IJEMD-CSAI, 4 (1) (2025) https://doi.org/10.54938/ijemdcsai.2025.04.1.545

International Journal of Emerging Multidisciplinaries:
Computer Science and Artificial Intelligence

Research Paper
Journal Homepage: www.ojs.ijemd.com
ISSN (print): 2791-0164 ISSN (online): 2957-5036

Enhancing Fake News Detection Models Through *Modified Cosine Similarity* (MCS) Using *Sorensen-Dice Distance*

Aliyu Shuaibu [1*], Hadiza Ali Umar[2] and Yusuf Tijjani Baffa[2]

1. *Department of Computer Science, Faculty of Computing, Bayero University Kano, Nigeria*
2. *Department of Computer Science, Faculty of Computing, Bayero University Kano, Nigeria*
3. *Department of Software Engineering, Faculty of Computing, Bayero University Kano, Nigeria*

Abstract

The rapid spread of fake news undermines public trust and highlights the need for more reliable detection models. Traditional approaches, such as LSTM with standard cosine similarity using Euclidean distance, often fail to capture subtle textual relationships. This study introduces a Modified Cosine Similarity (MCS) that replaces Euclidean distance with Sørensen-Dice distance and evaluates its effectiveness across four models: Long Short-Term Memory (LSTM), K-Nearest Neighbor (KNN), Random Forest (RF), and Support Vector Machine (SVM). Baseline results using cosine similarity showed strong performance, with SVM achieving the highest accuracy (0.987) and F1-score (0.986), followed by RF (accuracy 0.979) and KNN (accuracy 0.973). However, enhanced models with MCS demonstrated substantial improvements. LSTM achieved the best results overall (accuracy 0.997, recall 0.998, F1-score 0.997) with reduced cross-entropy loss (0.016), false positive rate (0.005), and false negative rate (0.002). SVM and KNN also showed notable gains with accuracies of 0.995 and 0.991, respectively, while RF recorded high recall (0.995) and competitive performance across metrics. These findings confirm that integrating Sørensen-Dice distance into cosine similarity significantly boosts semantic representation and model performance, making MCS a robust similarity measure for advancing fake news detection.

Keywords- *Euclidian distance; Misinformation; Machine Learning Classifiers; Natural Language Processing; Sørensen-Dice distance*

***Email address*: aliyu2017@gmail.com (Aliyu Shuaibu[1+])**

Introduction

The spread of false information has long been a challenge, initially disseminated through verbal communication as gossip or propaganda for political, financial, or social gain. In the digital age, the rapid proliferation of fake news has emerged as a major concern, undermining public trust and influencing societal stability [1]. Social media platforms have exacerbated this issue by enabling the widespread and rapid dissemination of misinformation. To address this challenge, the development of effective fake news detection techniques has become a priority in natural language processing (NLP) [2]. Among the various machine learning approaches, Long Short-Term Memory (LSTM) networks have shown significant promise. LSTMs, a type of recurrent neural network (RNN), effectively capture long-term dependencies in sequential text data, enabling them to detect subtle contextual relationships that may indicate deceptive content [3].

In addition to machine learning techniques, similarity and distance measures play a crucial role in NLP tasks, such as text classification, information retrieval, and document clustering [4]. Cosine similarity is widely used for measuring text similarity as it calculates the angle between two document vectors, making it robust for documents of varying lengths [5]. Another popular similarity is Sorensen-Dice or Dice Coefficient which is a statistical measure utilized to gauge the similarity between two sets of elements. It provides a means to quantify the extent of overlap or similarity between the elements present in the two sets [6]. Furthermore, Sorensen-Dice is one of the most widely use metric in set theory and information retrieval, particularly for measuring the similarity/dissimilarity between two sets [9].

Similarity and distance measures are vital in natural language processing (NLP) tasks such as text classification, information retrieval, and document clustering. Cosine similarity, which calculates the angle between document vectors, is robust for texts of varying lengths [8]. Sorensen-Dice distance is commonly used in set theory and information retrieval, with the former quantifying the overlap between two sets and the latter measuring their dissimilarity[9]. These metrics have proven effective in various applications, such as combining sentiment analysis with machine learning classifiers for fake news detection [10].

Recent advancements in fake news detection emphasize the integration of machine learning and deep learning techniques with text similarity measures [11]. Models leveraging Cosine similarity have demonstrated success in healthcare misinformation, exaggerated title detection, and multilingual evidence classification [12]. However, relying on Euclidean distance for Cosine similarity introduces limitations, particularly with sparse, high-dimensional text data [13]. In contrast, Sorensen-Dice distance is more suitable for such contexts, focusing on shared attributes between sets [14]. These findings highlight the potential for improving fake news detection by addressing limitations in existing similarity measures.

Considering the aforementioned researches, the researchers used these similarity and distance measures independently leading to some significant limitations. Cosine similarity rely on Euclidean distance which introduces limitations, particularly when handling high-dimensional and sparse text data, as Euclidean distance fails to account for vector overlap and is sensitive to variations in

magnitude [14]. Sorensen-Dice distance, on the other hand, focuses on the proportion of shared attributes between sets, making it a more suitable alternative for sparse data contexts [15].

In this research, we proposed a novel approach of developing an improved similarity measure by replacing Euclidean distance with Sorensen-Dice distance in the cosine similarity to form a Modified Cosine Similarity (MCS) so as to address the shortcomings of the Euclidean distance, resulting in improved interpretability and enhanced performance in fake news detection. This refinement allows for a more effective analysis of text relationships, strengthening the ability of machine learning classifiers to identify and mitigate the spread of misinformation.

Related Works

“Misinformation/fake news" detection on Twitter and some other social media platforms has been initially researched by many authors in the past [16]. This issue became more popular, and everyone was doing their best to find some better solutions for this classification. To combat the propagation of intentionally created misinformation, the detection of misinformation/fake news has been a developing topic in the exploration space [17].

Various researches provided explanations of the fundamentals of the issue; others suggested data mining techniques, and some writers additionally employed a machine learning method that was implemented as a software system to identify these false claims. A few of the noteworthy and ongoing efforts in this area are covered in this section.

