DETECTING FAKE NEWS THROUGH PROBABILISTIC SENTIMENT MODEL AND SENTENCE EMBEDDING TECHNIQUE

[1]May Me Me Hlaing

Faculty of Information Technology Supporting and Maintenance,
University of Computer Studies, Yangon
Myanmar
maymimihlaing88@ucsy.edu.mm

[2]Win Lelt Lelt Phyu

Faculty of Computer Science,
University of Computer Studies, Yangon
Myanmar
wllphyu@ucsy.edu.mm

Abstract

False information is purposely disseminated through fake news, aiming to spread hoaxes, propagation, and disinformation. Analyzing news truthfulness is a challenging problem because there is a massive information on social networking sites which turns out to be very difficult to manually analyze. Moreover, the impact of fake news is tremendously huge for internet users. As a result, academic research related to filtering and banning fake news has been highly demanding in very recent years. The paper contributes to this field by using a probabilistic sentiment model with a sentence embedding method, addressing the limitations of approaches relying on TF-IDF or Word2Vec methods. The proposed method incorporates probabilistic sentiment scores and sentence embedding features to improve the accuracy of the fake news detection system, employing Naïve Bayes and Support Vector Machine classifiers. The proposed method achieves a high accuracy of 99% F1-score by combining these features.

***Keywords*: Fake News Detection; Naïve Bayes; Probabilistic Sentiment Model; Sentence Embedding Method; Support Vector Machine.**

1. Introduction

The convenience of information propagation through social media has significantly increased in today's digitized world. The consumption of news through traditional mediums such as television and newspapers has significantly declined, giving way to the dominance of social media as the primary source of news for many people. Anyone can create and disseminate information instantly with just a smartphone. This can lead to easy manipulation of the existing news, thereby causing fake news which can have negative effects on society [Olan *et al.*, (2022)], [Paven *et al.*, (2020)]. Fake news is intentionally fabricated or misleading information presented as factual news, aiming to deceive readers, viewers, or listeners.

As the prevalence of fake news continues to rise, people's faith in real news diminishes, causing them to lose focus on the actual issues at hand [Nirav and Ganatra, (2022)]. The common dissemination of fake news can have a significant impact on society. Since fake news is purposely fabricated, it can be applied for personal gain, financial or political purposes, and to tarnish the reputation of individuals or organizations. Fake news is a ubiquitous challenge that is difficult to solve in today's digital age since there are so many places where misleading information can spread [Shravani *et al.*, (2023)]. Hence, it is very necessary to prevent the spread of fake news. Human readers have difficulty distinguishing true information from false accurately by just reading at short pieces of information [Naik and Patil, (2021)]. Recognizing the authenticity of information on the internet is crucial since misinformation can lead to significant societal disruptions. False information has the potential to incite riots, create chaos, and impact a wide segment of society [Rawat *et al.*, (2021)]. Detecting false news is a complicated task to overcome. Extensive research efforts are directed toward employing machine learning and deep learning methods to classify news articles as either fake or real [Senhadji and Ahmed, (2022)]. The combination of machine learning and text-based processing empowers us to discern fake news and create classifiers for news data classification [Ahmed *et al.*, (2022)].

The main objective of this research is to enhance the accuracy of fake news detection system using probabilistic sentiment scores and sentence embedding features. To increase the accuracy of the fake news detection model, capture semantic understanding, provide robust feature representation, enable contextual analysis, enhance generalization, and fuse multiple features have been applied in the proposed model. This paper includes three main contributions which are described in the following.

The first contribution this research employs the generation of probabilistic sentiment scores relies on the TF-IDF feature and a pre-existing sentiment model. However, when working with a big vocabulary, TF-IDF often produces a high-dimensional and sparse vector representation which can provide challenges in terms of memory usage and processing performance. The information gain approach for feature reduction is thus integrated into the sentiment model to speed up processing time. Moreover, this research employs sentence embeddings as a text feature representation technique which offers a more compact and meaningful representation of sentences compared to traditional bag-of-words or TF-IDF approaches. Secondly, this research examines the impact of probabilistic sentiment scores on the system for identifying fake news. Sentence embeddings and probabilistic sentiment scores can greatly enhance the effectiveness of fake news identification. Probabilistic Sentiment scores give the likelihood that news is fake or real during the model training, whereas sentence embeddings capture the semantic content and contextual data. The third contribution of this research shows SVM and NB classification models may easily incorporate the proposed compound features (probabilistic sentiment scores and sentence embedding features) with efficient accuracy for the detection of fake news.

The remainder of the paper is organized as follows. In section 2, we review the related work of the system. In section 3, we describe the proposed system and methodology. We discuss the nature of the ISOT fake news dataset, experimental results and discussion in section 4. The paper is concluded in section 5 by glimpsing the future direction of the current system.

2. Related Work

The quick dissemination of misinformation through social media has profound consequences, affecting not only individuals but also the broader human community, especially in political, reputational, and economic sectors. After the 2016 US election, analyzing news truthfulness on social media has become a particular attention in today's research area. There are many prior researches about this problem using machine learning techniques and most of them have been focused on classifying publicly available news posts on social media.

The system [Ahmad *et al.*, (2020)] used machine learning models and ensemble to classify fake news items. In this research, numerous textual features are used to distinguish between phony and true news. The experiment was carried out on four publicly available datasets from various domains, and the performance was calculated using performance measures. On the ISOT Fake News Dataset, random forest and Perez-LSVM reach 99% accuracy.

