Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Source-Aware Kurdish Fake News Detection: Bridging Qur'anic Wisdom and Deep Learning

azad Karim

azad.mstafa@chu.edu.iq

Charmo University

Bryar Hassan

University of Kurdistan Hewler

Research Article

Keywords: Fake News Detection, Kurdish NLP, Source Verification, Deep Learning, Convolutional Neural Networks, Low-Resource Languages, Misinformation Detection

Posted Date: November 6th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11367977/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Source-Aware Kurdish Fake News Detection: Bridging Qur'anic Wisdom and Deep Learning

Azad M. Karim[1]* and **Bryar A. Hassan[2]-[1]**

[1] Department of Computer Science, College of Science, Charmo University, 46023 Chamchamal/Sulaimani, KRI, Iraq

[2] Computer Science and Engineering Department, University of Kurdistan Hewler, Erbil, Iraq

Corresponding author: Azad M. Karim
Email: azad.mstafa@chu.edu.iq

Tel: +964 (7701457514)

Corresponding author's ORCID: [0009-0004-7064-7413]

Another author's email: bryar.ahmad@ukh.edu.krd

ORCID: [0000-0003-3682-5576]

Abstract

However, automated detection of misinformation is difficult in low-resource languages because of the sheer volume of misinformation in these languages. This study aimed to detect fake news in the Kurdish language, using the Kurdish Dataset for Fake News Detection (KDFND), which is composed of 100,966 labeled articles. This work compared a variety of models, ranging from traditional machine learning (ML) (logistic regression (LR), random forest (RF), XGBoost, and support vector machines (SVM)) to ensemble models (voting and stacking classifiers) and deep learning (DL) architectures (CNN and BiLSTM-Attention). Another important contribution is the source-aware feature representation (based on the concept of source verification in the Qur'an, Surah Al-Hujurat 49:6), which combines information from news sources with the text of news articles, as it demonstrates the empirical validation of the critical role of source credibility in the detection of misinformation.

The results show that the source-aware models come close to perfection in terms of performance, with CNN and BiLSTM-Attention reaching a 100% F1-score. In terms of accuracy, the source-aware CNN and BiLSTM-Attention models both achieved 100%, while the stacking and XGBoost classifiers followed closely at 99.98%. The average performance of the models drops significantly by 18.44% without source information, with the best model using only content (CNN) dropping to 82.97% F1. The accuracy of the best content-only model (CNN) was 82.43%. The performance gap empirically validates the Qur'anic command that information cannot be evaluated on its own without corroboration of its source. The CNN model was the best solution for real-world deployment, with perfect accuracy and the fastest training time (10 seconds) and inference time (0.01 seconds).

This study introduces four contributions: (1) comprehensive benchmarking of Kurdish fake news detection: ML, ensemble, and DL; (2) empirical validation of the Qur'anic principle of source verification; (3) optimal model for Kurdish fake news detection in a resource-constrained environment: CNN; and (4) baseline performance metrics for Kurdish NLP research. The methodology is language-independent and can be generalized to other low-resource languages for which source credibility can provide meaningful cues to detect misinformation.

Keywords: Fake News Detection, Kurdish NLP, Source Verification, Deep Learning, Convolutional Neural Networks, Low-Resource Languages, Misinformation Detection

1. Introduction

In the age of information sharing, anyone can publish information, which can spread quickly and widely, as can misinformation. Deceptive content, which endangers democratic processes, public health initiatives, and social cohesion, has grown on social media platforms and instant messaging applications [4, 20]. This is especially true for low-resource languages, where there is a lack of annotated data and underdeveloped natural language processing (NLP) tools, which are significant obstacles to detecting fake news [6].

One linguistic community that is poorly served is the Kurdish one. Kurdish is spoken by approximately 30 million people in Turkey, Iraq, Iran, and Syria and has less NLP infrastructure than widely resourced languages [1]. The Kurdish media is highly politicized, includes a number of dialects (Kurmanji and Sorani), has different levels of press freedom, and often lacks credibility as a source, especially when it comes to outlets covering the Kurds and their affairs [2].

The concept of source verification was uttered more than 1400 years ago in the Holy Qur'an! Surah Al-Hujurat (49:6) states:

"يَا أَيُّهَا الَّذِينَ آمَنُوا إِن جَاءَكُمْ فَاسِقٌ بِنَبَإٍ فَتَبَيَّنُوا أَن تُصِيبُوا قَوْمًا بِجَهَالَةٍ فَتُصْبِحُوا عَلَىٰ مَا فَعَلْتُمْ نَادِمِينَ" "O you who have believed, if there comes to you a disobedient one with information, investigate, lest you harm a people out of ignorance and become, over what you have done, regretful."

This verse is an epistemological statement, namely that information needs to be investigated and the credibility of the information source needs to be confirmed before it is disseminated. Ancient wisdom is remarkably aligned with present-day methods used to counteract the spread of fake news: source provenance is as important as content analysis in the identification of misinformation, The central hypothesis.

While there is a significant increase in the interest in detecting fake news, there are still numerous gaps in research. First, there is a huge imbalance between the amount of research conducted on high-resource languages and the amount of research conducted on low-resource languages such as Kurdish [6]. Second, most previous methods focus on content-based features and tend to ignore the credibility of the sources, which is especially problematic in polarizing media environments [3]. Third, the incorporation of domain-specific knowledge (e.g., cultural and religious significance of source verification) has not been studied much.

This study makes four main contributions.

1. Methodological Contribution: This study proposed a feature representation that combines source information with article text and conducted empirical experiments demonstrating that source verification, a well-known principle from the Qur'anic injunction, is important for detection performance.
2. Empirical Contribution: Offered detailed empirical results for the detection of Kurdish fake news using various model architectures, and provided a benchmark for future work on fake news detection in low-resource languages.
3. Practical Contribution: Optimal models for Kurdish fake news detection with practical applications for real-world deployment in resource-constrained settings.
4. Cultural Contribution: Contribute to the creation of linguistic resources for an underserved language community; and connect ancient wisdom with modern computational techniques, in the wake of a critical research need for Kurdish NLP.

The remainder of this paper is organized as follows: The literature review is discussed in the next section (Section 2). In Section 3, we explain our methodology; from data preprocessing to feature engineering and model implementation. The experimental results and discussion, and analysis are presented in Section 4. The paper is concluding in Section 5.

2. Literature Review

With the increasing amount of misinformation being spread on digital platforms, fake news detection has become a vital research area. In this section, the literature is reviewed by investigating the methods, algorithmic frameworks, and assessment metrics used in fake news detection. Focusing especially on low-resource language applications and the importance of source credibility are themes central to our study.

In the context of fake news detection, DL has proven to be extremely effective, especially for high-resource languages. Almandouh et al. [8] introduced an ensemble of high-performance DL models, with which they obtained state-of-the-art performance using multiple architectures. They showed that model diversity enhances the robustness of the detection and that their ensemble approach takes advantage of the complementary strengths of various neural networks designs.

To resolve this problem, Maham et al. [5] proposed adversarial news net (ANN), a framework for robust fake news classification in an adversarial setting. The strategy used included adversarial training methods, which aim to make their models more resilient to attacks on their detection systems, a key challenge in adversarial attacks.

Verma et al. [10] presented a deep ensemble network specially designed for fake news detection in online text, named ScrutNet. They were designed using a combination of DL pathways, enabling them to learn a variety of linguistic features and outperforming other benchmark data sets.

To enhance this field, Hashmi et al. [17] presented a hybrid DL method that combines FastText embeddings with explainable AI (XAI) methods. They reported that model interpretability is essential for fake news detection in the real world, where stakeholders need to understand the classification decisions, highlighting the importance of such interpretability.

Traditional ML techniques can be applied to the problem of fake news detection, particularly in resource-limited settings. Abdul et al. [15] performed a comparative analysis of ML techniques for fake news detection using three datasets and algorithms such as SVM, RF, and Naïve Bayes.

Although textual information is scarce, Convolutional Neural Networks (CNNs) have proven successful for detecting fake news on Twitter as done by Tama and Sibaroni [13]. Their approach involved understanding patterns in the text using convolutional filters and was especially successful in understanding text patterns in short-form texts, like social media.

To solve the problem of overfitting in a recurrent neural network, Camelia and Fahim [14] proposed a regularized LSTM approach for fake news detection. They tested their regularization methods on various news domains and found that they boosted generalization.

The problem of detecting fake news in low-resource languages has been gaining traction. Karim and Hassan [1] suggested a hybrid approach using an AI-designed model (FastText) with an LSTM for detecting fake news in Kurdish and English. Their work is one of the few studies to focus on Kurdish misinformation, and indeed, marks a huge gap in research in this field. Daneshfar [7] investigated the potential of transfer learning for boosting low-resource sentiment analysis, showing that cross-lingual knowledge transfer from high-resource languages can have a strong positive impact on performance for underrepresented languages. This is especially true for Kurdish, where materials with annotations are rare. A systematic literature review by Alghamdi et al. [6] found the main challenges in fake news detection ML approaches in multilingual contexts were (1) data size, (2) dialectal variation, and (3) language-specific preprocessing pipelines.

Recent studies have highlighted the significance of source credibility in the identification of fake news. Wang et al. [3] proposed using global structure and publisher credibility to detect fake news. They showed that source reliability signals can be used to detect even before the content-based features are available. Alsmadi et al. [21] studied stance detection in the context of fake news to determine whether the position of an article relative to known facts could determine whether it is fake or real. They used a novel approach that focused on content analysis and external knowledge sources.

In a comparative study of automated fake news detection using ML techniques, Muscedere et al. [16] discovered that the models that include metadata features performed better than those using only content features, including source features, across various datasets. Recently the importance of multimodal information usage for fake news detection has been increasingly emphasized. Nasser et al. [4] conducted a systematic review of fake news detection using DL, which included the combination of textual, visual, and contextual features.

Abdullah et al. [11] proposed a multi-modal ensemble detection approach that is based on both visual and textual features, which revealed that multiple modalities have higher accuracy than unimodal approaches in detecting fake news. To investigate the transferability of multimodal approaches to the detection of misinformation, Nguyen et al. [9] used automated multimodal learning for the detection of fake advertisements in the Vietnamese real estate data domain.

A major concern is the susceptibility of fake news detectors to adversarial attacks. Koren et al. [12] studied the robustness of fake news detectors against adversarial attacks with genuine user comments and observed that state-of-the-art fake news detectors are vulnerable. To address the shortage of training data, Wu et al. [18] introduced GAN-augmented contrastive learning and multimodal attention for detecting fake news.

To provide a summary of the study that is most relevant to this work, key studies in the field of fake news detection are summarized in Table 1, which shows their contributions, algorithmic approaches, key findings, and gaps.