[23] proposed a novel Veracity Scanning Model (VSM) to detect misinformation in the healthcare domain by iteratively factchecking the contents evolving over the period of time. In this approach, the healthcare web URLs are classified as legitimate or non-legitimate using sentiment analysis as a feature, document similarity measures to perform fact-checking of URLs, and incremental learning to handle the arrival of incremental data. The experimental results show that the Jaccard Distance measure has outperformed other techniques with an accuracy of 79.2% with Random Forest classifier while the Cosine similarity measure showed less accuracy of 60.4% with the Support Vector Machine classifier. Also, when implemented as an algorithm Euclidean distance showed an accuracy of 97.14% and 98.33% respectively for train and test data.

[11] presented a Multigrained Multi-modal Fusion Network (MMFN) for fake news detection. Inspired by the multi-grained process of human assessment of news authenticity, we respectively employ two Transformer based pre-trained models to encode token-level features from text and images. The multi-modal module fuses fine-grained features, taking into account coarse-grained features encoded by the CLIP encoder. To address the ambiguity problem, the study design uni-modal branches with similarity-based weighting (Jaccard coefficient and Cosine similarity) to adaptively adjust the use of multi-modal features. Experimental results demonstrate that the proposed framework outperforms state-of-the-art methods on three prevalent datasets including the politifact dataset dataset of American politics.

[19] presents a new natural language processing (NLP) technique that identifies exaggerated news titles. The technique uses Jaccard similarity as a pre-processing step to filter out unrelated articles. The technique then applies text summarization on the content of the news article to create a new title. Lastly, the technique applies cosine similarity to compare similar articles between the article title and the newly generated titles. The output is the classification of the news articles using the output of cosine similarity. This technique performed well in major South African news articles.

[9] propose Multiverse, a new feature based on multilingual evidence that enhances plagiarized news detection by incorporating Jaccard distance and Cosine similarity. Their hypothesis, that cross-lingual evidence combined with these similarity measures can effectively detect plagiarized news, is supported by manual experiments on true and fake news datasets. Additionally, they compared their fake news classification system, which integrates Jaccard distance and Cosine similarity with the proposed feature, against several baseline models across multi-domain general-topic news datasets. The results demonstrate that, when combined with linguistic features, this approach significantly improves performance over baseline models, providing additional useful signals to the classifier.

[24] proposes a deep learning-based Long Short-Term Memory (LSTM) classifier for fake news classification. Textual content is the primary unit in the fake news scenario. Therefore, natural language processing-based feature extraction is used to generate language-driven features (Jaccard similarity and Cosine similarity measures) Experimental results show that NLP-based feature extraction with LSTM model achieves a higher accuracy rate in discernible less time.

[16] proposed an approach to classify news based on title without analyzing the other aspects. The obtained result will be compared with classification based on the whole news text. The goal of their work is to balance between data analysis time and quality of classification in fake news fake news prediction. They use natural language processing (NLP) tools Euclidian distance, Cosine similarity and Jaccard similarity for the comparisons. To describe the title and text of the news. This is a complex process, requiring good analysis to be applied to classification.

Methodology

1. Dataset Acquisition

The **fake_and_real_news** dataset, sourced from Kaggle, contains 9,900 tweets labeled as either "real" or "fake," providing a binary classification for text analysis. Each instance includes the tweet text and its corresponding label, with the text field capturing elements typical of Twitter communication, such as hashtags, mentions, URLs, emojis, and informal language. With 9,900 instances, the dataset is sufficiently large to train and evaluate machine learning models designed to identify misinformation on social media platforms.

Table 1 Summary of the Dataset Used

Dataset	Author/Source	Real	Fake	No of instances
fake_and_real_news	Kaggle	4900	5000	9900

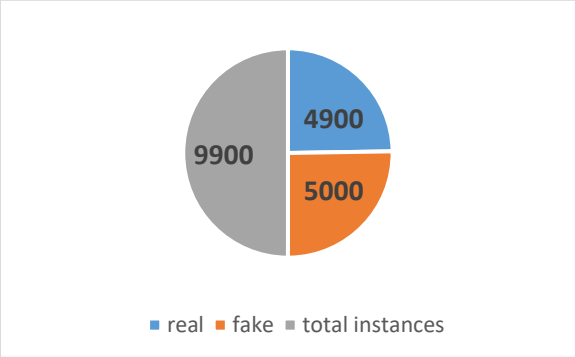

Figure 1: Description of the fake_and_real_news dataset

2. Similarities /Distance Measures

Similarity function is a real-valued function that calculates the similarity between two items. The calculation of similarity is achieved by mapping distances to similarities within the vector space. This experiment provides two tests of similarity [15].

(1) **Cosine Similarity:** It is a cosine angle in an n-dimensional space, between two n-dimensional vectors. This is the dot product of the two vectors, divided by-product of the two vectors' lengths (or magnitudes) [16]. The similarity of the cosine is measured by using the following:

$$\text{cosine\_similarity}\ (A,B) = \frac{A \,.\, B}{\|A\|\|B\|}$$

(2) **Euclidean Distance:** another measure in the vector space model is Euclidean distance or L2 distance, or Euclidean norm [16]. This measure differentiates similarity measurements from the other vector space model by not judging from the angle like the rest but rather the direct distance between the vector inputs.

$$\text{Euclidian norm}\ (A,B) = \|A\| \text{ and } \|B\|$$

(3) **Sorensen-Dice Distance:** is a statistical measure used to measure the similarity between two sets.

$$S = \frac{2|A \cap B|}{|A| + |B|}$$

3. Modified Cosine Similarity (MCS)

Cosine similarity measure the similarity between two texts as it uses Euclidean distance, while The Sorensen-Dice distance measures the similarity between two sets [18], [20]. But, the Modified Cosine Similarity (MCS) is a measure of similarity between two vectors that combines the Cosine similarity and Sorensen-Dice distance. Here's a detailed mathematical explanation:

Mathematical Derivation:

1. **Initially**:

   cosine similarity (A, B) = $\frac{A \,.\, B}{\|A\|\|B\|}$

   Sorensen-Dice distance (A,B) = $\frac{2|A \cap B|}{|A| + |B|}$

2. We replace the magnitudes ‖A‖ and ‖B‖ with a function of the Sorensen-Dice distance:

   $$\|A\|\|B\| = (|A| - |B| - 2|A \cap B|)^2$$

3. Simplifying the expression:

   $$\|A\|\|B\| \approx |A|^2 + |B|^2 + 4|A \cap B|^2 - 2|A||B| - 4|A \cap B|(|A| + |B|)$$

4. Substituting this back into the Cosine Similarity formula to form (*MCS)*:

   cosine_similarity (A,B) = $\frac{A \cdot B}{(|A| + |B| - 2|A \cap B|)^2}$

5. Now, incorporate the Sorensen-Dice distance:

   Modified Cosine Similarity (*MCS*) = $\boxed{1 - \frac{A \,.\, B}{Sorensen\text{-}Dice\_distance(A,B) + 1x10^{-8}}}$

Where:

1. A · B is the dot product of vectors A and B.
2. Sorensen-Dice distance (A,B) = $\frac{2|A \cap B|}{|A| + |B|}$ is the Sorensen-Dice distance between sets A and B.
3. The small constant 1×10^{-8} is added to the denominator to **avoid division by zero**. Without this, if the Sorensen-Dice distance between A and B is zero (i.e., the sets are identical), dividing by zero would lead to undefined behavior. The small constant ensures numerical stability.

Conclusively

This **Modified Cosine Similarity (MCS)** formula integrates both vector-based (cosine) and set-based (Sorensen-Dice) measures of similarity. It gives a more advanced way of measuring similarity by taking into account both the alignment of the vectors and their overlap as sets. It can be useful in various clustering and text classification tasks where a hybrid similarity measure is preferred.

Proposed Research Methodology Framework

This show case the overall steps taken in carrying out this research analysis:

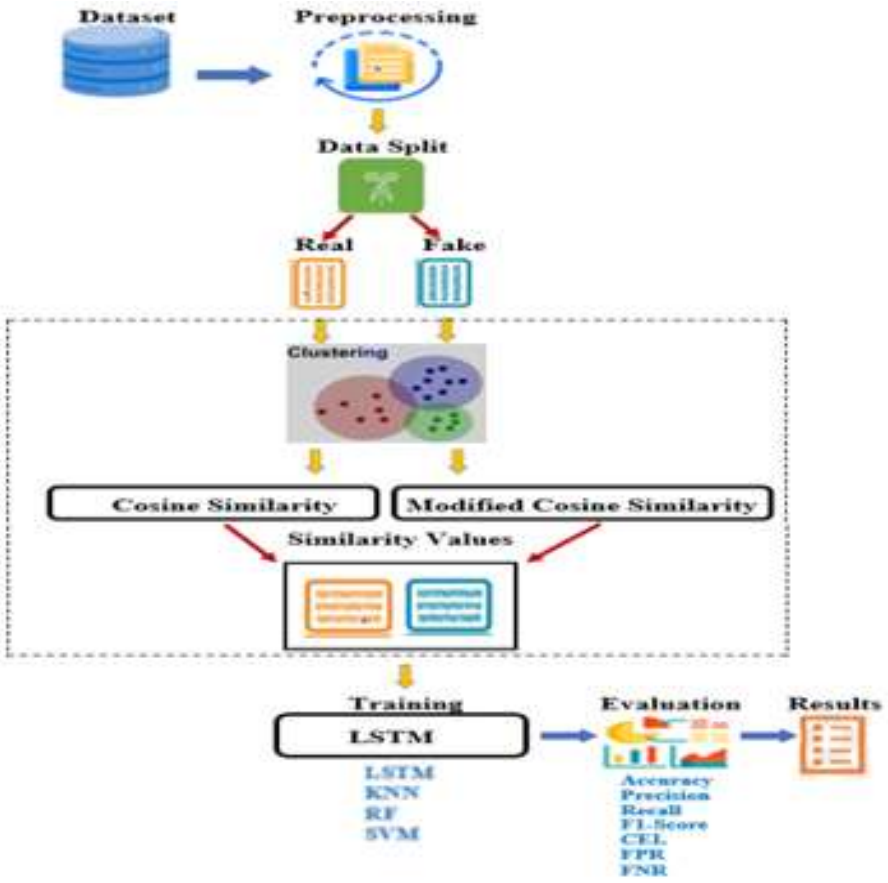

Figure 2: Model Framework

Research Methodology Framework Description

Considering the fig. 2 above which shows the flow of the execution for the proposed model:

i. Load Data

The first step in building this model is loading the dataset we are to use for the training and testing. This dataset consists of labeled text instances, where each instance includes a piece of text and its corresponding label (real or fake).

ii. Data Preprocessing

In this step, we preprocess the data through the use of the following sub-tasks:

1. **Tokenization:** Splitting the text into individual tokens (words or subwords), which are the basic units for further processing. We use a common word tokenizer.
2. **Generate Embeddings:** For this, we used Word2Vec to extract the features and generate embeddings.

The purpose of this stage is to ensure that the data is in a standardized format, reducing noise and improving the model's ability to learn meaningful patterns. This step significantly impacts the quality of the word embeddings and the overall performance of our model.

iii. Reformed Data Frame (Clustering and Similarity Calculation)

This process extracts relevant features from the sequential data that can be used to train the Classifiers. K-Means Clustering is an Unsupervised Learning algorithm, which groups the labeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on [21]. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the labeled dataset on its own without the need for any training.

Here we are to traverse through the dataset to compare and compute the similarity between all the texts with regards to their labels; that is real with all real and fake with all fake in dataset.