TF-IDF, word embedding and sentence embedding methods are applied for feature text representation and Linear SVM for classification tasks in [Sriram, (2020)]. The use of word embedding and sentence embedding techniques in fake news detection makes more sense because it allows for a better understanding of contextual meaning. Three datasets were obtained from the Kaggle website. The researchers obtained fake news articles from BuzzFeed, while reliable articles were sourced from the PolitiFact website. One major limitation was that Linear SVM's default hyperparameter was only used. The accuracy of the model typically decreased when any hyperparameters were adjusted.

The system [Khanam *et al.*, (2021)] suggested a count vectorizer and Tiff vectorizer to extract features from news articles. Six different algorithms were applied, namely Decision Tree, Random Forest, Support Vector Machine, Naïve Bayes, K-Nearest Neighbors, and Combining classifiers, to classify fake news. The detection of fake news was performed using the LIAR-PLUS Master dataset. While each algorithm showed significant performance and efficiency, certain overtrained methods employed by some algorithms led to issues such as overfitting, variances, biases, and generalization errors.

To differentiate between real and fake news [Pandey *et al.*, (2022)] described the Word2Vec technique to extract a feature vector, and applied KNN, SVM, DT, NB, and LR for classification. When dealing with the task of identifying fake news, the input data was in text format and involved a significant number of characteristics that required careful consideration. The model effectively solved this complex challenge by leveraging the suggested methods. The dataset, comprising 6553 English-language news articles, was collected from a GitHub public repository. Upon using Word2Vec, it was discovered that text processing was time-consuming, leading to the recommendation against its usage. Nonetheless, Word2Vec did provide semantic relations to facilitate the transformation of data into vectors.

In a study conducted by [Fayaz, (2021)], the classification of real or fake news was achieved through the implementation of a Random Forest (RF) classifier. To perform the experiment, the ISOT benchmark dataset was analyzed by extracting a total of twenty-three (23) features. The process of selecting the most relevant features from the original twenty-three was carried out using Chi2, Univariate, information gain and feature importance. The results of the comparative analysis showed that the proposed classifier was significantly models such as GBM, XG-Boost, and AdaBoost Regression models.

In previous studies [Pandey *et al.*, (2022)], [Sriram, (2020)], TF-IDF and Word2Vec are used to extract features from fake news. These existing fake news detection methods that use TF-IDF lack semantic understanding and face a considerable amount of feature dimension, and those that use Word2Vec, a word embedding model, fall short in capturing contextual details. To address these problems, information gain is used to reduce dimension of TF-IDF features in this paper. Moreover, sentence embedding is used to achieve both semantic understanding and capturing contextual details in feature extraction. Sentence embedding features and PSS are then combined to form effective feature vectors for fake news detection models, and they are trained and tested using SVM and NB classifiers. Finally, the impact of PSS on the accuracy of fake news detection models is experimented with using ISOT dataset in this paper.

3. Proposed System and Methodology

The proposed technique applies an effective Feature selection method with the combination of PSS model and sentence embedding for fake news classification. The PSS model can be obtained by using TF-IDF, information gain and logistic regression (LR). Sentence embedding ensures to get both word semantics and contextual information. Fig. 1. shows the overall flow of the proposed system for fake news detection and detailed explanation of each step is described in the following sub-sections.