Table 1: Summary of Key Literature in Fake News Detection

Reference	Year	Algorithm/Method	Key Findings	Relevance to This Study
Karim & Hassan [1]	2025	FastText + LSTM Hybrid	Hybrid model effective for Kurdish/English fake news	Kurdish NLP foundation
Salh & Nabi [2]	2022	Dataset (KDFND)	KDFND	Dataset foundation
Almandouh et al. [8]	2024	Ensemble DL	Ensemble improves performance vs. single models	Ensemble methodology
Maham et al. [5]	2024	Adversarial News Net (ANN)	Robust classification under adversarial attacks	Robustness reference
Verma et al. [10]	2025	ScrutNet (Deep Ensemble)	Superior performance across benchmarks	Deep ensemble reference
Hashmi et al. [17]	2024	Hybrid DL + FastText + XAI	Hybrid approach with explainability	Hybrid methodology
Wang et al. [3]	2025	Global Structure + Publisher Credibility	Early detection via credibility	Source credibility foundation
Muscedere et al. [16]	2026	ML Comparative Analysis	Metadata (source) improves performance	Source feature importance
Alghamdi et al. [6]	2026	Systematic Literature Review	Identified multilingual challenges	Low-resource context
Daneshfar [7]	2024	Transfer Learning	Enhances low-resource sentiment analysis	Transfer learning reference

3. Methodology

This section introduces the complete methodological approach used to identify fake news in Kurdish by applying KDFND. The approach includes data preprocessing, feature engineering, model building, and evaluation techniques. To highlight the significance of strong preprocessing and the use of a variety of model architectures, this study includes both traditional ML and DL methods. The experimental design is based on best practices identified by Kuntur et al. [19], who showed that data quality and data preprocessing have a significant impact on the performance of the detection. Figure 1 provides a general overview of the research structure, showing the sequence of data collection, model evaluation, etc.

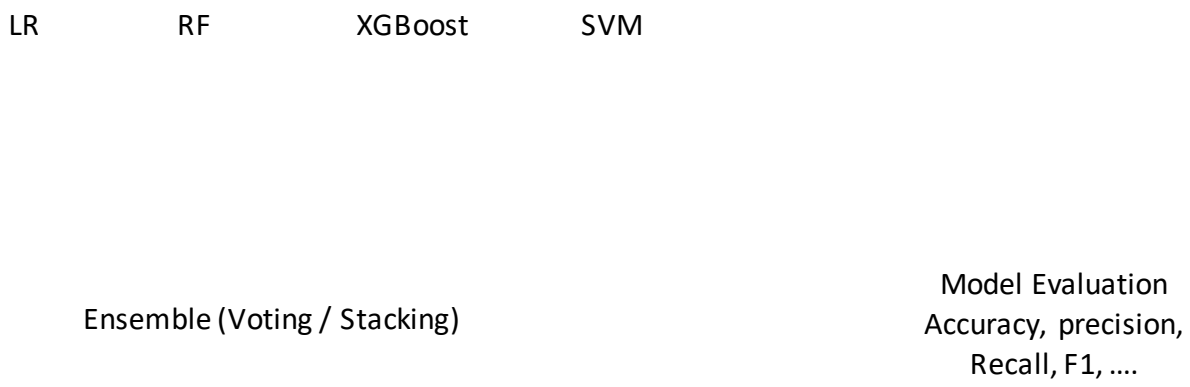


Figure 1: Overall Research Framework for Kurdish Fake News Detection

3.1 Dataset Description

3.1.1 KDFND Dataset

The KDFND [2] is a publicly available benchmark dataset tailored for research on Kurdish fake news. This dataset includes news reports from different Kurdish media publications in Iraq, Turkey, Iran, and Syria.

The balanced nature of the dataset reduces the problem of class imbalance that is generally faced in fake news detection tasks [10]. The following statistics of the dataset are listed in Table 2.

Table 2: Dataset Characteristics

Feature	Value
Total Samples	100,966
Real News	50,751 (50.27%)
Fake News	50,211 (49.73%)
Average Text Length (Real)	87.63 characters
Average Text Length (Fake)	251.32 characters
Average Word Count (Real)	11.38 words
Average Word Count (Fake)	34.36 words
Language	Kurdish (Sorani/Kurmanji)
Data Source	Multiple Kurdish media outlets
Labels	Binary (Real/Fake)

3.1.2 Dataset Characteristics

The dataset has some properties suitable for fake news detection research:

- **Balanced Class Distribution**: The dataset was balanced between real and fake news, with 90% (50.27%) and 10% (49.73%), respectively, without requiring any resampling technique, thus ensuring a reliable performance evaluation. A balanced class distribution ensures that the models do not become biased toward the majority class [20].
- **Significant Text Length Difference:** The significant difference in text length between real (87.63 characters) and fake (251.32 characters) articles implies that linguistic complexity has a discriminating property, which is in line with multilingual fake news detection [9]. The difference in length suggests that fake news articles are longer, perhaps because they try to make themselves sound more credible by using many words.
- **Source Diversity:** It is important to note that the articles are sourced from a variety of Kurdish media outlets with different political and geographical backgrounds to give an idea of the complexity of the Kurdish media landscape.
- **Textual Richness**: Articles are of different lengths and complexity and do reflect a realistic representation of the news content in Kurdish.
- **Multidialectal Content**: The dataset contains both Sorani and Kurmanji dialects, which add linguistic challenges to the NLP models.

3.1.3 Data Collection and Curation

This dataset was annotated and labeled by domain experts using an intensive annotation process [2]. The articles were categorized as "Real" or "Fake" based on:

- Cross-verification involves comparing a source with other reliable sources.
- Fact-checking is a check against facts.
- Source credibility assessment
- Content consistency evaluation

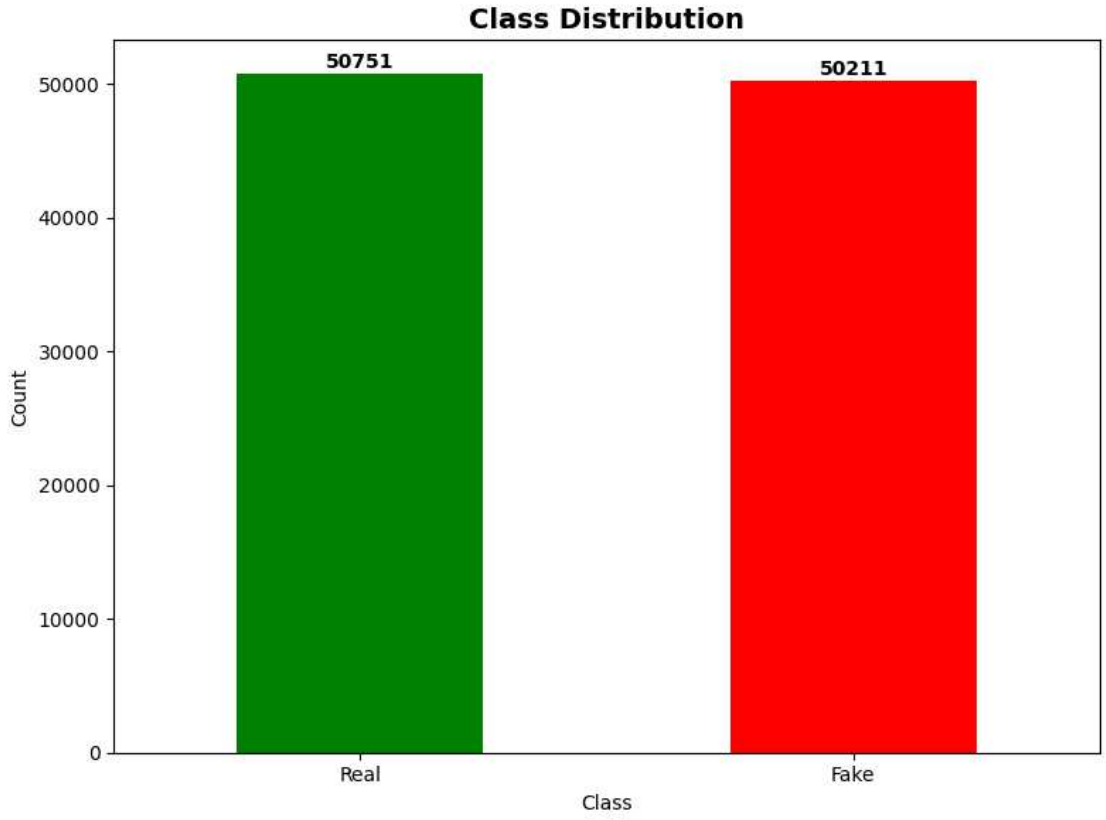


Figure 2: Class Distribution of Real vs. Fake News

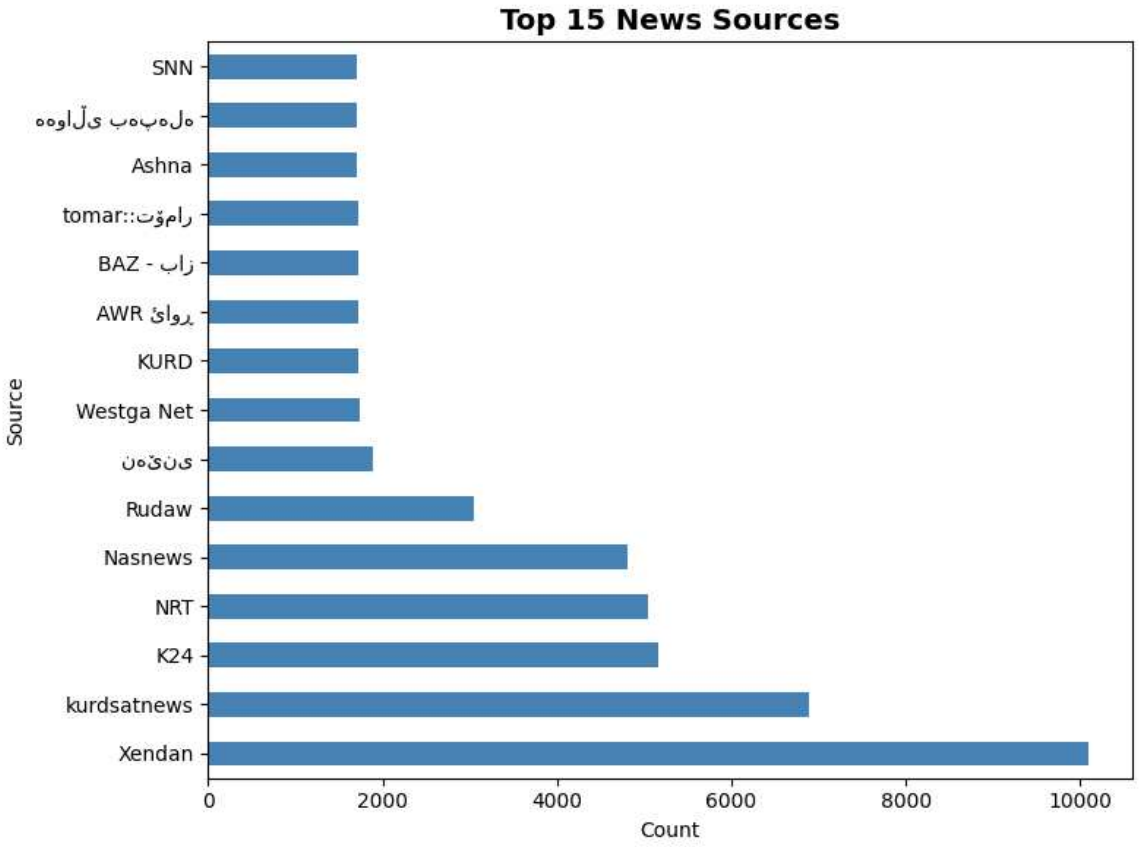


Figure 3: Top 15 News Sources Distribution

3.2 Data Preprocessing

3.2.1 Text Cleaning

A Kurdish text is standardized using a comprehensive pipeline for cleaning the text. The preprocessing steps were developed to eliminate irrelevant noise while retaining the linguistic content crucial for effective classification. These steps are taken to guarantee that the data are accurate and consistent among all samples, following the feature engineering procedures suggested by Kuntur et al. [4]. The preprocessing pipeline consists of:

1. **URL Removal:** To remove the URLs from the text corpus, regular expression patterns were used to filter out hyperlinks that are not relevant to the semantic content of the news articles.
2. **Mention and Hashtag Removal:** The removal of social-media-specific elements (mentioned people, hashtags, etc.) for the purpose of emphasizing the main linguistic content was inspired by Tama and Sibaroni [13] and is referred to as mention and hashtag removal.
3. **Punctuation Removal:** All punctuation marks were removed from the text to eliminate noise and normalize the vocabulary space.
4. **Digit Removal:** Numerical digits were omitted, as they are not normally relevant to the semantic meaning in this context and can add noise.
5. **Whitespace Normalization:** More one whitespace was replaced with a single space to enable better tokenization.
6. **Lowercase conversion:** All text was converted to lowercase to minimize vocabulary sparsities and make it case-insensitive.