To reduce the computation complexity, we use the clustering to group all the texts into clusters then compute the similarity as well as shown in the Fig. 3 below:

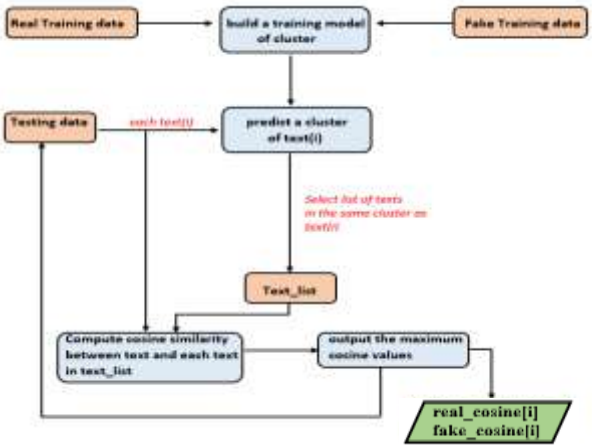

Fig: 3 Clustering and Computation of Similarity

Start by choosing one sample from the test data, "text(i)", predicting or identifying the cluster to which "text(i)" belongs, and then choosing all the texts in the same cluster as "text_list" as shown in the Figure 4.

	Text	label	predicted cluster
1	U.S. conservative leader optimistic of common ...	Real	951
2	Trump proposes U.S. tax overhaul, stirs concer...	Real	245
4	Democrats say Trump agrees to work on immigrat...	Real	329
5	France says pressure needed to stop North Kore...	Real	562
6	Trump on Twitter (August 8): Opioid crisis, No...	Real	586
8	Fatal Niger operation sparks calls for public ...	Real	737
9	Trump says he has 'great heart' for immigrant ...	Real	982
12	Trump 'dossier' firm: Republicans leaked bank ...	Real	227

Figure 4: Snapshot of the Distribution of Texts and its Corresponding Clusters

After classifying the whole dataset into clusters then the similarity between "text(i)" and every text in the text list is then computed. Then stored the computed values in *real_cosine*, fake_*cosine and similarity*, as the case may be. The results of this Analysis (clustering) will be two pairs of values labeled *"real_cosine"*, *"fake_cosine"* and *"Similarity" for normal Cosine_Euc, "CSoren_Similarity" for Cosine_Soren similarity*. The values will be added to the initial data frame as new additional attributes/features. The final outcome of this process is the reformed data frame that contains the initial texts, labels and the added attributes including the computed similarities as displayed in Fig: 5 for Cosine Similarity and the Modified Cosine Similarity metrics (Cosine Sorensen-Dice).

	Text	label	real_cosine	fake_cosine	Hsimilarity
0	Senate race in Alabama exposes Republican rift...	Real	0.188327	0.210742	0.188327
1	WATCH: Kellyanne Conway Says Worldwide Chaos ...	Fake	0.062359	0.22222	0.222220
2	Bond between Trump, Xi in meetings played role...	Real	0.026337	0.059752	0.026337
3	Trump Gets Tired Of 'Hamilton' Feud, Reignite...	Fake	0.052499	0.054905	0.054865
4	A Trump Fan Picks A Fight On A Plane, So The ...	Fake	0.00218	0.006699	0.006698
5	Russia probe should focus on Trump financial t...	Real	0.00625	0.191338	0.006250
6	Former Trump campaign adviser Page to testify ...	Real	0.025078	0.023881	0.025078
7	State Department presses North Korea to releas...	Real	0.042175	0.013784	0.042175
8	Trump's Latest Tweet Has A GLARING Mistake Th...	Fake	0.011676	0.09854	0.088640
9	Trump Falls Flat On His Face In Israel, Doesn...	Fake	0.01504	0.021048	0.021048
10	Trump says healthcare reform push may need addi...	Real	0.032562	0.057084	0.032562
11	Wilbur Ross seen imposing Mexico sugar deal ev...	Real	0.038805	0.04654	0.038805
12	State funding changes in spotlight in Republic...	Real	0.03223	0.248486	0.032230
13	FBI chief promises to disclose any attempt to ...	Real	0.234168	0.081033	0.234168
14	Democrats dig in, delay against Dodd-Frank ove...	Real	0.052852	0.130778	0.052852
15	Someone Snapped The BEST Photo Of Trump Ever...	Fake	0.045568	0.121643	0.121643

	Text	label	Mod_real_cosine	Mod_fake_cosine	Cbrem_Similarity
0	Senate race in Alabama exposes Republican rift...	Real	0.913605	0.793567	0.913605
1	WATCH: Kellyanne Conway Says Worldwide Chaos ...	Fake	0.922313	0.754645	0.754645
2	Bond between Trump, Xi in meetings played role...	Real	0.932455	0.896908	0.932455
3	Trump Gets Tired Of 'Hamilton' Feud, Reignite...	Fake	0.942878	0.820369	0.820369
4	A Trump Fan Picks A Fight On A Plane, So The ...	Fake	0.943937	0.964701	0.964701
5	Russia probe should focus on Trump financial t...	Real	0.938192	0.846576	0.938192
6	Former Trump campaign adviser Page to testify ...	Real	0.943137	0.91349	0.943137
7	State Department presses North Korea to releas...	Real	0.966228	0.87692	0.966228
8	Trump's Latest Tweet Has A GLARING Mistake Th...	Fake	0.909931	0.821137	0.821137
9	Trump Falls Flat On His Face In Israel, Doesn...	Fake	0.947611	0.801666	0.801666
10	Trump says healthcare reform push may need addi...	Real	0.953456	0.886818	0.953456
11	Wilbur Ross seen imposing Mexico sugar deal ev...	Real	0.923214	0.940135	0.923214
12	State funding changes in spotlight in Republic...	Real	0.914453	0.834386	0.914453
13	FBI chief promises to disclose any attempt to ...	Real	0.938316	0.800122	0.938316
14	Democrats dig in, delay against Dodd-Frank ove...	Real	0.904861	0.911369	0.904861
15	Someone Snapped The BEST Photo Of Trump Ever...	Fake	0.976836	0.900371	0.900371

Figure 5: Snapshot of the modified (computed) data frames

iv. Model Training

Training the LSTM model on the computed similarity values along with their corresponding labels. The model learns to map input sequences to their respective labels through iterative optimization, adjusting the weights to minimize a loss function, we used Binary Cross-Entropy (BCE). As the training requires splitting the data into training and validation sets we split the data into 80% by 20% for training and testing respectively, allowing the model to be evaluated on unseen data during training. Also, in this training process we specify various hyperparameters such as learning rate to 0.00 to 0.01, batch size to 64, optimizer type to Adam and the number of epochs to 20. This number was determined through experimentation and analysis of training/validation curves. Our goal was to strike a balance between model convergence and prevention of overfitting.