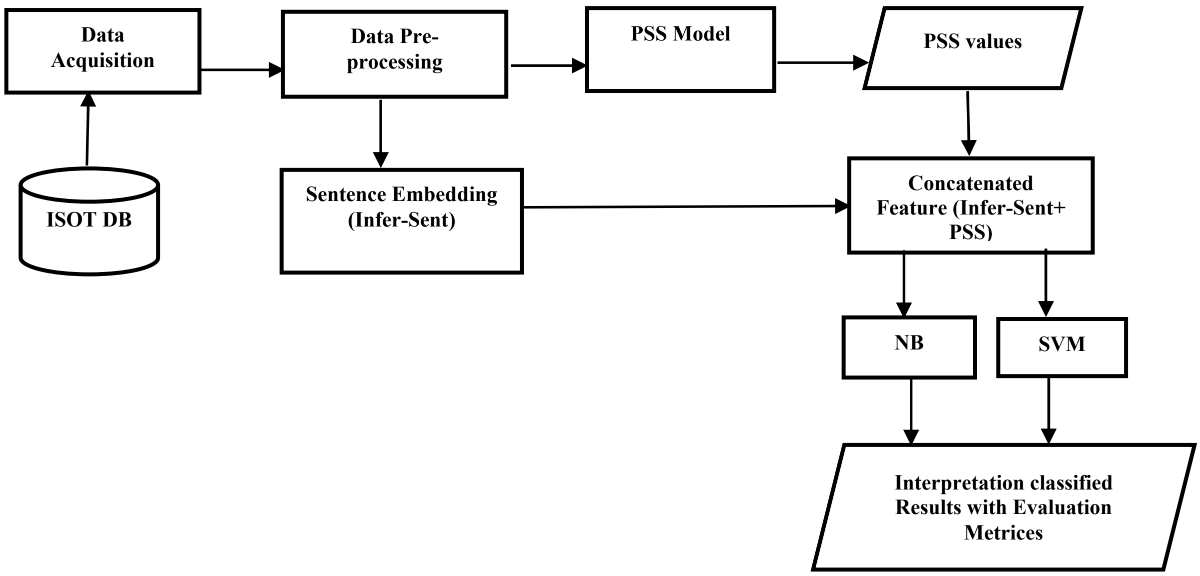

Fig. 1. Overall Flow of the Proposed System

3.1. *Data Pre-processing*

News articles from social media are unstructured, big and noisy. The data pre-processing step is very important as pre-processed data are reliable and consistent in building the machine learning models. Tokenization, stemming, feature extraction, POS (part of speech) tagging, and the removal of stop words are some of the preprocessing techniques frequently used in text analysis [Aksoy *et al.*, (2020)]. Fig. 2. illustrates the four primary steps involved in data pre-processing applied to this research work.

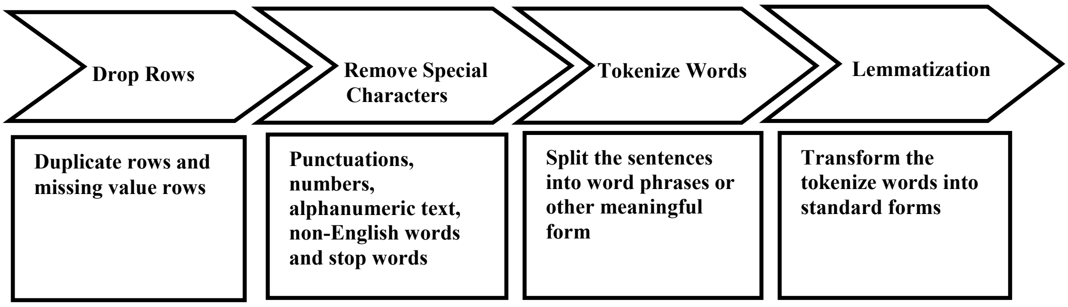

Fig.2. Steps of Data Preprocessing

3.2. *PSS Model*

The proposed technique has developed a probabilistic sentiment analysis model in Fig. 3. that can classify positive (real) and negative (fake) effectively. For constructing the model, the technique first takes pre-processed text, and it then counts the frequency of each word to calculate the TF-IDF score. The extracted TF-IDF features are in a considerable amount of 20000 words. Therefore, the proposed system applies information gain theory to select the best features and reduce dimension of the feature. To obtain PSS scores, the system implements a Logistic Regression classifier based on the set of selected features.

To vectorize the pre-processed documents, TF-IDF has to be employed. Using the TF-IDF vectorizer, you can tokenize documents, generate the vocabulary, determine inverse document frequency weights, and analyze new documents [Dzisevič and Šešok, (2019)]. To compute this metric, term frequency (TF) is first needed, and the inverse document frequency (IDF). TF-IDF calculates the relevance of a term in a document by considering its importance in the context of a larger set of documents.

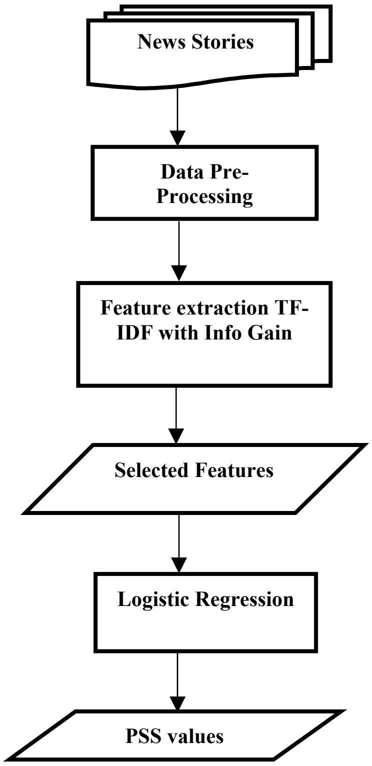

Fig. 3. The overview of PSS model

TF score is calculated by dividing the number of times a specific term appears within a document by the total number of terms in that document as described in "Eq. (1)" IDF is achieved by using "Eq. (2)". The TF-IDF of the term is calculated by multiplying TF and IDF score as shown in "Eq. (3)".

$$TF = \frac{no.\ of\ \ times\ the\ term\ appears \,\mathrm{in}\, the\ document}{total\ no.\ of\ \ terms \,\mathrm{in}\, the\ document} \tag{1}$$

$$IDF = \log \frac{no.\ of\ \ documents\ in\ the\ corpus}{no.\ of\ \ documents\ in\ the\ corpus\ contain \,\mathrm{in}\, the\ term} \tag{2}$$

$$TF\_IDF = TF * IDF \tag{3}$$

Information gain (Info-gain) can also be applied to the selection of features, by evaluating the gain of each variable in the context of the target variable. The method described by [Win and Kham, (2019)] for feature selection involved the integration of information gain parameters to improve the accuracy of classification algorithms. Mutual information between the two random variables is calculated in Info-gain. Mutual information I (X: Y) is the amount of uncertainty in X due to the knowledge of Y. Mathematically, mutual information is defined as shown in “Eq. (4)”.

$$I(X:Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \quad (4)$$

where, p (x, y) means the joint probability function of X and Y, p(x) is the marginal probability distribution function of X and p(y) is the marginal probability distribution function of Y.