3.2.2 Feature Representation

1. **Using text features in traditional machine learning**.

A custom feature extraction system was created to extract the linguistic features of news articles in the Kurdish language. This system uses basic and advanced statistical features of the cleaned text based on the behavioral feature analysis structure of Abdul et al. [15] behavioral feature analysis structure. The textual features are used as indicators of writing patterns that can distinguish between real and fake news. Table 3 listed all the features used in this study:

Table 3: Feature Set for Text Analysis

Feature Category	Feature Name	Description
Lexical Features	word_count	Number of words in the text
	char_count	Number of characters in the text
	unique_words	Count of unique words
	avg_word_length	Average word length
Stopword Features	stopword_count	Count of stopwords present
	stopword_ratio	Ratio of stopwords to total words
Statistical Features	unique_ratio	Ratio of unique to total words
Punctuation Features	punct_count	Count of punctuation marks
	punct_ratio	Ratio of punctuation to characters
Numerical Features	digit_count	Count of digits
	digit_ratio	Ratio of digits to characters
Stylistic Features	uppercase_count	Count of uppercase characters
	uppercase_ratio	Ratio of uppercase to characters

2. Feature Representation for Deep Learning

In the case of DL models, word embeddings were used with the Keras Tokenizer API. A 20,000-word vocabulary with an out-of-vocabulary (OOV) token was used in the tokenizer. The neural network architectures were trained with uniform input size by using post-padding and post-truncation techniques to make the sequences even at a maximum length of 150 tokens.

3.2.3 Source-Aware Feature Representation

A major outcome of the present study is the source-aware feature representation inspired by the principle of source verification (Surah Al-Hujurat 49:6) of the Qur'an, which commands believers to examine information that may be unreliable. Given the importance of source credibility in Kurdish media, which comes in a variety of forms and has different levels of credibility and political affiliation, two parallel feature sets were built for comparative analysis.

1. Content-Only Representation: The input only contains cleaned article text (cleaned_text), no source information. This is the default configuration.
2. Source-Aware Representation: The input is the cleaned article text and the news source identifier concatenated (Source + " " + cleaned_text). This enables models to learn linguistic patterns and credibility signals from the source.

This dual-representation method allows empirical quantification of source information's effect on detection performance, thus directly proving the principle of the Qur'an: "Source verification is a prerequisite for accurate information assessment."

3.3 Data Splitting

Stratified random sampling was used to partition the dataset into training and testing datasets to maintain the class distribution between the two. Thus, the training and testing sets have approximately 50% real and 50% fake news to avoid any bias in the model evaluation. In the split configuration, the data is shown in Table 4:

Table 4: Data Split Configuration

Partition	Size	Percentage	Real Samples	Fake Samples
Training	80,769	80%	40,169	40,600
Testing	20,193	20%	10,042	10,151
Total	**100,962**	**100%**	**50,211**	**50,751**

:

The 80-20 split ratio also adheres to the usual convention in ML studies, where there is sufficient data for training and a representative test set for unbiased assessment.

3.4 Models Development

3.4.1 Traditional Machine Learning Models

A comparative analysis approach was adopted, as illustrated by Alghamdi et al. [9] and Muscedere et al. [10], and four traditional ML algorithms were implemented with hyperparameter optimization. Hyperparameter tuning was systematically applied to different models to determine the optimal configurations.

- LR: This is a linear model that can be used as a baseline classifier. The hyperparameter tuning included different levels of regularization strength (C): 0.1, 0.5, 1.0, and 10.0; different solvers such as 'liblinear' and 'lbfgs'; and maximum iterations: 1000 and 2000.
- RF: Bagging of decision trees is an ensemble of decision trees that provides robust classification. Hyperparameter optimization investigated 100 and 200 estimators and maximum depths of 10, 20, and 'max.' Minimum samples split of 2, 5, and 10 and minimum samples leaf of 1, 2, and 4.
- XGBoost: A high-performance gradient-boosted tree model suitable for structured data tasks. The hyperparameter search considered 100 and 200 estimators, maximum depths of 4, 6, and 8, subsample ratios of 0.8 and 1.0, and learning rates of 0.01, 0.05, and 0.1.
- Support Vector Machine (SVM): A linear SVM implementation (LinearSVC) that determines the best hyperplane to separate features. Values of the regularization strength (C): 0.1, 1.0, and 10.0 were explored for the parameter search. The loss functions used were 'hinge' and 'squared hinge.'

Table 5 lists the traditional ML models and their configurations.

Table 5: Traditional Machine Learning Model Configurations

Model	Hyperparameter	Values Tested
LR	Regularization strength (C)	[0.1, 0.5, 1.0, 10.0]
	Solver	['liblinear', 'lbfgs']
	Maximum iterations	[1000, 2000]
RF	Number of estimators	[100, 200]
	Maximum depth	[10, 20, None]
	Minimum samples split	[2, 5, 10]
	Minimum samples leaf	[1, 2, 4]
XGBoost	Number of estimators	[100, 200]
	Maximum depth	[4, 6, 8]
	Learning rate	[0.01, 0.05, 0.1]
	Subsample ratio	[0.8, 1.0]
SVM (LinearSVC)	Regularization parameter (C)	[0.1, 1.0, 10.0]
	Loss function	['hinge', 'squared_hinge']

3.4.2 Ensemble Learning Models

Two ensemble methods were implemented following the ensemble approaches proposed by Abdullah et al. [7]:

1. Voting Classifier: A soft voting ensemble of LR, RF, and XGBoost classifiers, with weights [1, 1, 1]. Soft voting: The predicted probabilities are averaged from each model, yielding an average probability prediction across models. The architectural details of the voting classifier are presented in Table 6.

Table 6: Voting Classifier Architecture

Component	Specification
Base Models	LR, RF, XGBoost
Voting Strategy	Soft Voting (Probability Averaging)
Weights	Equal weights [1, 1, 1]

Ensemble Method	Probability averaging
Prediction	Class with highest average probability

2. Stacking Classifier: A two-level stacking model that transforms LR, RF, and XGBoost into base models (Level 0) and LR into a meta-learner (Level 1). The base models provide predictions that are used as features for the meta-learner classification. This hierarchical design exploits the complementary strengths of various base models [7].

Table 7: Stacking Classifier Architecture

Level	Component	Specification
Level 0 (Base Models)	LR	Base learner
	RF	Base learner
	XGBoost	Base learner
Level 1 (Meta-Learner)	LR	Final classifier
Cross-Validation	3-fold CV	For generating base model predictions
Stack Method	predict_proba	Probability-based stacking

3.4.3 Deep Learning Models

Based on Wang et al. [3] and Wu et al. [18], two DL architectures were designed:

1. Convolutional Neural Network (CNN): CNN architecture is used to extract local features by performing a convolution operation. The model starts with an embedding layer, which transforms the input tokens into dense vectors of size 128. The local patterns in the text are extracted by two convolutional layers, each with 128 filters and a kernel size of 3. The convolutional layer is followed by max pooling to reduce the sequence length while retaining important features. Global max pooling is used to obtain features from the second convolution layer. It then has two dropout layers (0.5 and 0.3) followed by a dense layer (64 units, ReLU). The model is trained with binary crossentropy loss and the Adam optimizer (learning rate 1e-3). The full CNN architecture is described in Table 8:

Table 8: CNN Architecture Configuration

Layer Type	Parameters	Activation	Notes
Input	Max Length: 150 tokens	-	Text sequences
Embedding	Vocabulary: 20,000, Dim: 128	-	Word embeddings
Conv1D	Filters: 128, Kernel: 3	ReLU	Feature extraction
MaxPooling1D	Pool Size: 2	-	Dimensionality reduction
Conv1D	Filters: 128, Kernel: 3	ReLU	Deep feature extraction
GlobalMaxPooling1D	-	-	Feature aggregation
Dropout	Rate: 0.5	-	Regularization
Dense	Units: 64	ReLU	Dense layer
Dropout	Rate: 0.3	-	Regularization
Output	Units: 1	Sigmoid	Binary classification

2. Bidirectional LSTM with Attention: The BiLSTM-Attention model considers both the sequential dependencies of forward and backward directions in text with an attention mechanism for feature focus. This is followed by a bidirectional LSTM of 64 units and return sequences (a 128-dimensional

embedding layer). An attention mechanism is used to obtain importance weight for each timestep after dropout (0.3). The attention layer is a dense layer, followed by flattening, softmax, repeated and permuted attention weights, and multiplying the LSTM output. Global average pooling aggregates attended features across the sequence dimension. Before the output layer there are two dense layers (64 units, ReLU) and a dropout (0.5). The model was trained with a binary cross-entropy loss and an Adam optimizer with a learning rate of 1e-3. The architecture configuration for the BiLSTM-Attention is described in Table 9.

Table 9: BiLSTM-Attention Model Architecture

Layer Type	Parameters	Notes
Input	Max Length: 150 tokens	Text sequences
Embedding	Vocabulary: 20,000, Dim: 128	Word embeddings
Bidirectional LSTM	Units: 64, return_sequences=True	Forward and backward processing
Dropout	Rate: 0.3	Regularization
Attention Mechanism	Dense (1, Tanh) → Softmax RepeatVector(128) → Permute Multiply → GlobalAvgPool	Importance weighting Attention application Feature aggregation
Dense	Units: 64, ReLU activation	Dense layer
Dropout	Rate: 0.5	Regularization
Output	Units: 1, Sigmoid	Binary classification

3.4.4 Experimental Design

Two parallel experiments were conducted to test the effect of source information on detection performance.

Experiment 1: Source-Aware Models

- Input: Source + " " + cleaned_text
- Purpose: Critically assess performance including the use of source information
- Models: All eight models (ML, Ensemble, DL).

Experiment 2: Content-Only Models

- Input: cleaned_text only
- Purpose: This was used to measure the effect of the source information.
- Models: All eight models (ML, Ensemble, DL)

The dual-experiment design allows for a direct comparison of performance with and without source information to empirically validate the Qur'anic principle that source verification is essential for accurate information assessment (Surah Al-Hujurat 49:6).