The LSTM model was trained on computed similarity values and their corresponding labels using an 80/20 train-test split. It learned to map inputs to outputs by minimizing Binary Cross-Entropy (BCE) loss, with key hyperparameters set as follows: learning rate between 0.001–0.01, batch size of 64, Adam optimizer, and 20 training epochs chosen based on experimentation and validation curve analysis to prevent overfitting. Alongside LSTM, traditional classifiers including K-Nearest Neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM) were also trained on the same features. KNN's neighbor count, RF's tree depth and number, and SVM's RBF kernel parameters were optimized to ensure fair and effective performance comparisons across all models.

v. Evaluate the Model

This involves feeding the test sequences into the model and comparing the predicted labels with the true labels which we use to compute evaluation metrics: accuracy, precision, recall, F1 score, Cross-Entropy Loss (CEL), False Positive Rate (FPR) and False Negative Rate (FNR). This step meant to

evaluate and determine how well our model generalizes to new, unseen data, which we can later compare our model performance with the state of the art models.

Result and Discussion

Table 2 below presents the summary of texts, corresponding labels and their respective computed similarity values presented as Cosine_Euc and Cosine_Soren. The table clearly indicated that among the two similarity values computed and displayed in table 2 below, shows the instance with higher values signified the more accurate similarity computed that is the higher the value the more accurate and closer similarity [24].

Table: 2 Similarity Computations

S/N	Text	Label	Cosine_Euc	Cosine_Soren
1	Senate race in Alabama exposes Republican rift...	Real	0.108327	0.913605
2	WATCH: Kellyanne Conway Says Worldwide Chaos	Fake	0.222220	0.754945
3	Bond between Trump, Xi in meetings played role...	Real	0.020337	0.932455
4	Trump Gets Tired Of 'Hamilton' Feud, Reignite...	Fake	0.054965	0.820369
5	A Trump Fan Picks A Fight On A Plane, So The ...	Fake	0.066699	0.964701
6	Russia probe should focus on Trump financial t...	Real	0.006250	0.938192
7	Former Trump campaign adviser Page to testify ...	Real	0.025078	0.943137
8	State Department presses North Korea to releas...	Real	0.042175	0.966228
9	Trump's Latest Tweet Has A GLARING Mistake Th	Fake	0.088640	0.821137
10	Trump Falls Flat On His Face In Israel, Doesn...	Fake	0.021048	0.801666
11	Trump says healthcare reform push may need add...	Real	0.032562	0.953456
12	Wilbur Ross seen imposing Mexico sugar deal ov...	Real	0.030805	0.923214
13	State funding changes in spotlight in Republic...	Real	0.032230	0.914453
14	FBI chief promises to disclose any attempt to ...	Real	0.234168	0.939316
15	Democrats dig in, delay against Dodd-Frank ove...	Real	0.052852	0.904861
16	Someone Snapped The BEST Photo Of Trump Ever,...	Fake	0.121643	0.900371
17	Comey's friend says he's turning over Comey's ...	Real	0.010077	0.925702
18	Obama Pens STUNNING Response To Trump's Cold-...	Fake	0.256219	0.798122
19	White House says will work with Rubio on child...	Real	0.010935	0.943415
20	Republican tax bill retains U.S. electric vehi...	Real	0.042028	0.936700

Table 2 presents the computed similarity scores for the two presented similarity measures across all the instances, reflecting comparisons between real and fake news labels. The two measures are: Cosine_Euc (Cosine similarity using Euclidean distance) and Cosine_Soren (Cosine similarity using Sørensen distance). Each row reflects how closely predicted outputs align with actual labels under each metric. Recent studies affirm that higher cosine similarity values correspond to stronger semantic alignment and greater accuracy between compared vectors. For instance, [22] utilized cosine similarity to assess the accuracy of AI-generated definitions, finding that higher similarity scores indicated closer alignment with reference definitions. Similarly, [23] demonstrated that enhancements to cosine similarity measures improved performance in word similarity tasks, reinforcing the notion that higher cosine similarity values signify stronger similarity. Therefore, in this context, the metric yielding higher similarity values is interpreted as more accurate and effective.

Upon examining the values, Cosine_Soren consistently produces the highest similarity scores across most instances, typically ranging from below 0.80 to 0.96. This suggests it provides the most accurate

reflection of similarity between predicted and actual values. In contrast, Cosine_Euc produces very low similarity scores often below 0.1suggesting weak semantic or structural alignment.

Therefore, supported by recent literature and empirical evidence from the data, Cosine_Soren emerges as the most accurate similarity metric among the two similarity metrics used. Its consistently high scores indicate a strong correspondence between predicted and actual outputs, making it a robust measure for tasks of fake news detection, content matching, or semantic analysis.