Logistic Regression (LR) is a predictive algorithm using independent variables to predict the dependent variable, and it is a supervised machine learning technique. Logistic Regression aims to establish a relationship between the independent variables (features) and the dependent variables that require prediction. [Strzelecka *et al.*, (2020)]. In Logistic Regression, the aim is to predict the values of a categorical dependent variable, which must be in a categorical or discrete form. The value can take on binary forms such as "Yes" or "No," "0" or "1," "true" or "false," and so on, without specifying the precise values of 0 and 1. LR employs the sigmoid function, an "S"-shaped curve, for predicting values of 0 or 1, in contrast to the conventional regression line. The sigmoid function carries out a probabilistic estimation of the result converting any real number into a value within the range of 0 to 1. It uses the concept of the threshold value, which defines the probability of either 0 or 1. LR is depicted in Fig. 4.

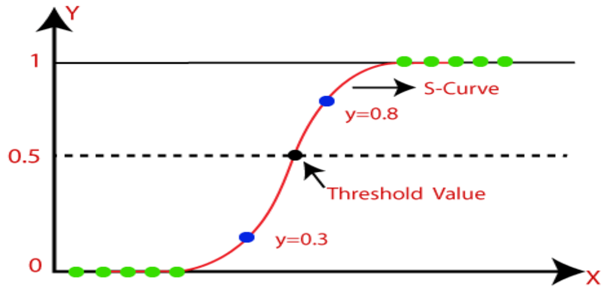

Fig. 4. Logistic Regression

3.3. *Sentence Embedding*

Machine learning models can handle only numeric values. The procedure of transforming text data into a numeric format is referred to as vectorization. Word embedding is a technique used in NLP to represent words as dense and continuous vectors in a high-dimensional space. Word embeddings may not capture the full range of word meanings, especially when dealing with polysemy or ambiguous words.

In contrast to word embedding, sentence embeddings enable the encoding of entire sentences into vectorized representation rather than focusing solely on individual words. Sentence embedding techniques aim to transform variable-length sequences of words into fixed-length representations. These representations are useful in various NLP tasks, such as text classification, sentiment analysis, information retrieval, machine translation, and question-answering systems [Wang *et al.*, (2019)].

Sentence encoders such as Google’s BERT and USE, Facebook’s InferSent, and AllenAI’s SciBERTand ELMo, have received significant attention in recent years [Hassan *et al.*, (2019)]. The proposed system used the InferSent sentence embedding method. InferSent developed by Facebook AI Research is a popular sentence embedding model in the field of NLP. InferSent is designed to learn general-purpose sentence representations by leveraging the information from natural language inference (NLI) data.

Within InferSent's word encoding model, the process starts by encoding each word in the sentence through pre-trained word embeddings, like GloVe or fastText. Word embeddings capture the semantic meaning of individual words. InferSent uses a bidirectional Long Short-Term Memory (BiLSTM) neural network to encode the sentence. The BiLSTM reads the sequence of word embeddings in both forward and backward directions, capturing contextual information. After the BiLSTM encoding, max-pooling is performed over the output of the BiLSTM layer. Max-pooling extracts the most salient information from each dimension of the BiLSTM output, resulting in a fixed-length vector representation of the sentence. The max-pooled vector is fed into a Multi-Layer Perceptron (MLP) with fully connected layers. The MLP further transforms the representation, mapping it to a lower-dimensional space. InferSent is trained using a supervised learning approach on NLI data. During training, the model is trained to forecast the relationship between the premise and hypothesis in the sentence pairs, using the softmax function to calculate the probabilities for entailment, contradiction, and neutral categories. The model's parameters are adjusted through backpropagation and gradient descent to minimize the classification loss. Once trained, InferSent can generate sentence embeddings for unseen sentences by passing them through the learned network. These sentence embeddings capture semantic and contextual information, allowing for various downstream tasks such as text classification, semantic similarity, clustering, and more. The architecture of InferSent is composed of two primary components: sentence encoder and NLI classifier. The sentence encoder is responsible for transforming the input sentence into the fixed-length vector that captures its meaning and context. NLI classifier uses the encoder vectors to perform text classification. The general flow of InferSent is illustrated in Fig. 5.

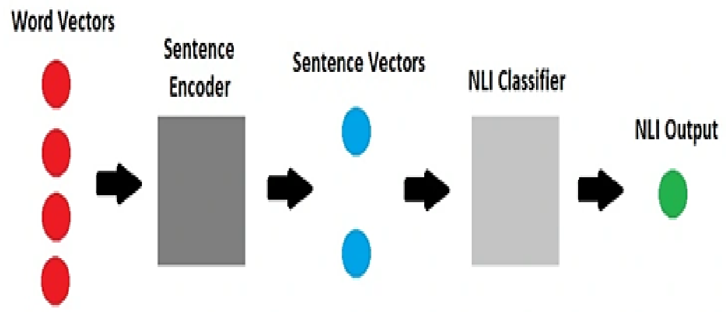

Fig. 5. General flow of InferSent

3.4. *Feature Combination and Concatenation*

The PSS model extracts positive (real) or negative (fake) sentiment effectively with high accuracy. The scores of ten news articles from the PSS model are shown in Fig. 6. If the PSS score of a news is greater than or equal to 0.5, it means that this news is real; unless fake. To combine this score with the sentence embedding features set, the system drops out the PSS values from any one column. The PSS of sample 10 news is shown in Table 1. According to PSS values from this table, we assume news id “0” is “Real”, news id “1” is “Real” and news id “6” is “Fake” etc.,

The sentence embedding features vector is combined or concatenated with PSS features vector to form the proposed features vector. In the proposed system, the size of the proposed features vector for training is of (38209, 101), for testing, (4469, 101). The first 100 features are sentence embedding features is shown in Fig. 7. and Fig. 8. The proposed concatenated feature set, as depicted in Fig. 9. in which the last value is PSS score.