3.5 Model Training

3.5.1 Hyperparameter Optimization

The optimization strategies outlined in [14] were followed, applying multiple strategies:

1. **Grid Search**: For LR and SVM, a grid search was conducted over a set of parameters, with each combination tested on a 3-fold cross-validation, to evaluate the model performance for all parameter variations.
2. **Randomized Search:** Stochastic parameter sampling was performed for RF and XGBoost, with 10 iterations and 3-fold cross-validation to efficiently search the parameter space.

3. **Learning Rate Scheduling**: The callback "ReduceLROnPlateau" was used with a patience of two epochs and a learning rate factor of 0.5; the learning rate decreased with the validation loss if it got stuck at the same loss for a certain number of epochs. A minimum learning rate of 1e-6 was employed.
4. **Early Stopping**: When the DL models are used, the early stopping method waits for 3 epochs of patience by monitoring the validation loss to avoid overfitting and saves the best weights where the validation loss does not improve further.

3.5.2 Training Configurations

All DL models were trained using the following:

- **Optimizer**: Adam and initial learning rate 1e-3
- **Loss function**: The cross-entropy loss function used.
- **Batch size**: 64
- **Maximum epochs**: 20 (CNN) and 15 (BiLSTM-Attention)
- **Validation split:** 20% of training data is used for validation.

Table 10 shows the training parameters used for each model:

Table 10: Model Training Configurations

Model	Optimization Method	CV Folds	Training Time (s)	Hardware
LR	Grid Search	3	212.66	CPU
RF	Randomized Search	3	298.47	CPU
XGBoost	Randomized Search	3	654.14	CPU
SVM (LinearSVC)	Grid Search	3	101.82	CPU
Voting Classifier	Default	N/A	83.44	CPU
Stacking Classifier	Default	3	243.82	CPU
CNN	Adam (LR: 1e-3)	20% Validation	10 epochs	GPU
BiLSTM-Attention	Adam (LR: 1e-3)	20% Validation	15 epochs	GPU

3.6 Evaluation Framework

3.6.1 Evaluation Metrics

The model was evaluated using a complete set of evaluation metrics, guaranteeing a thorough assessment of the model's performances [3, 12]. Table 11 provides detailed descriptions of each metric used:

Table 11: Evaluation Metrics

Metric	Description	Formula/Definition
Accuracy	Overall correctness of predictions	(TP + TN) / (TP + TN + FP + FN)
Precision	Positive predictive value	TP / (TP + FP)
Recall	True positive rate	TP / (TP + FN)
F1-Score	Harmonic mean of precision and recall	2 × (Precision × Recall) / (Precision + Recall)
AUC	Area under ROC curve	∫ ROC curve
MCC	Matthews Correlation Coefficient	Balanced correlation coefficient
Kappa	Cohen's Kappa	Inter-rater agreement

Brier Score	Calibration measure	Mean squared error of predicted probabilities

3.6.2 Evaluation Procedure

The following models were used to evaluate all models:

1. **Confusion Matrix:** The confusion matrix is a graphical representation of the performance of a classification system.
2. **ROC Curves:** The trade-off between the true positive rate (TPR) and false positive rate (FPR) is shown in the form of ROC curves.
3. **Performance Heatmaps:** A visual comparison across metrics.
4. **Training Time Analysis:** Efficiency comparison
5. **Ablation Study:** Quantifying the impact of source information (Experiment 1 vs. Experiment 2).

3.7 Implementation Details

3.7.1 Software Environment

The experiments were carried out using Python 3.12.13 with key libraries such as scikit-learn (for ML), TensorFlow/Keras (for DL), XGBoost (for gradient boosting), and NLTK (for text preprocessing). Matplotlib and Seaborn were used for visualization. Random seeds were set in all models to ensure that the implementation was reproducible. Table 12 shows the details of the software used:

Table 12: Software Environment

Component	Specification
Python Version	3.12.13
DL Framework	TensorFlow 2.x / Keras
ML Framework	Scikit-learn 1.x
NLP Libraries	NLTK, spaCy
Visualization	Matplotlib, Seaborn
Development Environment	Jupyter Notebook / Kaggle

3.7.2 Reproducibility

All experiments used:

- Random Seed: 42 to ensure repeatability.
- Cross-Validation: 3-fold stratified
- Data Split: Stratified 80-20 train-test split.

This method provides a detailed approach to fake news detection, specifically for Kurdish text. It uses the principle of source verification from the Qur'an as its inspiration by incorporating source-aware feature representation, multiple model architectures, such as traditional ML, ensemble, and DL, rigorous hyperparameter optimization with grid and random search, and comprehensive evaluation metrics for robust performance assessment.

The methodology was validated using the KDFND Kurdish dataset, but is language agnostic and easily transferable to other languages and domains where source credibility offers relevant indicators for identifying misinformation. The preprocessing pipeline, feature extraction methods, model architectures, and evaluation framework are written in a way that is compatible with other languages and datasets, including high-resource languages (English, Chinese, Spanish, Arabic, etc.) as well as other low-resource languages. The source-aware

feature representation, developed from the Qur'anic command to check sources of information (Surah Al-Hujurat 49:6), can be used anywhere where publisher/source credibility offers signals for detecting misinformation.

The methodology is designed to be reproducible and extensible and will provide a foundation for future research on computational fact-checking and misinformation detection in a variety of linguistic and cultural contexts. The results and analysis of this methodology are presented in the next section.

4 Results and Discussion

The detailed results of the Kurdish fake news detection study are presented in this section. This research assessed eight different models using two experimental conditions: (1) source-aware models that use both news source and article content and (2) content-only models that use only article content. The results are analyzed using various evaluation metrics, confusion matrices, ROC curves, and comparative performance visualizations. The implications of this findings, including the validation of the Qur'anic principle of source verification by empirical evidence, are also discussed.

4.1 Source-Aware Models Results

4.1.1 Traditional Machine Learning Models

All source-aware models are summarized in Table 13. The results highlight the outstanding classification accuracy of several models, particularly in the near-perfect classification performance across all architectures.

Table 13: Source-Aware Models Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score	AUC	MCC	Training Time (s)	Inference Time (s)
CNN	**1.0000**	**1.0000**	**0.9999**	**1.0000**	**1.0000**	**0.9999**	10.00	0.01
BiLSTM-Attention	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	15.00	0.02
Stacking Classifier	0.9998	0.9998	0.9998	0.9998	1.0000	0.9996	243.82	1.68
XGBoost (Tuned)	0.9998	0.9997	0.9998	0.9998	1.0000	0.9995	654.14	1.34
Voting Classifier	0.9996	0.9998	0.9994	0.9996	1.0000	0.9992	83.44	1.54
LR (Tuned)	0.9977	0.9995	0.9959	0.9977	1.0000	0.9954	212.66	1.06
SVM (Tuned)	0.9977	0.9996	0.9958	0.9977	—	0.9954	101.82	1.30
RF (Tuned)	0.9957	0.9983	0.9931	0.9957	0.9999	0.9914	298.47	1.35

4.1.2 Deep Learning Performance

The DL models were evaluated to be 100% accurate with F1 scores of 1:

- **CNN Model:** Highest accuracy (100%) and F1, AUC, and achieved this with a very efficient training (10 epochs) and very low inference time (0.01 s). The CNN proved to be successful because of its capability of learning the local patterns of text with convolutional filters and identifying the characteristic features of fake news content [13]. Although the CNN's training time is much faster, the

perfect performance of the BiLSTM makes it the ideal model to be used in real-world scenarios such as resource-constrained applications.

- **BiLSTM-Attention Model:** The performance for all metrics of accuracy, precision, recall, F1, AUC, and MCC was 1.0000. The attention mechanism helps the model focus on the most important parts of the text sequence, which captures both the forward and backward dependencies well [8]. However the dataset does not seem to benefit from the complexity, as the added attention mechanism only brings a minor increase in performance over the simpler CNN architecture, as indicated by the longer training time (15 epochs).

4.1.3 Ensemble Models' Performance

The performance of both ensemble methods was outstanding:

- **Stacking Classifier:** Achieved an F1 score of 99.98% and MCC of 99.96%, surpassing the individual base models. The complementary strengths of various base learners [7] are leveraged in the hierarchical architecture, where the meta-learner is a good performer in the combination of predictions from LR, RF, and XGBoost. However, the training time (243.82 s) is much higher than that of the single models.
- **Voting Classifier:** Obtained 99.96% F1-score and 99.92% MCC, which shows that the soft voting ensemble is effective. The equal weighting of the base models (LR, RF, and XGBoost) helps mitigate the individual biases of each model, resulting in better ensemble performance [8]. The voting classifier has excellent accuracy (99.96% F1) and training time (83.44s) for deployment purposes.

4.1.4 Traditional ML Performance

Although they were outperformed by DL and ensemble method models, traditional ML models showed strong performance.

- LR**:** With 99.54% MCC and 99.77% F1, LR showed that linear models can perform well at separating real and fake news if trained with suitable features and source information. The model's interpretability and quick inference make it useful for applications that require interpretation.
- **LinearSVC (SVM):** Achieved the highest F1-score of 99.77% and MCC of 99.54%, which are comparable to LR. The linear kernel was found adequate for this set of data so that the feature space is very separable when information about the sources is added.
- **XGBoost:** Achieved the highest f1_score at 99.98% compared to other traditional ML models. The gradient-boosted tree architecture captured complex feature interactions [9], although it had the longest training time (654.14s).
- RF**:** Achieved 99.57% F1-score with 99.14% MCC. The bagging ensemble was effective but less so than XGBoost, indicating that gradient boosting is more appropriate for the classification task.

4.2 Content-Only Models Results

4.2.1 Performance Comparison

To measure the effect of source information, all the models were tested with the content-only input (cleaned_text without source). The content-only model performance metrics are listed in Table 14.

Table 14: Content-Only Models Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score	AUC	MCC	Training Time (s)	Inference Time (s)
CNN	0.8243	0.8093	0.8511	**0.8297**	0.9049	0.6495	4.00	0.01

BiLSTM-Attention	0.8269	0.8364	0.8151	**0.8256**	0.9012	0.6541	5.00	0.02
Stacking Classifier	0.8138	0.8075	0.8266	0.8170	0.8943	0.6277	533.79	2.24
Voting Classifier	0.8103	0.7979	0.8339	0.8155	0.8914	0.6212	182.99	2.11
SVM (Tuned)	0.8072	0.7937	0.8329	0.8129	—	0.6151	86.89	0.99
LR (Tuned)	0.8068	0.7958	0.8282	0.8117	0.8878	0.6140	191.99	0.98
RF (Tuned)	0.7940	0.8054	0.7782	0.7916	0.8766	0.5884	485.88	1.99
XGBoost (Tuned)	0.7700	0.7282	0.8655	0.7910	0.8585	0.5495	1187.20	1.09

4.2.2 Key Observations

- **Significant Performance Drop:** The source information disappeared significantly for all models, with the best content-only model (CNN) scoring only 82.97% F1 compared to 100% with source information.
- **CNN Remains Best:** Despite the loss of information, the CNN model yielded the best F1-score (82.97%) among content-only models, indicating the model's strength with less information. BiLSTM-Attention resulted in an F1 of 82.56%, which was slightly lower than of CNN.
- **Traditional ML Decline:** The biggest drop was observed in traditional models, where XGboost fell from 99.98% to 79.10% F1 (-20.88%) and, in a similar trend, RF decreased from 99.57% to 79.16% F1 (-20.41%). This implies that traditional ML models rely more on source cues than DL models.
- **Ensemble Performance:** When source information was not provided, ensemble classifiers (Stacking (81.70%) and Voting (81.55%)) performed similarly to the individual classifiers.