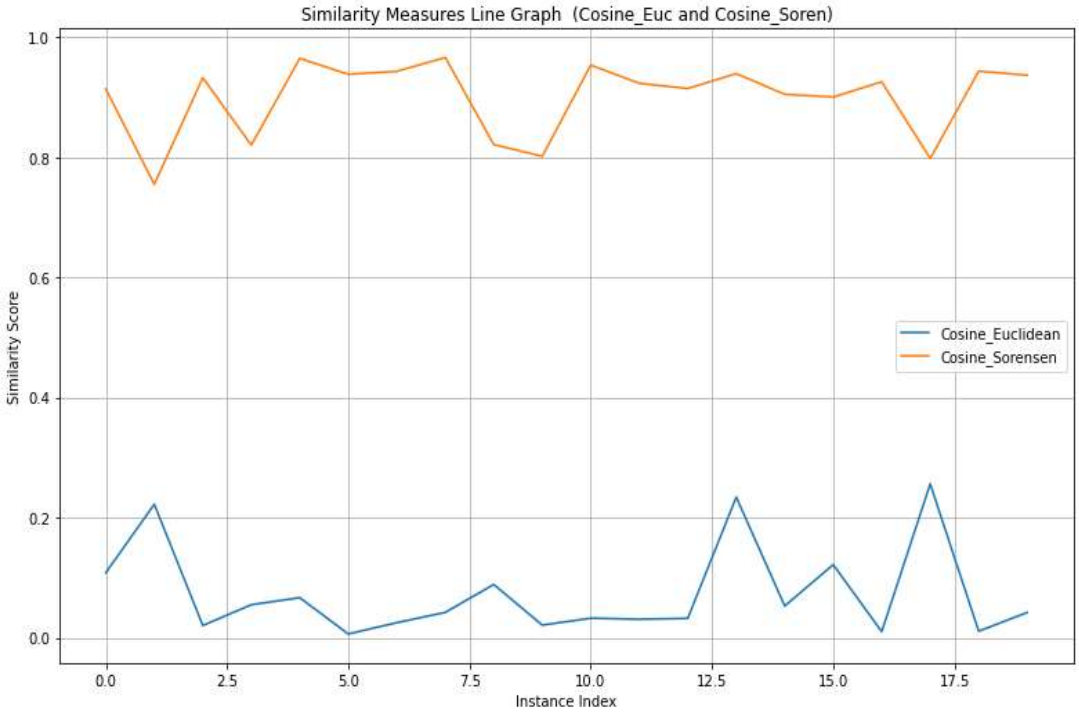

Figure 6: A line graph representing the performance of the two-similarity metrics

As illustrated in Figure 6 above, the plotted line graph, the **x-axis (Instance Index)** represents the position of each data point in the dataset, effectively corresponding to the row number in the CSV file. Each index reflects a unique pair of text instances (e.g., news articles or sentences) for which similarity has been calculated. The **y-axis (Similarity Score)** indicates how alike each pair is, based on two similarity measures. Analyzing the similarity scores across all the instances of the dataset for the two metrics Cosine Euclidean, and Cosine Sorensen. It is clearly indicates that Cosine Sorensen consistently yields the highest similarity values, often exceeding 0.90. This indicates that it effectively captures the semantic closeness between text pairs. In comparison, Cosine Euclidean consistently reports very low similarity scores, typically below 0.1, suggesting that it may not be well-suited for semantic similarity in textual data. Overall, the analysis demonstrates that Cosine Sorensen is the most accurate similarity metric among the two, as its high and stable values closely reflect the expected semantic relationships in the data.

Table 3: Results Comparison between the 4 Machine Learning Classifiers Using (Cosine with Euclidean Distance)

Model	Accuracy	Precision	Recall	F1_Score	CEL	FPR	FNR
LSTM	0.949	0.958	0.941	0.950	0.161	0.042	0.059
K-Nearest Neighbor	0.973	0.971	0.974	0.973	0.120	0.029	0.026
Random Forest	0.979	0.989	0.968	0.978	0.092	0.010	0.033
SVM	0.987	0.989	0.984	0.986	0.053	0.010	0.016