	0	1
0	0.972079	0.027921
1	0.977765	0.022235
2	0.976047	0.023953
3	0.607438	0.392562
4	0.994298	0.005702
5	0.929328	0.070672
6	0.086406	0.913594
7	0.169626	0.830374
8	0.985901	0.014099
9	0.960284	0.039716

Fig. 6. Probabilistic Sentiment Scores of ten news articles from PSS model

News ID	PSS score
0	0.972079
1	0.977765
2	0.976047
3	0.607438
4	0.994298
5	0.929328
6	0.086406
7	0.169626
8	0.985901
9	0.960284

Table 1. The PSS scores for samples 10 news articles

```
array([-0.356298  , -0.9708023 , -0.8131391 , -0.94271684, -2.0595422 ,
       -0.20203313,  2.047049  , -1.5560294 , -2.0564077 , -0.11149304,
       -1.1510105 , -1.3021445 , -0.01798234,  0.8491376 ,  1.3420774 ,
       -0.45485964,  2.5911045 ,  0.9078252 , -0.12256911,  0.04492952,
       -0.64614147, -1.0363629 ,  0.87329054, -0.8073886 ,  0.4469836 ,
       -2.2796252 ,  0.66568357, -0.36830965, -2.6160486 ,  0.58305174,
        2.5458844 ,  1.8198209 , -0.21816179,  1.8387325 , -1.2759304 ,
       -0.48836553,  0.66925275, -0.5990626 , -0.04637432,  0.09905079,
       -0.97831964, -0.29723758,  1.4903473 ,  1.8631814 , -0.7136153 ,
       -0.8199292 , -0.66341925,  0.5903203 ,  1.0379261 ,  0.2756216 ,
        1.1758718 ,  0.2343356 ,  0.9698275 , -1.2291751 , -0.5848837 ,
       -0.2261102 ,  1.7494977 ,  1.0212746 ,  0.0245354 ,  1.0990841 ,
        1.7408248 , -0.10135457, -1.7149194 , -0.547552  , -0.8317477 ,
        0.2622178 ,  0.23497152,  0.35079843, -0.837125  ,  0.29094574,
       -0.00624786, -2.7520332 ,  1.746678  , -0.7303855 ,  0.5894148 ,
        0.19358955, -0.33190817, -0.83852166,  0.31196588, -0.41932595,
        0.8356228 , -0.57885456, -0.24028236,  1.5787357 , -0.03365032,
       -0.94193125,  0.41075596, -0.35013926,  0.07900082, -1.0485425 ,
        0.4215319 ,  2.378378  , -0.4699563 ,  0.24102546,  2.5823376 ,
        2.19714   ,  0.7007924 , -1.0051608 ,  1.5856403 , -0.24402286],
      dtype=float32)
```

Fig.7. Sentence Embedding Feature Set

	0	1	2	3	4	5	6	7	8	9	...
0	-0.356298	-0.970802	-0.813139	-0.942717	-2.059542	-0.202033	2.047049	-1.556029	-2.056408	-0.111493	...

...	90	91	92	93	94	95	96	97	98	99
...	0.421532	2.378378	-0.469956	0.241025	2.582338	2.19714	0.700792	-1.005161	1.58564	-0.244023

Fig.8. Sentence Embedding Features for vector size "100"

	0	1	2	3	4	5	6	7	8	9	...
0	-0.356298	-0.970802	-0.813139	-0.942717	-2.059542	-0.202033	2.047049	-1.556029	-2.056408	-0.111493	...

...	91	92	93	94	95	96	97	98	99	100
...	2.378378	-0.469956	0.241025	2.582338	2.19714	0.700792	-1.005161	1.58564	-0.244023	0.947345

Fig. 9. Concatenated Features (Sentence Embedding Features and PSS score)

3.5. *Classification Methods*

SVM is widely recognized as a powerful supervised machine learning algorithm capable of tackling intricate classification and regression tasks [Rajeswari and Karthikaa, (2022)]. In the proposed model, we apply with a binary classification model that splits the dataset attributes of different classes linearly by finding the best hyperplane evaluated according to "Eq. (5)". SVM can work well with data that has a large number of features. SVM is particularly well-suited for text classification tasks, where the number of features can be very large. In the system, a binary SVM algorithm is used that finds the best optimal hyper-plane to distinguish the input text data whether it belongs to real or false based on the training data. The radial basis function kernel is the most suitable one for large applications [Reddy *et al.*, (2019)], [Wang *et al.*, (2022)]. The advantage of using SVM is that it performs efficiently with high dimensional feature vectors. It can work well when the number of features can be much larger than the number of data points. The boundary function, f(x), is determined by the weight values (w) for input words (x), with the initialized weight (w_0) and the transposed vector (w^T) adjusting based on the input values.

