Fake News Detection Using Machine Learning and Deep Learning

Mustabshir Ibn Amin
Computer Science and Engineering
Islamic University of Technology
Dhaka, Bangladesh
mustabshir@iut-dhaka.edu

Jarin Tasnim Hridy
Computer Science and Engineering
Islamic University of Technology
Dhaka, Bangladesh
tasnimhridy@iut-dhaka.edu

Khawaja Ishmam Arian
Computer Science and Engineering
Islamic University of Technology
Dhaka, Bangladesh
ishmamarian@iut-dhaka.edu

***Abstract*—The rapid expansion of fake news has become a major issue worldwide. The escalation of fake and misleading news has led to many social and economic impacts, affecting industries from finance to healthcare. People now are chronically online, so, they read news from different social networking sites and believe those without giving much thought. In this paper, we aim to classify whether the news is true or false following some methodologies related to machine learning as well as deep learning using Natural Language Processing (NLP). We have used an English news dataset as our base. Our work uses traditional machine learning models, such as - Support Vector Machine (SVM) with different kernels, Logistic Regression, Multinomial Naïve Bayes, and Random Forest. As for the deep learning models, we used a simple Recurrent Neural Network (RNN) and a Long Short-Term Memory (LSTM). Our model uses feature extraction techniques such as Term Frequency-Inverted Document Frequency (TF-IDF). We achieved an accuracy score of 99.63% using Linear SVM (LSVM) on the benchmark dataset, but it failed to capture the intricacies of the data; however, that was captured by our LSTM model with an accuracy score of 99.84%. Our code is available on GitHub: [1]**

***Index Terms*—neural networks, linear support vector machine, logistic regression, multinomial naive bayes, random forest, recurrent neural networks, long short-term memory, natural language processing**

I. Introduction

Today is the age of information, there is digital news content from various sources which also leads to the spread of fake news through various avenues like social media, websites, etc. Fake news is misleading or fabricated information disguised as legitimate news. Finding out if a news is genuine or false is crucial for maintaining a proper information system.

The motivation behind our project is the societal impact of fake news. Fake news misinforms the public, influences decision-making, and can incite violence, so to remedy this, we have chosen our project to detect fake news in the information sphere.

Fake news constitutes a wide range of inaccurate content: misleading headlines, satirical content, biased reporting, and fabricated stories. The challenge of detecting fake news is the evolution of misinformation propagation, making it hard to use a rule-based approach. The news being fake depends on the context, as a result machine learning models need to be taught the context through NLP algorithms. There is also adata imbalance since there is more real news compared to fake news.

Ahmad et al. [1] discussed the difficulty of automatically classifying news as genuine or fake and suggested the ensemble approach over individual machine learning algorithms. Conroy et al. [2] explore linguistic cue approaches using machine learning and network analysis approaches, suggesting a hybrid method containing multi-layer linguistic processing, network behavior, and augmenting human judgment. Ahmed et al. [3] proposed a model using n-gram analysis and applying machine learning techniques. Comparing two feature extraction methods and six classification models, the best performance was obtained using TF-IDF and Linear SVM, achieving an accuracy score of 92%. Monti et al. [4] introduced a new model for classifying fake news on social media. They coined it ”geometric deep learning.” This approach generalizes convolutional neural networks (CNNs) to graphs by emphasizing news propagation patterns in detecting fake news due to the dissimilarity between real and fake news patterns. The model can detect fake news with 92.7% ROC AUC at an early stage. Nasir et al. [5] discussed a hybrid deep learning model called FA-KES, a hybrid of CNN and RNN for fake news detection. The hybrid model consists of 6 layers, including an LSTM layer. Seven different classifiers were compared against the FA-KES model which outperformed all of them in four metrics of accuracy. Thota et al. [6] use a dense deep learning approach to detect fake news using news titles and descriptions. The best model using dense neural networks and TF-IDF vectorizer achieved 94.21% accuracy on test data. AbdulRahman and Baykara [7] in their paper used in total of ten machine learning models and deep learning models using four feature extraction methods. They found the highest accuracy ($\approx 100\%$) using hybrid architecture (CNN+LSTM) and AdaBoost with TF-IDF feature extraction. Alghamdi et al. [8] conducted a massive comparative study of fake news detection using machine learning and deep learning methods and also transformer-based models conducting experiments across multiple datasets to compare their effectiveness. The study found that no single method outperforms others across all datasets. Bangyal et al. [9] conducted a study of fake news surrounding COVID-19 by sentiment analysis and using eight machine learning

[1]https://github.com/Musaimin/Fake-News-Detector.git

algorithms and four deep learning algorithms. Kumar et al. [10] focused on detecting fake news from social media like Twitter and using media sources like Pollifact to train deep learning models such as CNNs, LSTMs, ensemble methods, and attention mechanisms. The best-performing model that they got was the CNN and bidirectional LSTM ensemble with an attention mechanism that achieved an accuracy of 88.78%.