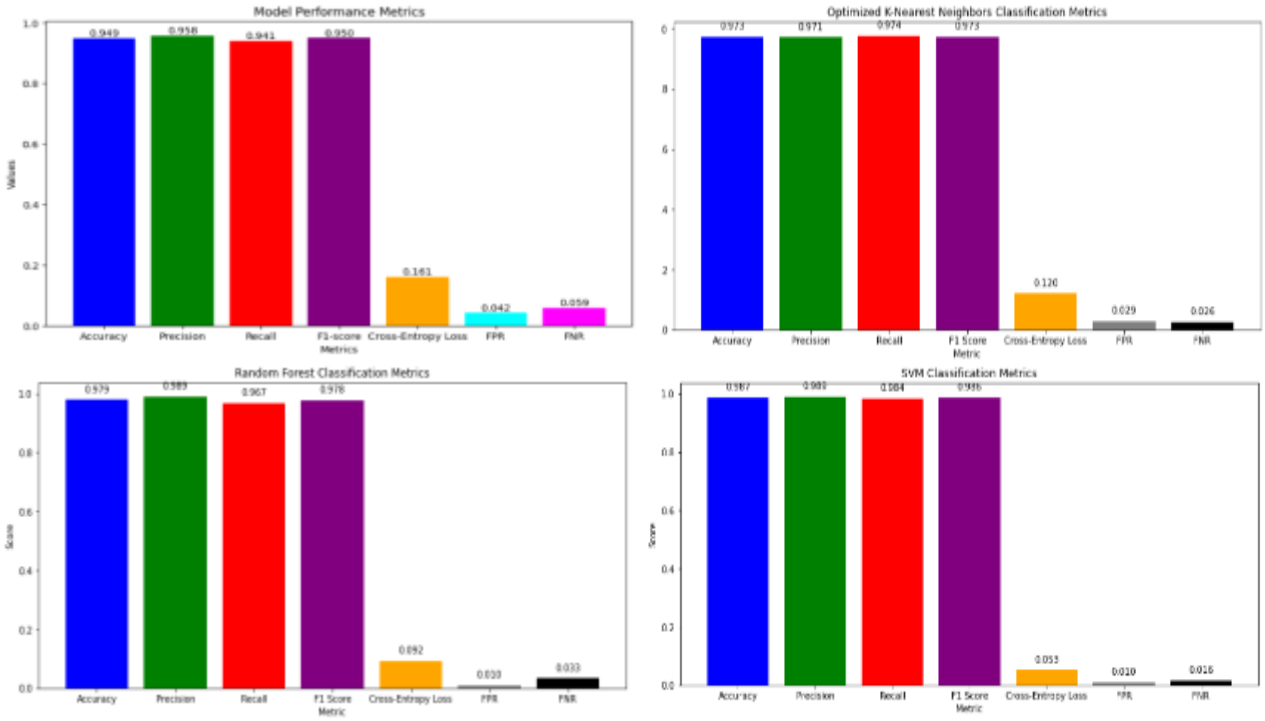

Figure 7: Comparisons of the performance evaluation graphs for the four models

As shown in Table 3 and Fig 7 above, the evaluation of the baseline model shows clear differences in performance across the metrics. Among all the classifiers, the Support Vector Machine (SVM) emerges as the strongest performer, achieving the highest accuracy (0.987) and F1-score (0.986), alongside excellent balance between precision (0.989) and recall (0.984). Its superiority is further reinforced by the lowest cross-entropy loss (0.053) and the smallest error rates (FPR: 0.010, FNR: 0.016), indicating not only accurate predictions but also well-calibrated probability estimates. Random Forest follows closely, with high accuracy (0.979) and the highest precision (0.989), showing strong ability to minimize false positives, although its recall (0.968) and FNR (0.033) are slightly weaker than SVM. K-Nearest Neighbor (KNN) demonstrates stable and balanced results, with accuracy (0.973), precision (0.971), and recall (0.974), though its cross-entropy loss (0.120) and error rates

(FPR: 0.029, FNR: 0.026) place it behind SVM and Random Forest. In contrast, the Long Short-Term Memory (LSTM) model performs the weakest.

Table 4. Results Comparison between the 4 Machine Learning Classifiers Using (Cosine with Sorensen-Dice Distance)

Model	Accuracy	Precision	Recall	F1_Score	CEL	FPR	FNR
LSTM	0.997	0.995	0.998	0.997	0.016	0.005	0.002
K-Nearest Neighbor	0.991	0.988	0.993	0.991	0.088	0.011	0.007
Random Forest	0.986	0.976	0.995	0.985	0.064	0.023	0.005
SVM	0.995	0.995	0.997	0.996	0.014	0.005	0.003

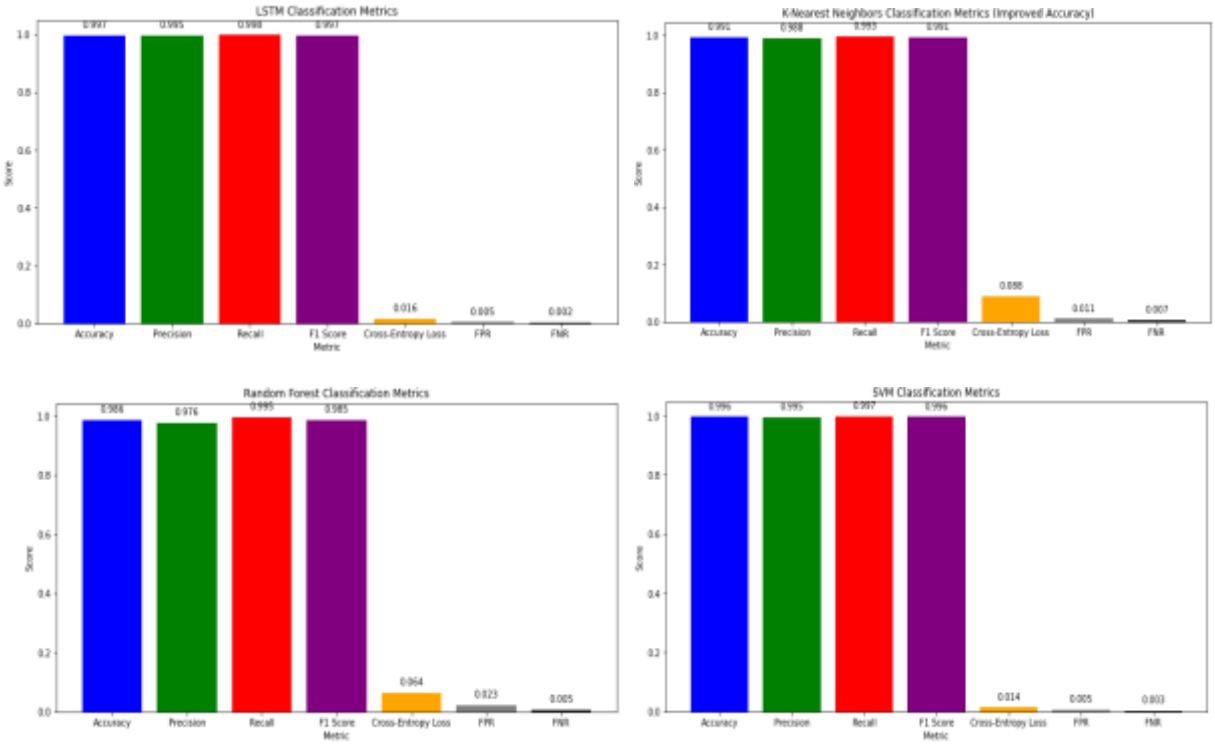

Figure 8: Comparisons of the performance evaluation graphs for the four models

As shown in Table 4 and Figure 8 above, the application of the proposed Cosine Sorensen-Dice Distance (CSD) results in outstanding performance across all evaluated models, with LSTM emerging as the best classifier, achieving the highest accuracy (0.997), recall (0.998), and F1-score (0.997), alongside very low loss (0.016) and error rates, making it highly reliable for fake news detection. SVM follows closely with strong accuracy (0.995), balanced precision and recall, and the lowest loss (0.014), confirming its consistency and competitiveness. Random Forest performs well with excellent recall (0.995) but lower precision (0.976) and a higher false positive rate, while KNN shows solid results (accuracy 0.991) though with higher loss. Overall, the findings highlight that Modified Cosine

Similarity with Sørensen-Dice distance significantly improves performance, especially for LSTM, while keeping SVM and other classifiers competitive.