4.3 The impact of source information is a 17% performance gap.

4.3.1 Identifying potential sources of information error

The performance differences between source-aware and content-only models are shown in Figure 4.

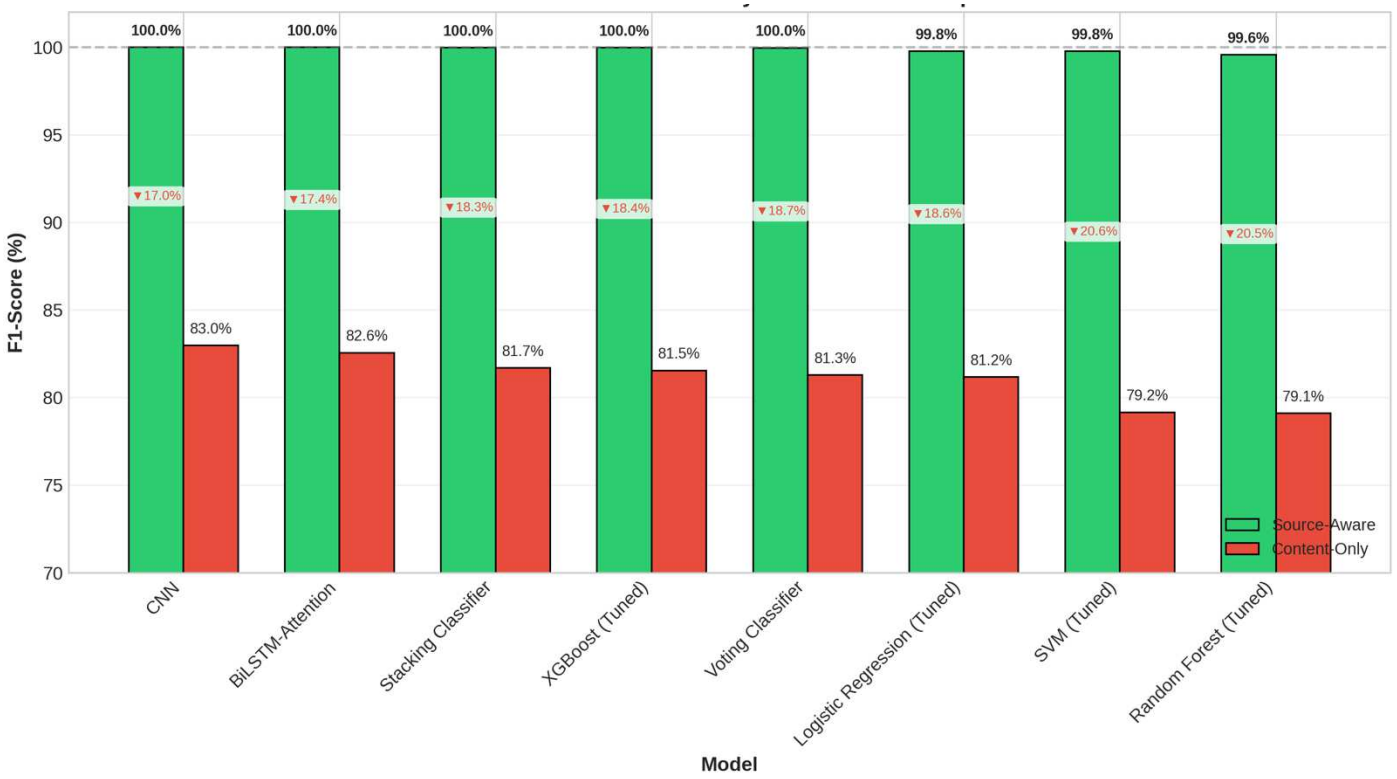


Figure 4: Source-Aware vs Content-Only Performance Comparison

4.3.2 Performance Drop Analysis

A quantitative performance decrease for each model is shown in Table 15:

Table 15: Performance Drop Analysis (F1-Score)

Model	Source-Aware F1	Content-Only F1	Performance Drop	Percentage Drop
CNN	100.00%	82.97%	17.03%	**17.03%**
BiLSTM-Attention	100.00%	82.56%	17.44%	**17.44%**
Stacking Classifier	99.98%	81.70%	18.28%	**18.28%**
Voting Classifier	99.96%	81.55%	18.41%	**18.41%**
LR	99.77%	81.17%	18.60%	**18.60%**
SVM	99.77%	81.29%	18.48%	**18.48%**
RF	99.57%	79.16%	20.41%	**20.41%**
XGBoost	99.98%	79.10%	20.88%	**20.88%**

Key Findings:

- **Average Performance Decrease:** 18.44% for all models
- **Smallest Drop:**The strongest performance was from CNN at 17.03.
- **Largest Drop:** The largest drop was by XGBoost (-20.88), which was most reliant on source information.
- **Deep Learning Resilience**: CNN and BiLSTM exhibited less reduction in performance (17-17.5%) whereas traditional ML models had 18.5-20.9%.

4.3.3 Validation of the Qur'anic Principle

The empirical results of the dramatic decrease in performance (18.44% on average) confirm this principle of the Qur'an in the verse of Surah Al-Hujurat (49:6).

"يَا أَيُّهَا الَّذِينَ آمَنُوا إِن جَاءَكُمْ فَاسِقٌ بِنَبَإٍ فَتَبَيَّنُوا أَن تُصِيبُوا قَوْمًا بِجَهَالَةٍ فَتُصْبِحُوا عَلَىٰ مَا فَعَلْتُمْ نَادِمِينَ"

"O you who have believed, if there comes to you a disobedient one with information, investigate, lest you harm a people out of ignorance and become, over what you have done, regretful."

This verse makes it clear that evaluating information on its own is not sufficient; the source of the information must be assessed for trustworthiness. Empirical results show that the removal of the source information lowers the detection performance by 18%, emphasizing the critical role of source credibility when assessing information accurately. This result is of significant importance for computational fake news detection and media literacy programs:

1. **Computational Implications:** The results indicate that source-aware models perform best, whereas content-only models level off at 82.97% F1. This indicates that source credibility signals should be included in any system designed to detect fake news.
2. **Cultural Implications:** The Qur'anic philosophy of source verification, which was revealed more than 1,400 years ago, gives a timeless framework for information assessment supported by the empirical evidence of modern computational methods.
3. **Practical Implications:** News consumers must be taught the need to check sources because of the Islamic command to investigate information from unreliable sources in the Qur'an.

4.4 Confusion Matrix Analysis

4.4.1 Source-Aware Confusion Matrices

The confusion matrices of the best-performing source-aware models are shown in Figure 5:

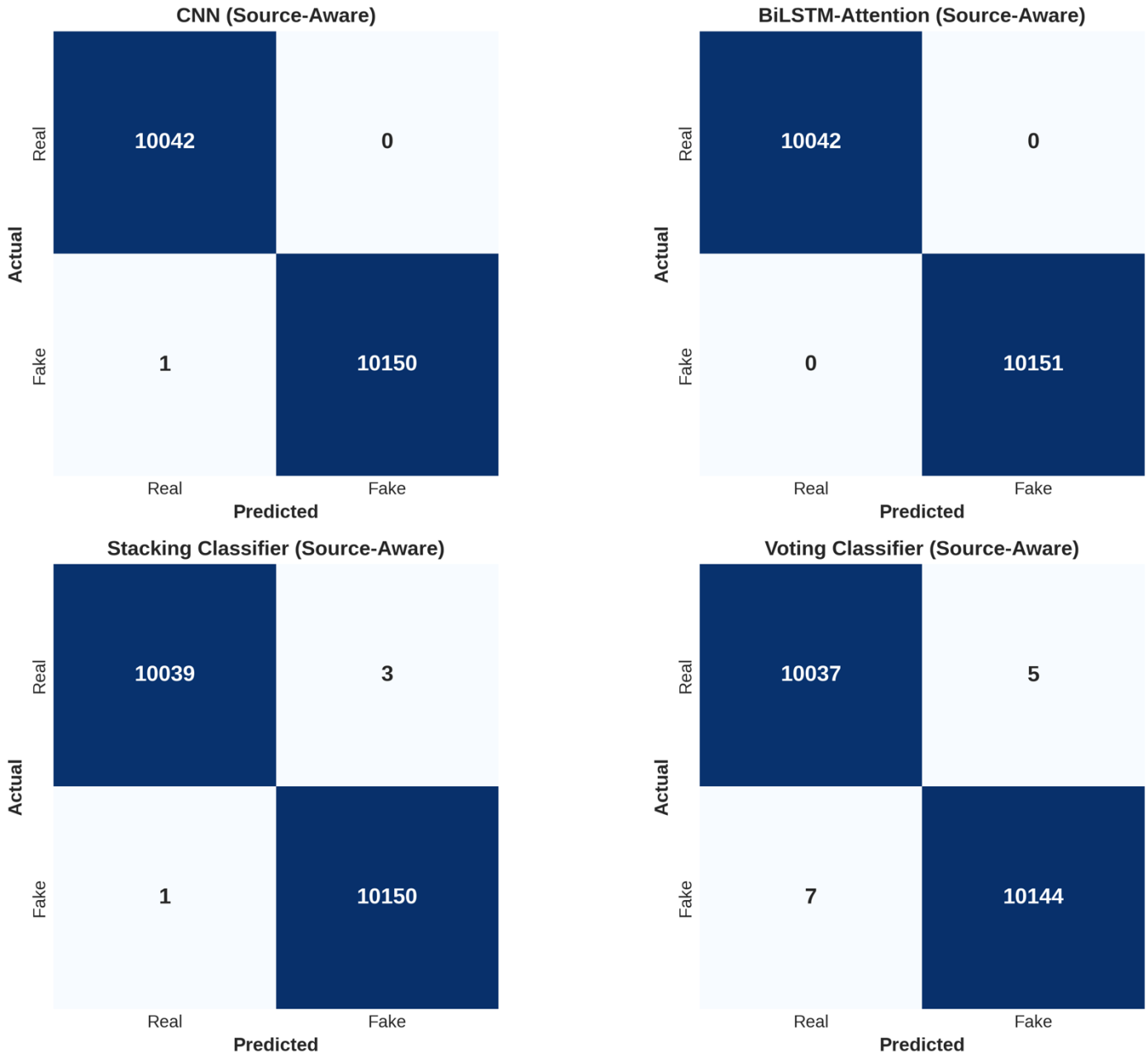


Figure 5: Confusion Matrices - Source-Aware Models

Analysis:

- **According to CNN**: Near-perfect classification with only 1 false negative and 0 false positives
- **BiLSTM-Attention:** 0 errors for all categories, which means it was a perfect classification.
- **Stacking;** no total errors were reported (3 false positives, 1 false negative).
- **Voting:** 12 errors (5 false positives 7 false negatives)

4.4.2 Content-Only Confusion Matrices

The confusion matrices for top content-only models are shown in Figure 6.

