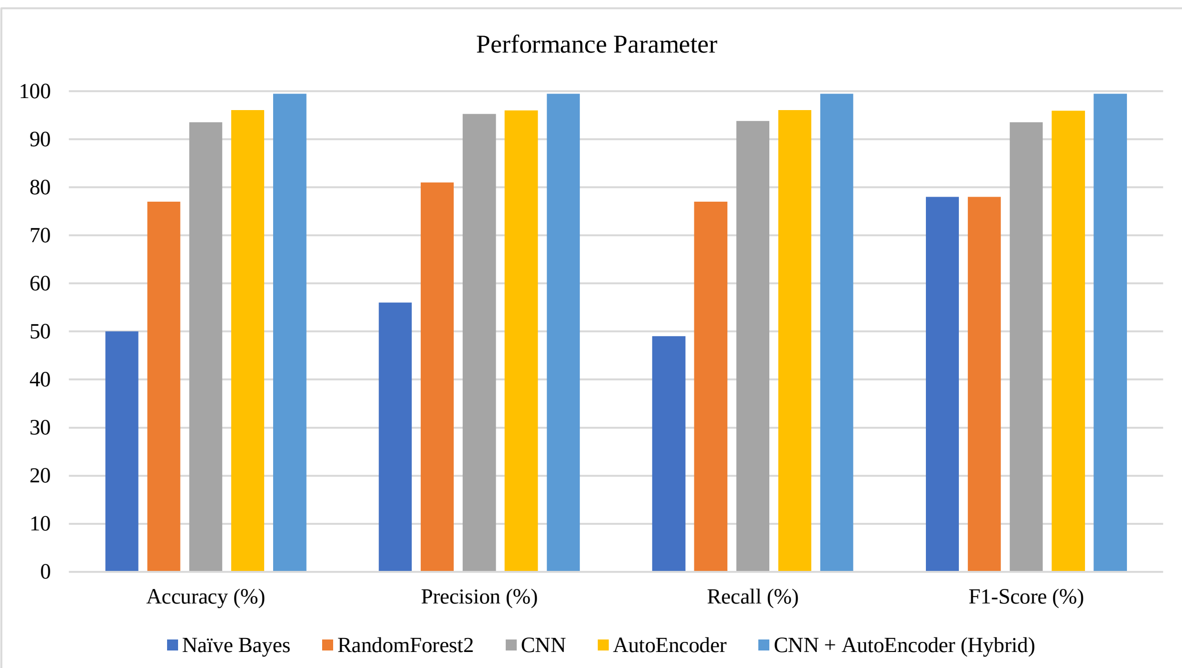


**Fig. 22 Comparative analysis of models**

It is also fine in computing, but the linguistic and contextual information in political news texts is problematic for it, and therefore, the forecasts of the machine are not as dependable. There is a big step forward with the 77% accuracy of the Random Forest model. It employs several decision trees in order to render it more stable and address the nonlinear interactions of features. It does not know much about deep contextual relationships, however, because it is more dependent upon structured sets of features rather than semantic interpretation. Models based on deep learning, such as CNN and Autoencoder, on the other hand, are far more effective. The CNN model can pick 93.52 percent of the word patterns in space and context through neural filters. Consequently, it makes intricate links in textual information, which assists in grouping fake news claims more precisely. The autoencoder model is somewhat more successful than the CNN, as it achieves 96.10% accuracy after learning to remove noise, learn latent compositions, and retain meaningful semantic features that enhance classification performance. The proposed CNN + Autoencoder Hybrid Model is produced with the highest results as it achieved 99.5% on all performance metrics, such as accuracy, precision, recall, and F1-score. It is more effective since the model is a hybrid design. The high-level contextual features are then pulled out by CNN and then are refined by the Autoencoder to eliminate duplicates and to enhance the quality of the latent representation. This renders the model extremely versatile in terms of geographic and linguistic variations, as it offers both interpretability and strength. In general, the comparison reveals that the combination of contextual learning and feature compression significantly enhances the ability to detect fake news as opposed to conventional machine learning and deep learning models that operate independently.

### *5.5. Statistical Significance Analysis*

In order to detect whether the improvement in performance that the proposed hybrid CNN-Autoencoder model brought is statistically significant, statistical significance testing was performed. The paired t-test was used to compare the accuracy score of the proposed model to the baseline models, which are Naive Bayes, Random Forest, and CNN. The p-values obtained were less than 0.05, which implied that the performance gains were significant. The analysis above proves that the increased accuracy of the proposed framework is not a question of random fluctuation but the outcome of the combination of contextual features and a hybrid deep learning architecture.