$$f(x) = w^T x_i + w_0 \tag{5}$$

NB is a popular machine learning algorithm that works particularly well in NLP tasks. It is based on Bayes' theorem, which provides a way to calculate the probability of an event. NB classifiers are a family of simple probabilistic classifiers with strong (naive) independence assumptions between the feature [Chen *et al.*, (2020)], [Krishna and Kumar (2021)], [Poovaraghan *et al.*, (2019)]. NB calculates their co-occurrences, and finds the probability by comparing them with its class labels to classify text data as "Eq. (6)". According to "Eq. (6)", when applying Naïve Bayesian classification to news articles if the probability value for "Real" class label is greater than of "Fake", the article is classified as "Real". Otherwise, it was concluded as "Fake".

$$p(c \mid x_i) = \frac{p(x_i \mid c)\, p(c)}{p(x_i)} \tag{6}$$

where, $p(c \mid x_i)$ refers the posterior probability, $p(x_i \mid c)$ means likelihood, p(c) represents class prior probability, and $p(x_i)$ denotes predictor prior probability. Naive Bayes and SVM were applied for the fake news detection model, and the greatest results were obtained in detecting fake news [Anjali *et al.*, (2019)].

4. Implementation and Evaluation

4.1. *Dataset*

The ISOT dataset was launched by the Information Security and Object Technology (ISOT) Research Lab. The ISOT dataset is shown in Table 2. This dataset consisted of both real and fake articles which were collected from media coverage in the modern era; real news articles were extracted from Reuter.com, whereas fake articles were acquired from websites that had been identified as unreliable by fact-checking organizations such as PolitiFact and Wikipedia. While the dataset includes a broad range of articles spanning numerous topics, the majority of articles focus on political and world news. The news distribution of "Real" and "Fake" articles is shown in Fig. 10.

Type	**Number of Articles**	**Subjects**	
Real	21417	World News	10145
		Political News	11272
Fake	23481	Government News	1570
		Middle East	778
		US News	783
		Left-news	4459
		Politics	6841
		News	9050

Table 2. News categories and number of news articles per category

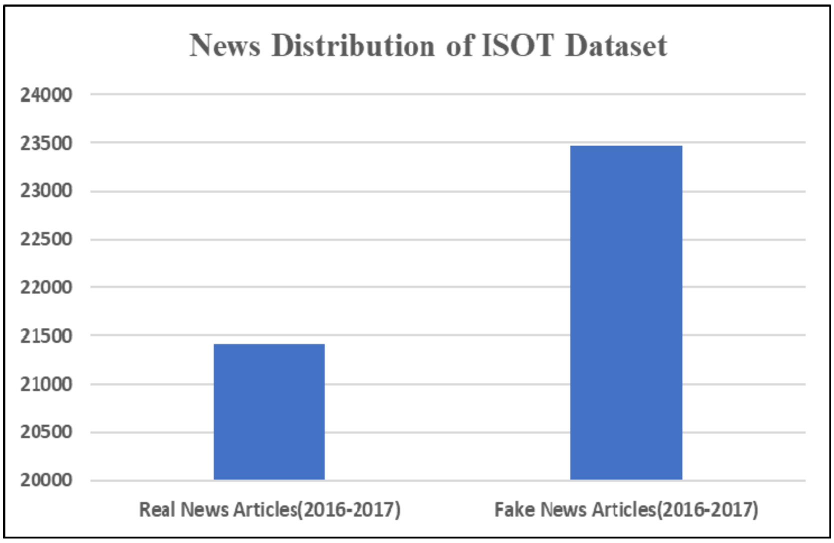

Fig. 10. News Distribution of ISOT Dataset

4.2. *Development Setting*

The proposed system is implemented with Python programming. Numpy and Pandas libraries are used to manage data preparation and file handling. Scikit-learn, Keras and TensorFlow libraries are exploited to apply NB and SVM. Due to the limitation of computing resources, we only use five thousand selected features using information gain theory from the dataset in this system. The dataset is divided into 20% for testing purposes and 80% for training in our approach. For the parameters for classification algorithms, we employ default parameters included in Keras and TensorFlow libraries as these settings are both straightforward and robust for achieving effective classification.

4.3. *Experimental Results and Discussion*

The results of fake news detection based on the proposed model are achieved by training and testing it with NB and SVM classifiers. These results are described in Table 3, Fig. 11. and Fig. 12. The accuracy, precisions, recall and f1-score can be computed using confusion matrices, and these are used to evaluate the proposed technique. For binary classification with the positive and negative classes, true positive (TP), false positive (FP), false negative (FN) and true negative (TN) are used to create confusion matrices that are used to compute the evaluation matrices. For each classifier, a single confusion matrix is constructed.