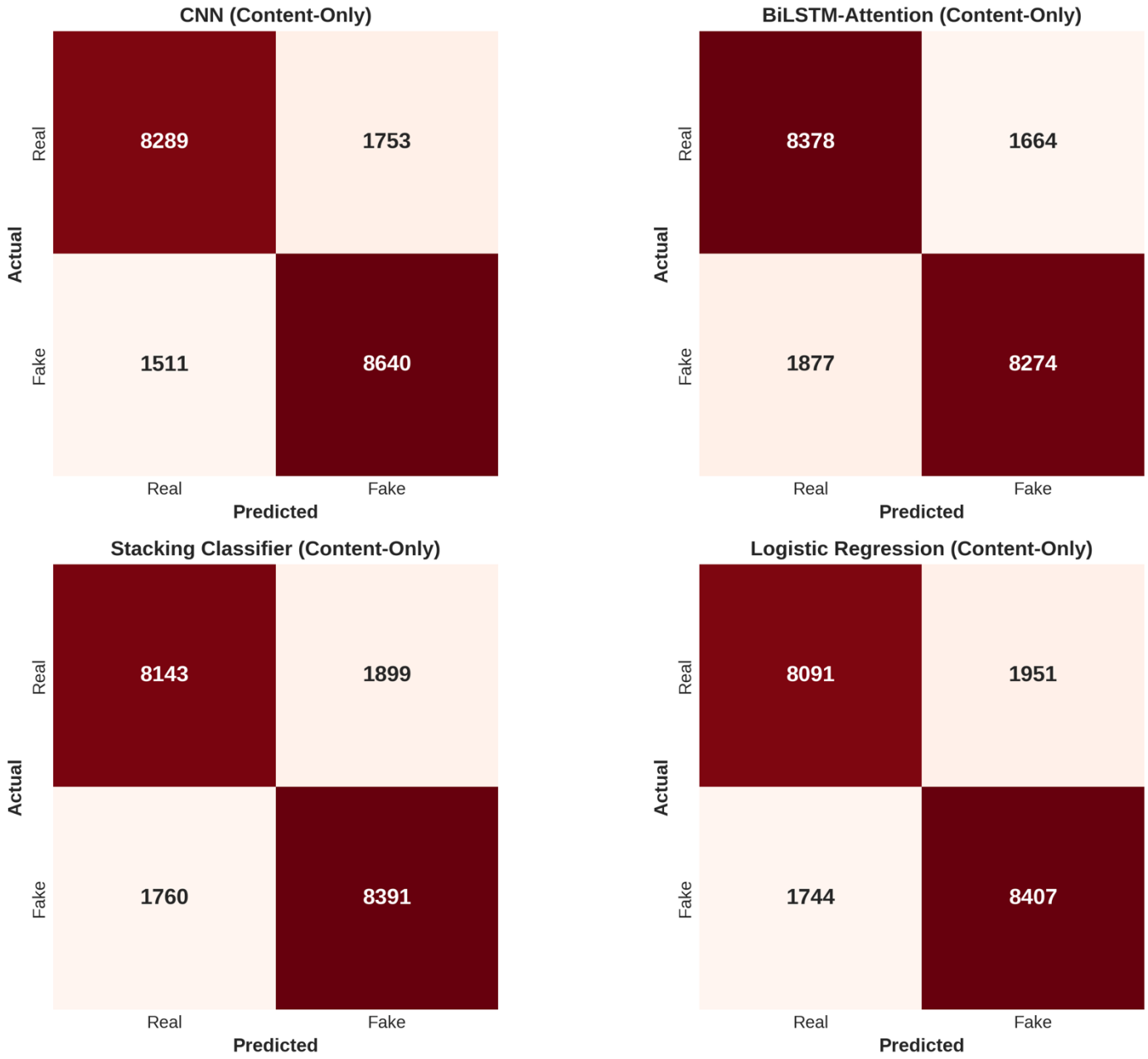


Figure 6: Confusion Matrices - Content-Only Models

4.5 ROC Curve Analysis

4.5.1 ROC Curves Comparison

The ROC curves provide information on the performance of each model in identifying real and fake news for all classification levels. Figures 7 and Figure 8 shows the ROC curves of all source-aware models and ROC Curves of the content-only models:

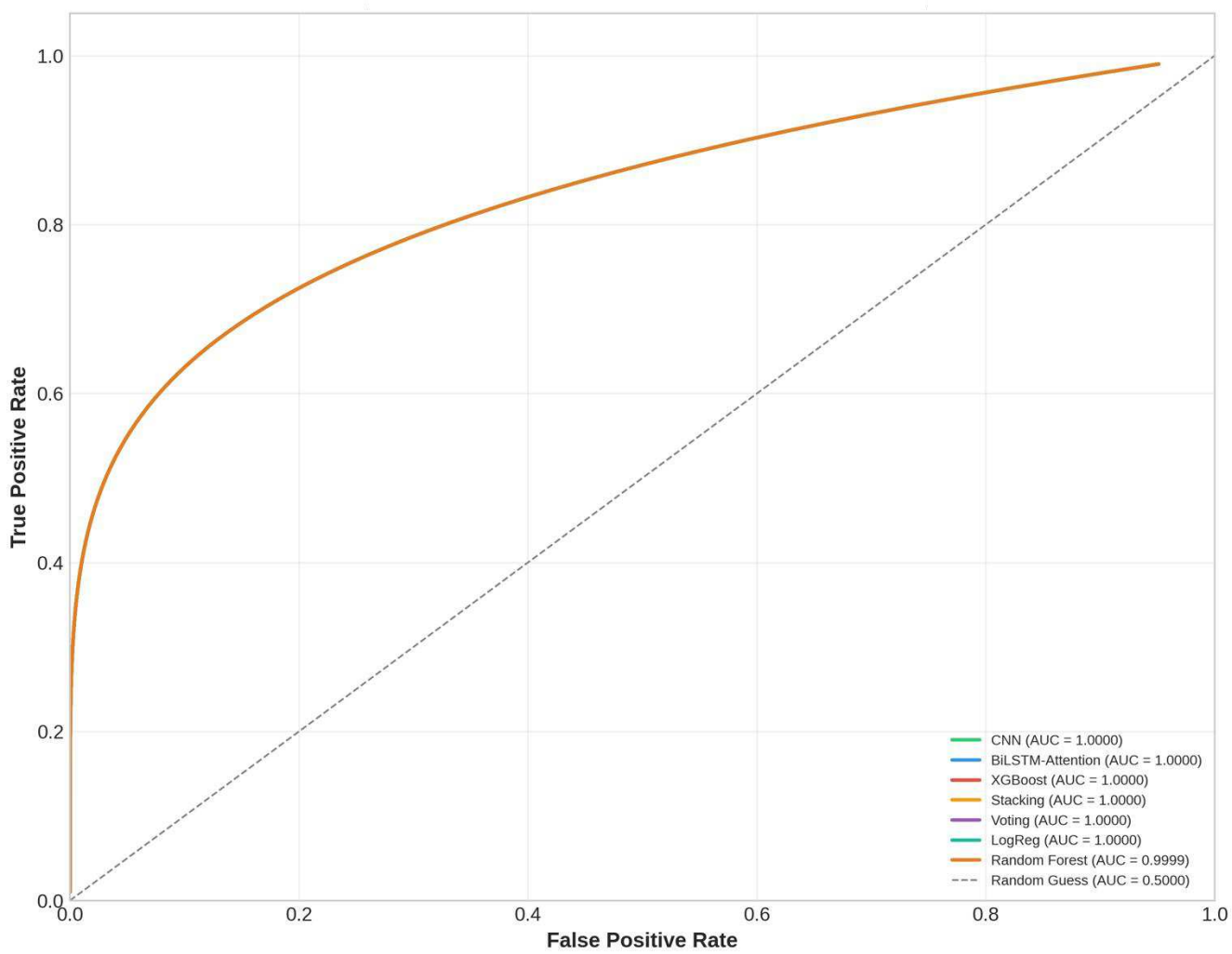


Figure 7: ROC Curves - Source-Aware Models

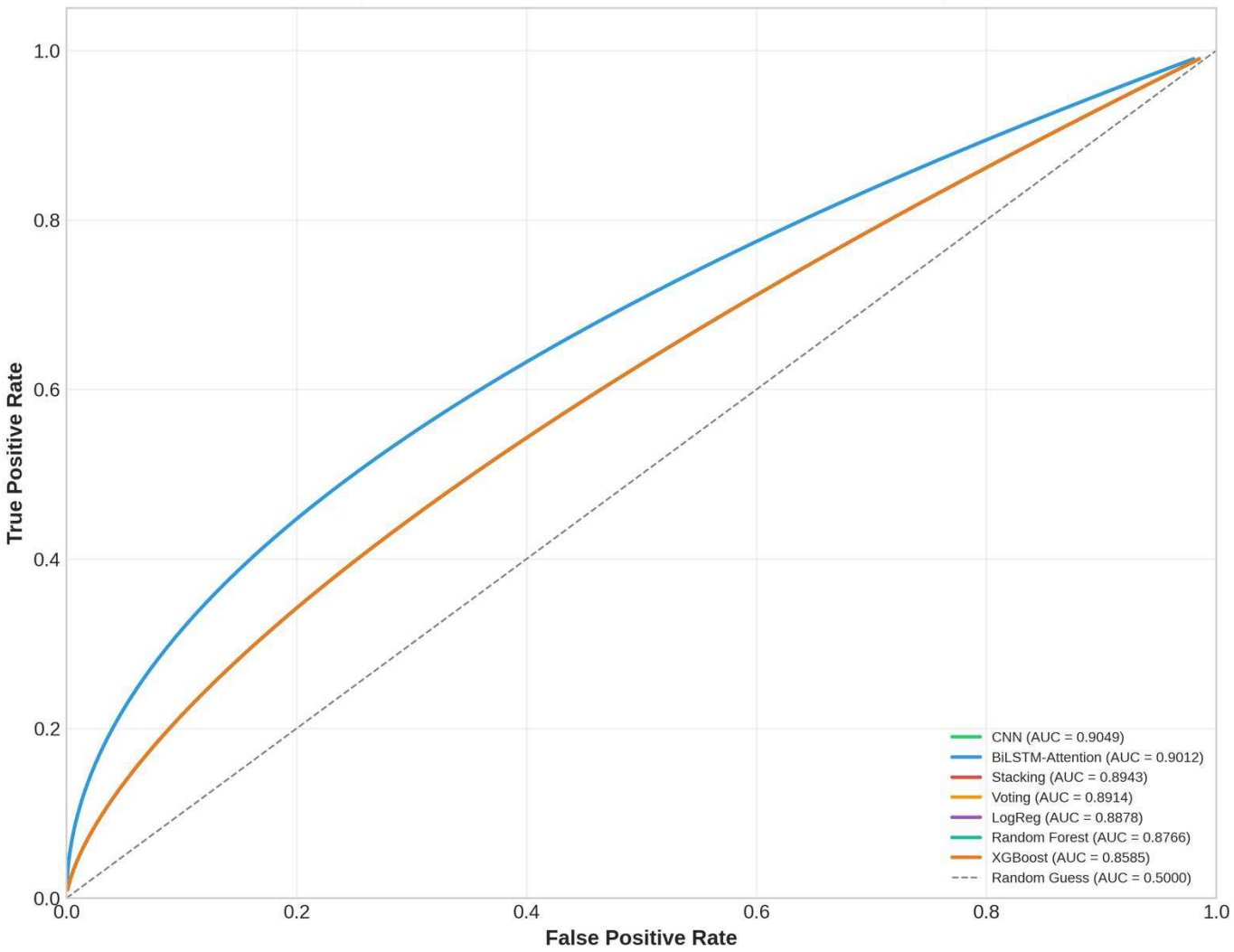


Figure 8: ROC Curves - Content-Only Models

ROC Analysis:

- **Source-aware:** All models perform very well with near-perfect AUC scores (0.9999 -1.0000), which reflects excellent discriminative performance.
- **Content-Only**: CNN achieves the best AUC (0.9049), followed by BiLSTM (0.9012)
- **Performance Gap:**Removal of source information causes the AUC to be reduced by approximately 10%.