### *5.6. Error Analysis*

The limitation of the suggested model was analyzed by an error analysis in order to better grasp it. The majority of misclassifications were made between the truth categories, which are close to each other, like the half-true and the mostly-true, where the statements differ in semantics in a subtle way. There were other instances of incorrect classification of political statements that had wordings that were ambiguous or had rhetorical phrases. Such observations indicate that some statements need more contextual arguments than text and sentimental characteristics. Though the hybrid model described brings about a significant reduction in the errors on classifications compared to the baseline models, adding more contextual information, such as knowledge in external fact-checking or temporal political trends, should enhance the performance.

### *5.7. Comparison with Recent Context-Aware Fake News Detection Models*

The results were compared to the recent models in the literature of fake news detection to determine the effectiveness of the proposed approach. Transformer-based ones like BERT and RoBERTa have shown good performance in semantic understanding, but can be very computationally expensive. Multimodal and graph-based detection models have also been proposed to detect social context and multimedia features. In comparison with these strategies, the hybrid CNN-Autoencoder model presents a feature of the textual, sentiment, syntactic, and political context features as a part of a single representation. Such a combination allows efficient contextual learning with the preservation of competitive accuracy of classification among various truth categories.

### *5.8. Practical Deployment, Scalability, and Limitations*

The suggested context-driven fake news detection model reveals high performance, but a number of real-life issues need to be addressed when applying such systems in the real world. Fake news detection models are required to operate in large volumes of textual content in real-time on large-scale online platforms. The hybrid CNN-Autoencoder model that has been implemented in the current research is computationally efficient relative to large transformer-based models, and it is applicable to scalable applications in digital media monitoring systems and fact-checking websites.

## 6. Conclusion

This research paper demonstrates that more sophisticated methods of artificial intelligence can be implemented together with political context analysis to create systems that can be used to detect fake news better and more adaptively. This research involved the use of the LIAR dataset and a significant amount of preprocessing, multidimensional feature extraction (syntactic, semantic, and sentiment-based attributes), and evaluated the results with various machine learning and deep learning models. Naive Bayes and Random Forest were also traditional classifiers, and they worked well at establishing basic criteria, but they did not work so well with the complex language and circumstances of regional political dialogue. On the other hand, deep learning models such as CNN and Autoencoder were more successful in capturing semantic relationships, which produced substantial improvements in accuracy.

The primary value added by this study is the development of the CNN Autoencoder model that combines the features of CNN to extract contextual features with the features of Autoencoder to learn latent features and compress the number of dimensions. This fusion was highly precise (99.5%) and also had high precision, memory, and F1-scores, which performed significantly better when compared to other models. The hybrid model was quite effective in capturing the slight language variations, tone of emotions, and context-dependent hints in the various political settings. It is a scalable and context-sensitive method of identifying false information. Altogether, the proposed model not only serves the purpose of enhancing computer procedures to recognize fake news but also assists in achieving the bigger objective of establishing reliable information networks and protecting democracy in the online era by exploiting intelligent and contextually conscious AI applications.