TP refers to the number of positive observations that are correctly classified; FN refers to the number of positive observations that are incorrectly classified as negative; FP refers to the number of false observations that are incorrectly classified as positive; TN refers to the number of false observations that are correctly classified. Accuracy calculated using “Eq. (7)” measures the percentage of overall effectiveness made by the model. Precision calculated using “Eq. (8)” indicates that the model is correctly identifying positive instances. Recall calculated using “Eq. (9)” measures the ratio of true positive predictions out of all positive instances in the data. F1-score calculated using “Eq. (10)” is a machine learning evaluation metric that combines precision and recall, and computes how many times a model made a correct prediction across the entire dataset.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$\Pr ecision = \frac{TP}{TP + FP} \tag{8}$$

$$\operatorname{Re} call = \frac{TP}{TP + FN} \tag{9}$$

$$F1-score = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

In the proposed model, the combination of sentence embedding features and PSS feature with information gain, trained by SVM has the highest accuracy of 99 % F1-score which outperforms the other models as shown in Table 3.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
NB (With PSS)	98.46	98.5	98.5	98
NB (Without PSS)	95.55	95.5	95.5	95.5
SVM (With PSS)	98.99	99	99	99
SVM (Without PSS)	97.65	97.5	97.5	98

Table 3. Performance Analysis of the Proposed Model

Apparently in Table 3, Fig. 11. and Fig. 12, our proposed methods: NB (with PSS) and SVM (with PSS) could stand tallest around (98.46%,98.99%), (98.5% ,99%) and (98.5% ,99%) in accuracy, precision and recall rather than NB (without PSS) and SVM (without PSS). All performance evaluations of two classifiers with PSS are better than these two classifiers without PSS values. Moreover, based on the results, we can also conclude that SVM performs better than NB in all evaluation metrics. Because, NB cannot handle well sparsely of languages due to its solely examination of co-occurrences with class labels, which is a little far precision from the language linguistics and semantic concepts.

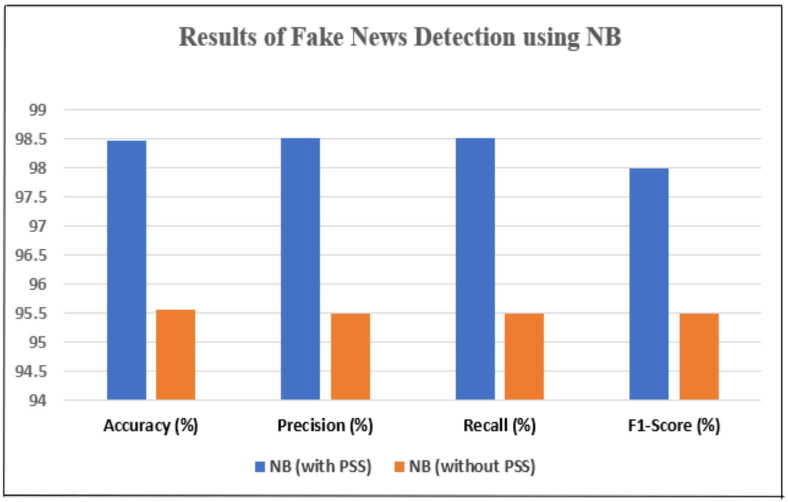

Fig. 11. Results of fake news detection by NB

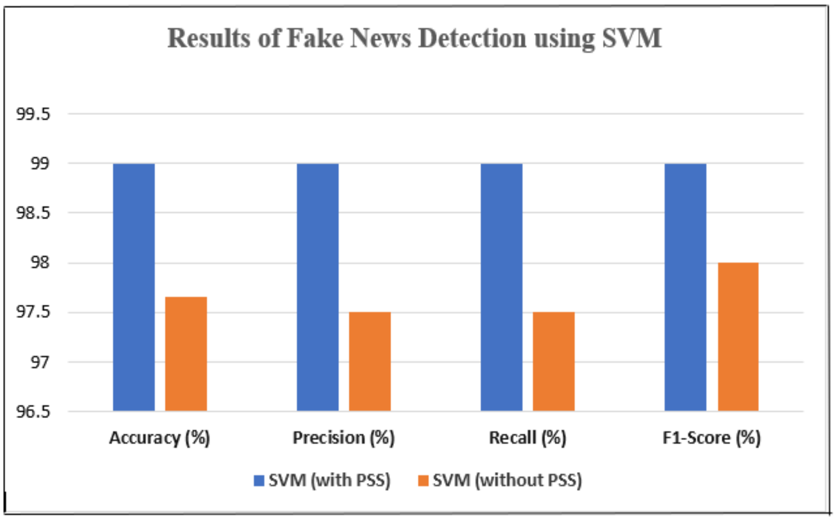

Fig. 12. Results of fake news detection by SVM

5. Conclusion

Fake news spreading has extremely negative impact on individuals and today's society. This study aims to develop a robust fake news detection technique using effective feature extraction from fake news articles or documents and machine learning. The massive number of dimensions of feature vectors are unable to be trained and tested on most of computers, and it is still a challenging task in NLP.

The proposed technique first contributes to the impact of PSS on detecting fake news. The information gain theory and logistic regression techniques are applied to reduce feature dimensions and extract PSS feature vectors. Sentence embedding is then applied to extract features vectors from documents. The PSS feature vector and the sentence embedding feature vector are then concatenated to form an effective feature vector for fake news detection. Finally, the proposed feature vector is trained and tested using SVM and NB classifiers.

The good impact of PSS will increase significantly when using other datasets. Further studies can be done by testing various datasets using the proposed fake news detection method. The key to this study is that the proposed two techniques, TF-IDF + information gain and Sentence Embedding + PSS, not only reduce the number of feature dimension but also improve the accuracy of the fake news detection model.

Acknowledgments

I would like to reveal my gratitude to my dearest supervisor for their guidance, motivation, and ongoing support throughout this study. Throughout my work, their extensive knowledge and innovative thinking have been an invaluable support.

Conflicts of Interest

The authors have no conflicts of interest to declare.