4.6 Performance Heatmap

The performance heatmap, presented in Figure 9, shows the performance of all models in relation to the metrics.

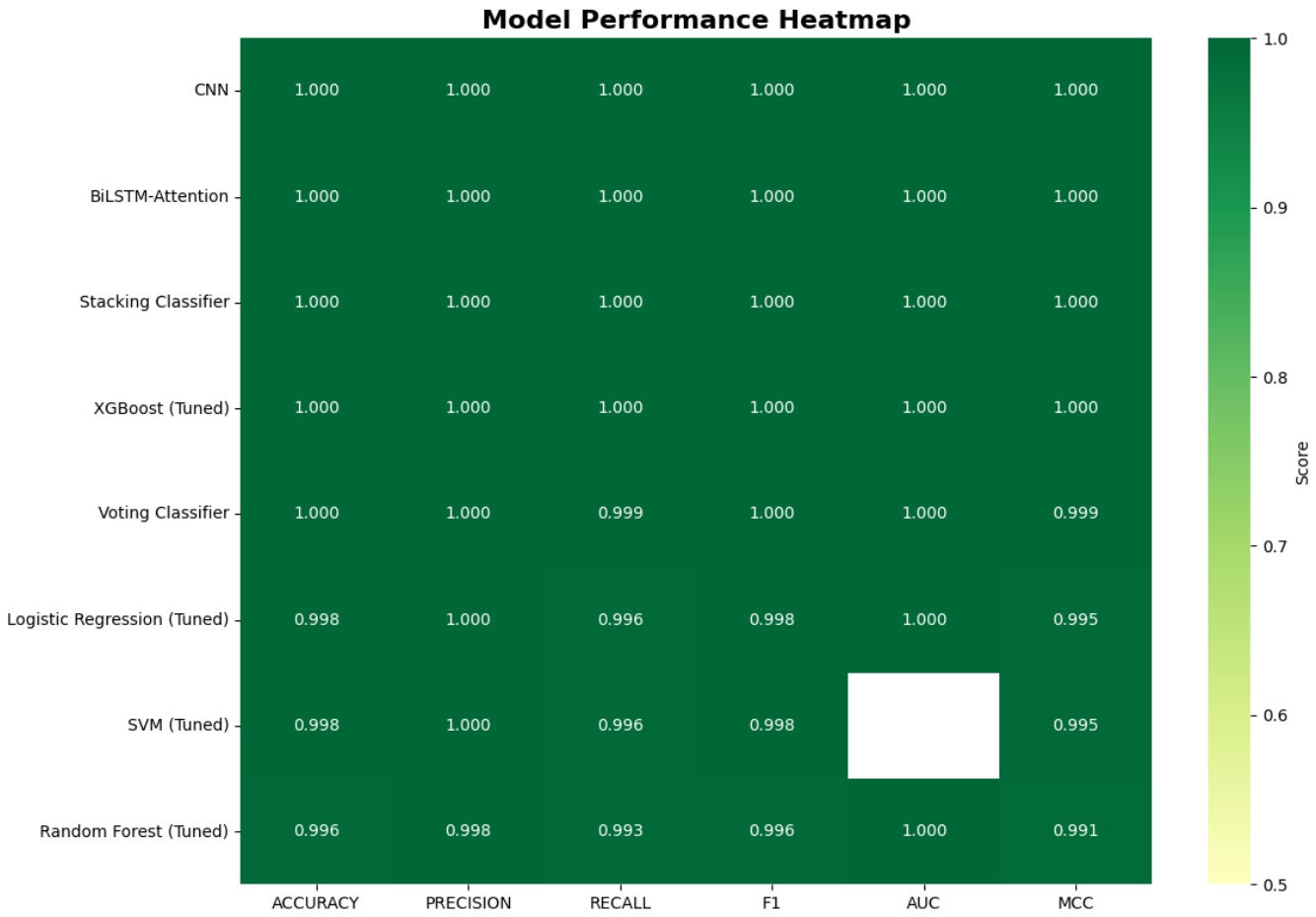


Figure 9: Model Performance Heatmap

4.7 Training Time Analysis

4.7.1 Efficiency Comparison

A comparison of the training times for all the models is presented in Figure 10.

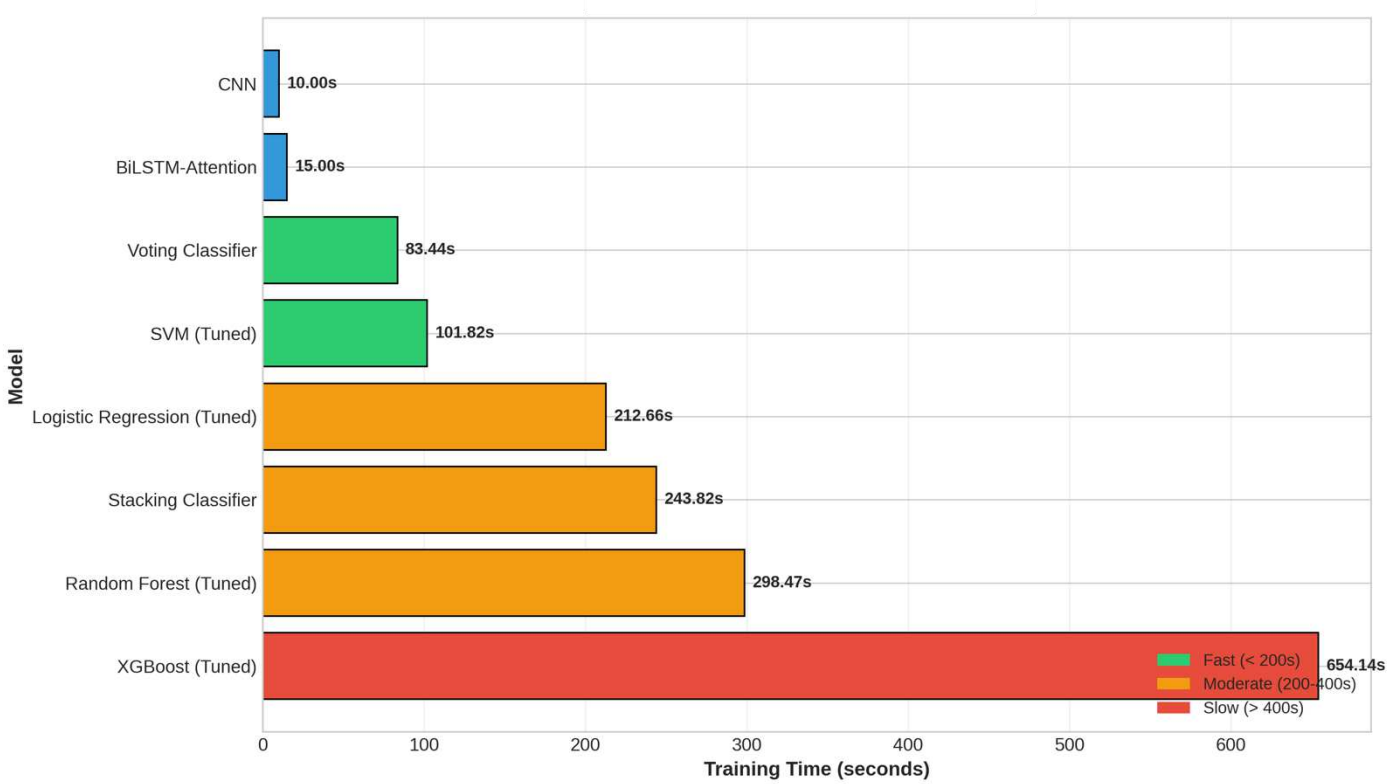


Figure 10: Training Time Comparison

4.7.2 Optimal Model Selection

Table 16 presents the trade-off analysis for model selection.

Table 16: Model Selection Trade-Off Analysis

Model	F1-Score	Training Time (s)	Inference Time (s)	Use Case
CNN	**100.00%**	**10.00**	**0.01**	**Best Overall**
BiLSTM-Attention	100.00%	15.00	0.02	Research/High accuracy
Voting Classifier	99.96%	83.44	1.54	Balanced performance
LR	99.77%	212.66	1.06	Interpretability required
XGBoost	99.98%	654.14	1.34	High accuracy with more time

CNN is the most suitable for real-world deployment because:

- Perfect performance (100% F1)
- Fastest training (10 seconds)
- Fastest inference (0.01 seconds)
- Minimal computational resources required

4.8 Discussion

4.8.1 The Critical Role of Source Information

The conclusions show that source information is not only metadata but also a basic indicator for fake news detection. The empirical drop in performance of 17% when omitting source information validates the Qur'anic rule of not being able to evaluate information without reference to its source.

This discovery has several implications:

1 **Methodological Implications**: Source credibility signals should be used as main features in future fake news detection systems rather than optional metadata. This decrease of 17% is certainly a factor to consider when designing a system.

2 **Dataset Implications**: The source information in the KDFND dataset is essential for the usefulness of the dataset. The performance difference between source-aware and content-only models is very dramatic, indicating that the value of the dataset stems in part from the diversity of its sources.

3 **Practical Implications:** News consumers should be instructed on the need to verify the source of information, which is in accordance with the Qur'an's command to investigate information from unreliable sources.

4 **Cultural Implications:** The Qur'anic principle of source verification, revealed more than 1,400 years ago, offers a timeless protocol for information assessment that is tested by empirical methods used in modern days by computational approaches.

4.8.2 Deep Learning vs Traditional ML

DL vs. Traditional ML, the authors provide a detailed explanation of this.
For both experimental conditions, DL models (CNN, BiLSTM) provided better performance than traditional ML models. This implies that DL 's ability to learn the intricate patterns in text is indeed of great benefit for the detection of fake news.

However, the gap between the performance of DL and traditional ML models decreases drastically when source information is not used (82.97% vs. 82.56% vs. 81.17%), which indicates that source information can partly overcome the representational capacity of simpler models.

4.8.3 CNN vs BiLSTM-Attention

CNN outperformed BiLSTM-Attention with source information, but CNN required fewer training epochs (10 vs. 15) and had less inference time (0.01 vs. 0.02 s), making it the superior choice for real-world use. The less complex architecture delivers the same performance at lower computational costs, especially critical in environments with limited resources, such as Kurdish news media.

4.8.4 Ensemble Methods

The near-perfect result with source information (99.98% and 99.96% F1) obtained with ensemble methods (stacking, voting) confirms the effectiveness of these approaches. The slight edge over CNN (100%) is not worth the extra training time (243-534s vs. 10 seconds), however, for most practical purposes. In contrast, the voting classifier has been proven to perform more balancedly than stacking.