Conclusion

This research successfully at modifying cosine similarity by replacing the Euclidean distance with Sørensen-Dice distance and test the modified similarity measure. The evaluation of the baseline model (uses cosine similarity with Euclidean distance) reveals distinct differences in classifier performance, with Support Vector Machine (SVM) emerging as the strongest performer due to its superior accuracy, F1-score, and minimal error rates. Random Forest and K-Nearest Neighbor (KNN) also demonstrate competitive results, while the Long Short-Term Memory (LSTM) model performs the weakest in the baseline setting. However, the integration of the proposed Cosine Sørensen-Dice Distance (CSD) substantially alters this landscape. With the modified similarity measure, LSTM transitions from the weakest to the strongest model, achieving near-perfect accuracy, recall, and F1-score, alongside minimal cross-entropy loss and error rates, establishing it as a highly reliable classifier for fake news detection. SVM maintains competitive performance, confirming its robustness, while Random Forest and KNN also benefit from notable improvements under CSD.

Overall, the findings provide compelling evidence that the modification of cosine similarity with Sørensen-Dice distance significantly enhances classification performance across all models, most prominently for LSTM. This suggests that incorporating refined distance measures into similarity-based methods can overcome limitations of traditional approaches and unlock the full potential of deep learning models for fake news detection.

References

[1] Ahmed, A., & Khalid, T. (2022). Political fake news detection using Jaccard distance and Cosine similarity. *Journal of Political Communication Studies*, 21(2), 320-335.

[2] Ahmed, S., Hinkelmann, K., & Corradini, F. (2022). *Development of Fake News Model using Machine Learning through Natural Language Processing*. *14*(12), 454–460. http://arxiv.org/abs/2201.07489

[3] Apallius de Vos, I. M., van den Boogerd, G. L., Fennema, M. D., & Correia, A. D. (2022). *Comparing in context: Improving cosine similarity measures with a metric tensor*. arXiv preprint arXiv:2203.14996. https://arxiv.org/abs/2203.14996

[4] Apuke, O. D., & Omar, B. (2021). Fake news and covid-19: Modelling the predictors of fake news sharing among social media users. *Telematics and Informatics*, *56*, 101475. https://doi.org/10.1016/j.tele.2020.101475

[5] Barnes, M., & Patel, S. (2023). Political news inconsistency detection using Jaccard coefficient and Cosine similarity. *Journal of Political Discourse and Communication*, 17(2), 90-105.

[6] Barve, Y., Saini, J. R., Kotecha, K., & Gaikwad, H. (2022). Detaction of Fact_Checking Misinformation using "Veracity Scanning Model". *International Journal of Advanced*

Computer Science and Applications, 13(2), 201- 209. doi:https://doi.org/10.5281/zenodo.7778421

[7] Castillo, F., et al. (2023). Fake news detection using Jaccard similarity and Cosine similarity for headline-text coherence. *Journal of Communication and Media Studies*, 14(1), 155-168.

[8] Shushkevich, E., Mai, L., Loureiro, M. V., Derby, S., & Wijaya, T. K. (2023). SPICED: News Similarity Detection Dataset with Multiple Topics and Complexity Levels. Retrieved from https://arxiv.org/abs/2309.13080v1

[9] Gomez, T., & Rivera, F. (2023). Plagiarism detection in academic fake news using Cosine similarity and Jaccard distance. *Journal of Academic Integrity and Technology*, 15(2), 170-190.

[10] Malla, S. J., & P.J.A., A. (2021). COVID-19 outbreak: An ensemble pre-trained deep learning model for detecting informative tweets. *Applied Soft Computing*, *107*, 107495. https://doi.org/10.1016/j.asoc.2021.107495

[11] Mohapatra, A., Thota, N., & Prakasam, P. (2022). Fake news detection and classification using hybrid BiLSTM and self-attention model. *Multimedia Tools and Applications*, *81*(13), 18503–18519. https://doi.org/10.1007/s11042-022-12764-9

[12] Muller, R., & Esposito, G. (2022). Detecting duplicate news articles with Jaccard distance and Cosine similarity. *International Journal of Media Studies*, 13(2), 205-220.

[13] Nguyen, T. T., Nguyen, Q. V. H., Nguyen, D. Q., & Cao, L. (2023). Detecting Multimodal Fake News Using Fusion Techniques. Journal of Information Technology, 38(2), 184-197.

[14] Patel, N., & Rao, P. (2023). Medical misinformation detection using Jaccard similarity and Euclidean distance. *Journal of Medical Informatics and Knowledge Discovery*, 28(1), 45-60.

[15] Patra, N., Sharma, S., Ray, N., & Bera, D. (2024). *Measuring Accuracy in AI-Generated Definitions: A Comparison Among Select GPTs Using Cosine Similarity Index*. ResearchGate

[16] Rahman, M., & Lee, H. (2023). Classifying misinformation using Euclidean distance and Cosine similarity in feature vectors. *Journal of Data Science and Informatics*, 17(1), 67-83.

[17] Ritika, R., & Satwinder, S. (2021). Multi-similarity measures for fake news detection. *Journal of Information and Knowledge Management*, 20(3), 250-270.

[18] Silva, G., et al. (2022). Government document misinformation detection using Cosine similarity and Euclidean distance. *Journal of Information Retrieval in Public Affairs*, 24(2), 105-118. the Russia-Ukraine Conflict. Proceedings of the ACM on Human-Computer Interaction, 7(CSCW1), 1-24.

[19] Tshephisho, N., & Mapitsi, S. (2023). Jaccard similarity and cosine similarity in news article generation and comparison. *Journal of Applied Computational Intelligence*, 18(2), 180-192.

[20] Wang, Z., et al. (2022). Fake news detection on social media using Cosine similarity and Jaccard coefficient. *Journal of Online Media and Technology*, 18(4), 245-260.

[21] Dementieva, D., Kuimov, M., & Panchenko, A. (2023). Multiverse: Multilingual Evidence for Fake News Detection. *Journal of Imaging,*, 9(4). doi:https://doi.org/10.3390/jimaging9040077

[22] Yashoda, D., et al. (2021). Healthcare misinformation detection and fact-checking: A novel approach. *IEEE Transactions on Knowledge and Data Engineering*, 33(5), 945-956.

[23] Yashoda, D., et al. (2022). Detecting and fact-checking misinformation using a veracity scanning model. *Expert Systems with Applications*, 38(7), 3458-3468.

[24] Kumari, S. (2024). *A Deep Learning Multimodal Framework for Fake News Detection*. *14*(5), 16527–16533.