References

[1] Ahmad, I., et.al (2020): Fake news detection using machine learning ensemble methods. *Complexity*, *2020*, pp.1-11.

[2] Ahmed, S., et.al (2022): Development of fake news model using machine learning through natural language processing. *arXiv preprint arXiv:2201.07489*

[3] Aksoy, et.al. (2020): "Data Pre-processing in Text Mining." Who Runs the world: Data :123.

[4] Anjali Jain, et.al. (2019): A smart system for fake news detection using machine learning,2019 IEEE 2nd International conference on ICICT.

[5] Chen. S, et.al (2020): A novel selective naïve Bayes algorithm. *Knowledge-Based Systems*, *192*, p.105361002E.

[6] Dzisevič, R. and Šešok, D. (2019): April. Text classification using different feature extraction approaches. In *2019 Open Conference of Electrical, Electronic and Information Sciences (eStream)* (pp. 1-4). IEEE.

[7] Fayaz, M. (2021): *Machine Learning Fake News Classification with Optimal Feature,* Research Article, University of Peshawar.

[8] Hassan, H.A.M., et.al. (2019): Bert, elmo, use and infersent sentence encoders: The panacea for research-paper recommendation? In *RecSys (Late-Breaking Results)* (pp. 6-10).

[9] Krishna, I. M. V; Sai Kumar, S. (2021): Fake News Detection Using Naive Bayes Classifier, *International Journal of Creative Research Thoughts (IJCRT) ISSN: 2320-2882*, Vol. 9, Issue 6.

[10] Khanam, Z., et.al. (2021): Fake news detection using machine learning approaches. In *IOP conference series: materials science and engineering* (Vol. 1099, No. 1, p. 012040). IOP Publishing.
[11] Nirav Shah, M. and Ganatra, A. (2022): A systematic literature review and existing challenges toward fake news detection models. *Social Network Analysis and Mining*, *12*(1), p.168.
[12] Olan, F., et.al. (2022): Fake news on social media: the Impact on Society. *Information Systems Frontiers*, pp.1-16.
[13] Paven, M, et.al. (2020): Fake News Detection using Machine Learning, International Research Journal of Engineering and Technology (IRJET), Vol. 7, No.12.
[14] Pandey, S., et.al. (2022): Fake news detection from online media using machine learning classifiers. In *Journal of Physics: Conference Series* (Vol. 2161, No. 1, p. 012027). IOP Publishing.
[15] Poovaraghan, R.J., et.al (2019): Fake news accuracy using naive bayes classifier. *International Journal of Recent Technology and Engineering (IJRTE)*, *8*(1C2), pp.2277-3878.
[16] Rawat MS, et.al. (2021): Detection of Fake News Using Machine Learning. International Journal of Engineering and Applied Physics. 1(2):205-9.
[17] Rajeswari, M; Karthikaa, R.S. (2022): Fake News Detection Using Machine Learning Using SVM Algorithm, *Journal of Emerging Technologies and Innovative Research (JETIR)*, Vol. 9, Issue. 6.
[18] Reddy, P., et.al. (2019): A Study on Fake News Detection Using Naïve Bayes, SVM. *Neural Netw. LSTM J Adv Res Dyn Control Syst*, *1*, pp.942-947.
[19] Samrudhi Naik, and Amit Patil. (2021): Fake News Detection using NLP, *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, Vol. 9, No. 12.
[20] Senhadji, S. and San Ahmed, R.A.(2022): Fake news detection using naïve Bayes and long short term memory algorithms. *IAES International Journal of Artificial Intelligence*, *11*(2), p.746.
[21] Shravani B. V, et.al. (2023): Fake News Detection Using Machine Learning, *International Journal of Research Publication and Reviews* ISSN 2582-7421, Vol 4, no 5, pp 463-467.
[22] Sriram, S. (2020): An Evaluation of Text Representation Techniques for Fake News Detection Using: TF-IDF, Word Embeddings, Sentence Embeddings with Linear Support Vector Machine.
[23] Strzelecka, A; Kurdyś-Kujawska, A; Zawadzka, D. (2020): Application of logistic regression models to assess household financial decisions regarding debt. *Procedia computer science*, *176*, pp.3418-3427.
[24] Wang, B., et. al. (2019): Evaluating word embedding models: Methods and experimental results. *APSIPA transactions on signal and information processing*, *8*, p.e19.
[25] Wang, Q. (2022): Support Vector Machine Algorithm in Machine Learning. In *2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)* (pp. 750-756). IEEE.
[26] Win, T.Z. and Kham, N.S.M. (2019): *Information gain measured feature selection to reduce high dimensional data* (Doctoral dissertation, MERAL Portal).

Authors Profile

May Me Me Hlaing received the M.C.Sc. degree from the University of Computer Studies, Yangon, Myanmar in 2010. She is currently working as a Lecturer in the University of Computer Studies, Panglong, Myanmar. Her other research interests are Natural Language Processing, Machine Learning and Deep Learning. She can be contacted at email: maymimihlaing88@ucsy.edu.mm.

Dr.Win Lelt Lelt Phyu got Ph.D (IT) in 2006. Currently, she is working as a professor of the Artificial Intelligence Lab at the University of Computer Studies, Yangon, Myanmar. She has been supervising Master's thesis and Ph.D. research on Machine learning and artificial intelligence. She can be contacted at email: wllphyu@ucsy.edu.mm.