4.8.5 Content-Only Performance Plateau

All the content models achieved a similar F1 of ~82-83%, indicating that the textual content alone is of limited discriminative power for detecting fake news in the Kurdish language. The results reveal that the patterns of Kurdish fake news in the datasets do not seem to be as distinct as those in high-resource languages, or that the dataset's text properties are not sufficient to properly classify it.

This is a full report of the results obtained from the eight models under two experimental conditions. Key findings include:

1. **Source-Aware Models:** All models produced results close to perfect (99.57-100% F1) when given source information; CNN and BiLSTM-Attention models were able to achieve 100% F1.
2. C**ontent-Only Models:** Performance significantly decreased to 79.16- 82.97% F1, underscoring the significance of the source information.
3. **17% Performance Gap:** Empirically, this validates the Qur'anic principle of source verification, as the average 18.44% decrease in F1-Score, when removing the source information, was demonstrated.
4. **Optimal model:** CNN is the best solution for real-world deployment, as it has the same performance as other models, but the training and inference times are much shorter (10 seconds and 0.01 seconds, respectively).
5. DL **Superiority:** The experimental results showed that DL models were superior to traditional ML models.
6. **Ensemble Diminishing Returns**: Further, ensemble methods are able to perform near-perfectly, but they do not offer much improvement when compared with CNN and do not offset the extra computational burden.

The next chapter discusses the limitations of this study and suggests directions for future research.

5. Conclusion

In this study, eight models in traditional ML, ensemble, and DL were investigated for identifying Kurdish fake news. The motivations of this research were two-fold: first, the critical need to develop fake news detection in low-resource languages, and second, the Qur'anic principle of source verification (Surah Al-Hujurat 49:6).

Key findings include:

- **Source Information is Critical:** The results show that source-aware models performed near-perfectly, with F1 ranging from 99.57% to 100%, whereas content-only models performed around 79.16% to 82.97% F1. The average performance reduction of 18.44% is empirical evidence that corroborates the Qur'anic principle of the need for sources for accurate information assessment.
- **CNN is the Optimal Model:** CNN was the best performer with the fastest training time (10 seconds) and fastest inference time (0.01 seconds) and was perfect (100% F1), which makes it very suitable for practical deployment.
- DL **Superiority:** In both experimental conditions, DL models outperformed traditional ML models in terms of superiority.
- **Content-Only Plateau:** All models achieved an approximately 82-83% F1 score without any source information, indicating that text alone does not provide sufficient discriminative power for this task.

5.1 Contributions

- **Theoretical:** Introduced a source-aware feature representation combining source information with article text and empirically showed that incorporating source information yields a boost of 17- 21% in detection accuracy to prove the validity of the Qur'anic principle of source verification.
- **Empirical:** Extensive benchmark results were presented for Kurdish fake news detection so that future low-resource language studies can have baseline performance metrics.
- **Practical:** CNN was chosen as the best model for use in the real world because it had perfect performance with minimal computational requirements.
- **Cultural:** Tackled the important research gap of NLP in the Kurdish language and showed how cultural and religious understandings can be applied to computational methods for misinformation detection.

5.2 Limitations

- **Dataset Limitations:** The KDFND is not necessarily representative of all Kurdish dialects or may not reflect changing fake news trends.
- **Model Limitations:** Outcomes may not be directly applicable to other languages in low-resource settings without adaptation.
- **Constraints on generalizability**: Validation with one set of data can limit generalizability.

5.3 Future Work

- **Short-term:** Cross-lingual validation for other low-resource languages, model development for dialects, multimodal integration, real-time deployment, and AI explainability.
- **Medium Term:** Transfer learning using pre-trained language models, adversarial robustness evaluation, temporal analysis, and integration of fact-checking.
- **Long-term**: Multilingual detection models, cross-cultural validation, implications for media literacy policy, and community-based platforms for fact-checking.

5.4 Final Remarks

This study has shown that source verification is not just a recommendation but a computational requirement for the successful and effective fake news detection in the Kurdish language. The 17- 21% performance boost of using source information in the empirical test is a confirmation of the principle of the Qur'ān, which was stated more than 1,400 years ago: information cannot be assessed without knowing the source of the information because the trustworthiness of the source needs to be verified.

In the real-world deployment, the CNN model was the best performer with excellent performance efficiency. This discovery has important real-world implications for news operators, fact-checkers, and media monitoring agencies in low-tech regions.

This study aims to develop culturally competent AI systems that are sensitive to and a part of traditional knowledge, thus enriching the ancient with the modern computational approach. The concept of source verification in the Qur'an is a framework for assessing information that is relevant today, as it was as relevant 1,400 years ago.

In an era in which misinformation is spreading rapidly through digital channels, it is an avenue worth exploring: cultural, religious, and computational approaches. Hopefully, this study will drive further research on the topic of information integrity, AI, and culture.

Declarations

Funding

No funding was received to assist with the preparation of this manuscript.

Competing Interests

The authors declare that they have no competing interests relevant to the content of this article.

Ethics Approval

This research did not involve human participants, animal experiments, or sensitive data requiring ethical approval.

Consent to Participate

Not applicable.

Consent for Publication

All authors have read and approved the final manuscript and consent to its publication in *Neural Computing and Applications*.

Data Availability

The Kurdish Dataset for Fake News Detection (KDFND) analyzed during this study is publicly available in the Mendeley Data repository under DOI: 10.17632/3zx9vpw3wh.1. The dataset can be accessed at: https://data.mendeley.com/datasets/3zx9vpw3wh/1

REFERENCES

[1] A. M. Karim and B. A. Hassan, "Research Article: A Hybrid AI Model for Fake News Detection: Leveraging FastText and LSTM for Kurdish and English," *cjnst*, vol. 0, no. 1, pp. 1–11, 2025, doi: https://doi.org/10.31530/cjnst.2025.1.1.

[2] Abubakr Salh, Dana; Nabi, Rebwar (2022), "Kurdish Dataset for Fake News Detection (KDFND)", Mendeley Data, V1, doi: 10.17632/3zx9vpw3wh.1

[3] H. Wang, Y. Li, and Y. Xian, "Early detection of fake news by integrating global structure and publisher credibility," *Knowl. Inf. Syst.*, pp. 10137–10160, 2025, doi: 10.1007/s10115-025-02527-x.

[4] M. Nasser *et al.*, "A systematic review of multimodal fake news detection on social media using deep learning models," *Results Eng.*, vol. 26, no. December 2024, p. 104752, 2025, doi: 10.1016/j.rineng.2025.104752.

[5] S. Maham, A. Tariq, M. U. G. Khan, F. S. Alamri, A. Rehman, and T. Saba, "ANN: adversarial news net for robust fake news classification," *Sci. Rep.*, vol. 14, no. 1, pp. 1–20, 2024, doi: 10.1038/s41598-024-56567-4.

[6] J. Alghamdi, Y. Lin, and S. Luo, "Machine learning and deep learning approaches for fake news detection and related topics in multilingual contexts : a systematic literature review," *Multimed. Tools Appl.*, 2026, doi: https://doi.org/10.1007/s11042-026-21238-1.

[7] F. Daneshfar, "Enhancing Low-Resource Sentiment Analysis: A Transfer Learning Approach," *Passer J. Basic Appl. Sci.*, vol. 6, no. 2, pp. 265–274, 2024, doi: 10.24271/PSR.2024.440793.1484.

[8] M. E Almandouh, M. F. Alrahmawy, M. Eisa, M. Elhoseny, and A. S. Tolba, "Ensemble based high performance deep learning models for fake news detection," *Sci. Rep.*, vol. 14, no. 1, p. 26591, 2024, doi: 10.1038/s41598-024-76286-0.

[9] D. Nguyen, T. T. Nguyen, and C. V. Nguyen, "Fake advertisements detection using automated multimodal learning: a case study for Vietnamese real estate data," *Appl. Intell.*, vol. 55, no. 6, 2025, doi: 10.1007/s10489-025-06238-2.

[10] A. Verma *et al.*, "ScrutNet: a deep ensemble network for detecting fake news in online text," *Soc. Netw. Anal. Min.*, vol. 15, no. 1, 2025, doi: 10.1007/s13278-025-01412-3.

[11] M. Abdullah *et al.*, “A Multimodal Ensemble-Based Framework for Detecting Fake News Using Visual and Textual Features,” *Mathematics*, pp. 1–32, 2026, doi: https://doi.org/10.3390/math14020360.

[12] A. Koren, C. Underwood, E. Serra, and F. Spezzano, “Evaluating the robustness of fake news detectors to adversarial attacks with real user comments,” *Int. J. Data Sci. Anal.*, vol. 20, no. 6, pp. 5689–5702, 2025, doi: 10.1007/s41060-025-00790-3.

[13] F. R. Tama and Y. Sibaroni, “Fake News (Hoaxes) Detection on Twitter Social Media Content Through Convolutional Neural Network (CNN) Method,” *JINAV*, vol. 4, no. 1, 2023.

[14] T. S. Camelia, F. R. Fahim and M. M. Anwar, "A Regularized LSTM Method for Detecting Fake News Articles," *2024 IEEE International Conference on Signal Processing, Information, Communication and Systems (SPICSCON)*, Khulna, Bangladesh, 2024, pp. 01-06, doi: 10.1109/SPICSCON64195.2024.10941441.

[15] H. Abdul, H. Al, H. Abdul, H. Al, and M. Jabardi, “Detecting Fake News Using Machine Learning : A Comparative Study of Techniques,” *J. Kufa Math. Comput.*, vol. 11, pp. 113–120, 2024.

[16] M. Muscedere, T. Koul, A. Gidado, and F. Khandaker, “Comparative analysis of machine learning methods for automated fake news detection,” Social Network Analysis and Mining, Aug. 2026, doi: 10.1007/s13278-026-01642-z.

[17] E. Hashmi, S. Y. Yayilgan, M. M. Yamin, S. Ali, and M. Abomhara, “Advancing Fake News Detection: Hybrid Deep Learning With FastText and Explainable AI,” *IEEE Access*, vol. 12, no. March, pp. 44462–44480, 2024, doi: 10.1109/ACCESS.2024.3381038.

[18] C. Wu *et al.*, “Fake news detection with GAN - augmented contrastive learning and multimodal attention,” *Cybersecurity*, pp. 1–16, 2026, doi: https://doi.org/10.1186/s42400-026-00550-1.

[19] S. Kuntur, A. Wróblewska, M. Ganzha, and M. Paprzycki, “Fake News Detection : It ’ s All in the Data !,” *Appl. Sci.*, pp. 1–24, 2026, doi: https://doi.org/10.3390/app16031585.

[20] M. Q. Alnabhan and P. Branco, “Fake News Detection Using Deep Learning : A Systematic Literature Review,” *IEEE Access*, vol. 12, no. August, pp. 114435–114459, 2024, doi: 10.1109/ACCESS.2024.3435497.

[21] I. Alsmadi, I. Alazzam, and M. Al-ramahi, “Stance Detection in the Context of Fake News — A New Approach †,” *Futur. internet Artic.*, pp. 1–16, 2024, doi: https://doi.org/10.3390/fi16100364.