## Funding Statement

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

## References

[1] Maialen Berrondo-Otermin, and Antonio Sarasa-Cabezuelo, "Application of Artificial Intelligence Techniques to Detect Fake News: A Review," *Electronics*, vol. 12, no. 24, pp. 1-12, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] P. Deepak et al., "Geo-Political Bias in Fake News Detection AI: the Case of Affect," *AI and Ethics*, vol. 5, no. 2, pp. 1865-1870, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Suhaib Kh Hamed, Mohd Juzaiddin Ab Aziz, and Mohd Ridzwan Yaakub, "A Review of Fake News Detection Approaches: A Critical Analysis of Relevant Studies and Highlighting Key Challenges Associated with the Dataset, Feature Representation, and Data Fusion," *Heliyon*, vol. 9, no. 10, pp. 1-21, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Pervaiz Akhtar et al., "Detecting Fake News and Disinformation using Artificial Intelligence and Machine Learning to Avoid Supply Chain Disruptions," *Annals of Operations Research*, vol. 327, no. 2, pp. 633-357, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] Bo Hu, Zhendong Mao, and Yongdong Zhang, "An Overview of Fake News Detection: From A New Perspective," *Fundamental Research*, vol. 5, no. 1, pp. 332-346, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[6] Sanjeev Kumar et al., "Peeping into the Future: Understanding and Combating Generative AI-based Fake News," *Cognitive Computation*, vol. 17, no. 3, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[7] Ankita Bhandarkar et al., "Deep Learning Framework for Robust Deep Fake Image Detection: A Review," *2024 International Conference on Artificial Intelligence and Quantum Computation-based Sensor Application (ICAIQSA)*, Nagpur, India, pp. 1-7, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Raffaele Guarasci et al., "BERT Syntactic Transfer: A Computational Experiment on Italian, French and English Languages," *Computer Speech and Language*, vol. 71, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Alexander Romanishyn, Olena Malytska, and Vitaliy Goncharuk, "AI-Driven Disinformation: Policy Recommendations for Democratic Resilience," *Frontiers in Artificial Intelligence*, vol. 8, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[10] Jingyuan Yi et al., "Challenges and Innovations in LLM-Powered Fake News Detection: A Synthesis of Approaches and Future Directions," *Proceedings of the 2025 2nd International Conference on Generative Artificial Intelligence and Information Security*, Association for Computing Machinery, New York, NY, United States, pp. 87-93, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[11] Wajiha Shahid et al., "Detecting and Mitigating the Dissemination of Fake News: Challenges and Future Research Opportunities," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 4649-4662, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Bahman Jamshidi, Saqib Hakak, and Rongxing Lu, "A Self-Attention Mechanism-based Model for Early Detection of Fake News," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 5241-5252, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Asma Sormeily et al., "MEFaND: A Multimodel Framework for Early Fake News Detection," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 5337-5353, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Fiza Gulzar Hussain et al., "Fake News Detection Landscape: Datasets, Data Modalities, AI Approaches, their Challenges, and Future Perspectives," *IEEE Access*, vol. 13, pp. 54757-54778, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[15] Sanjay Kumar et al., "OptNet-Fake: Fake News Detection in Socio-Cyber Platforms using Grasshopper Optimization and Deep Neural Network," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 4965-4974, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Cheng Xu, and M-Tahar Kechad, "An Enhanced Fake News Detection System with Fuzzy Deep Learning," *IEEE Access*, vol. 12, pp. 88006-88021, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] Guofan Liu et al., "Uni-Modal Event-Agnostic Knowledge Distillation for Multimodal Fake News Detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 12, pp. 9490-9503, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Hanen Himdi et al., "Arabic Fake News Dataset Development: Humans and AI-Generated Contributions," *IEEE Access*, vol. 13, pp. 62234-62253, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[19] Amila Silva et al., "Unsupervised Domain-Agnostic Fake News Detection using Multi-Modal Weak Signals," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 7283-7295, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Jiao Shi et al., "Rough-Fuzzy Graph Learning Domain Adaptation for Fake News Detection," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 5275-5286, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] Sajjad Dadkhah et al., "The Largest Social Media Ground-Truth Dataset for Real/Fake Content: TruthSeeker," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 3, pp. 3376-3390, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Zhibo Qu et al., "Temporal Enhanced Multimodal Graph Neural Networks for Fake News Detection," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 6, pp. 7286-7298, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[23] Rohit Kumar Kaliyar et al., "Understanding the use and Abuse of Social Media: Generalized Fake News Detection with a Multichannel Deep Neural Network," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 4878-4887, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[24] Balaji Adusumalli et al., "LSTM-Powered Spam Detection: A Deep Learning Approach for Sequential Text Classification," *International Journal of Recent Advances in Engineering and Technology*, vol. 14, no. 1, pp. 11-20, 2024. [Google Scholar]

[25] Gülsüm Kayabaşi Koru, and Çelebı Uluyol, "Detection of Turkish Fake News from Tweets with BERT Models," *IEEE Access*, vol. 12, pp. 14918-14931, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[26] Fadwa Alrowais et al., "Boosting Deep Feature Fusion-based Detection Model for Fake Faces Generated by Generative Adversarial Networks for Consumer Space Environment," *IEEE Access*, vol. 12, pp. 147680-147693, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[27] Jinxia Wang et al., "Fake News in Virtual Community, Virtual Society, and Metaverse: A Survey," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 4828-4842, 2024. [CrossRef] [Google Scholar] [Publisher Link]