II. Methodology

Let's discuss the methodology of our project. Let us go step by step. First, we collected the dataset from Kaggle. Then, we viewed the data to understand its shape and characteristics. Then, we preprocessed the data separately for the machine learning and deep learning models. Then, we trained the selected models on the data by splitting it using a train-test split. Lastly, we found out the accuracy and errors from the test data.

A. Dataset Collection

The dataset was collected from Kaggle which is publically available [2]. The dataset contains news from 2015 to 2017 and it includes 17903 fake news articles and 20,826 real news articles.

B. Data Preprocessing

For the preprocessing step, we removed duplicated and null data from the fake and true news datasets. Then, we **labeled** the fake data with 1 or, positive class and true data with 0 or, negative class. After that the fake and true datasets were merged and shuffled to induce randomness. Then, the title and description of the news were cleaned by making the letters lowercase, removing HTTP links, HTML tags, text within square brackets, newline characters, and words containing numbers. Then, stopwords were removed, and stemming was applied to the text. Additionally, for the deep learning models, a sentiment intensity analyzer was applied to the title, and description and the title and text were combined to make a content column. Lastly, for the deep learning preprocessing, random oversampling was used to make a balanced class distribution since the positive/fake class was less in quantity than the negative/true class.

C. Feature Extraction

Feature extraction in Natural Language Processing (NLP) is the process of changing raw text data into structured, quantitative representations that can be used for machine learning tasks, especially NLP. For the machine learning models, we used the **Term Frequency-Inverse Document Frequency (TF-IDF)** and measured the importance of a word in a file relating with the word's frequency across all files. For, the deep learning models, we used **Tokenization**. In tokenization, we break down the text into smaller usable units called tokens to create word embeddings using the embedding layer of the LSTM model.

[2]https://www.kaggle.com/datasets/bhavikjikadara/fake-news-detection/

D. Model Algorithms and Architecture

For the traditional machine learning models, we used the Scikit-learn library. We choose Logistic Regression (LR), Multinomial Naive Bayes (MNB), Support Vector Machine (SVM) with the linear, polynomial, and sigmoid kernel, and the Random Forest (RF) classifier with linear, polynomial and sigmoid kernel from the Scikit-learn library. For the deep learning model, we choose simple Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) neural network using the TensorFlow Keras library. In Figure 1, we see the architecture of our RNN that consists of three layers- an embedding layer that converts integer encoded words from tokens to dense vectors of fixed size, a simple RNN layer with 32 hidden units that uses the ReLU activation function and also includes dropout for regularization to combat over-fitting during training, and an output layer with only a single node that has a sigmoid activation function for binary classification of the news.

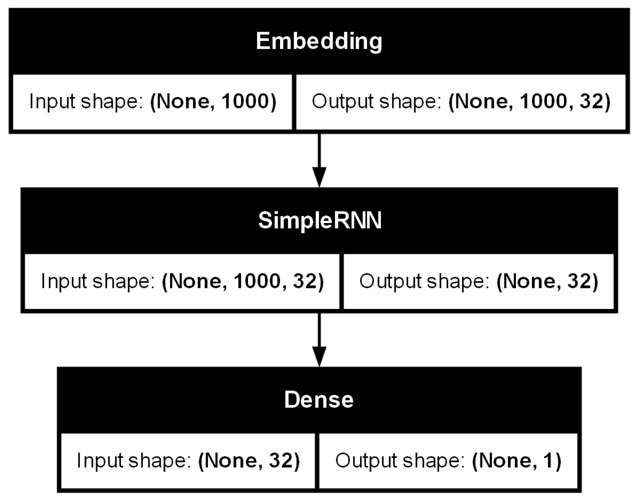

Fig. 1. Architecture for the simple RNN model used in our project

In Figure 2 we see the architecture of our bidirectional LSTM model that consists of five layers- an embedding layer that converts integer encoded words from tokens to dense vectors of fixed size, a bidirectional LSTM layer with 32 hidden units that utilizes both forward and backward information flow to capture context and return sequences to subsequent layers, a dropout layer for regularization to combat over-fitting during training, a second unidirectional LSTM layer with 32 hidden units, and a dense output layer with a single node with a sigmoid activation function for binary classification with L2 regularization applied to prevent overfitting.

In both of the architectures, the Adam optimizer was used, the loss function was the binary cross entropy function, and the metrics were set to accuracy.

E. Model Training Procedure

In both traditional ML and deep learning models, we performed a train-test split where 80% was training data and 20% was test data. All the machine learning models were

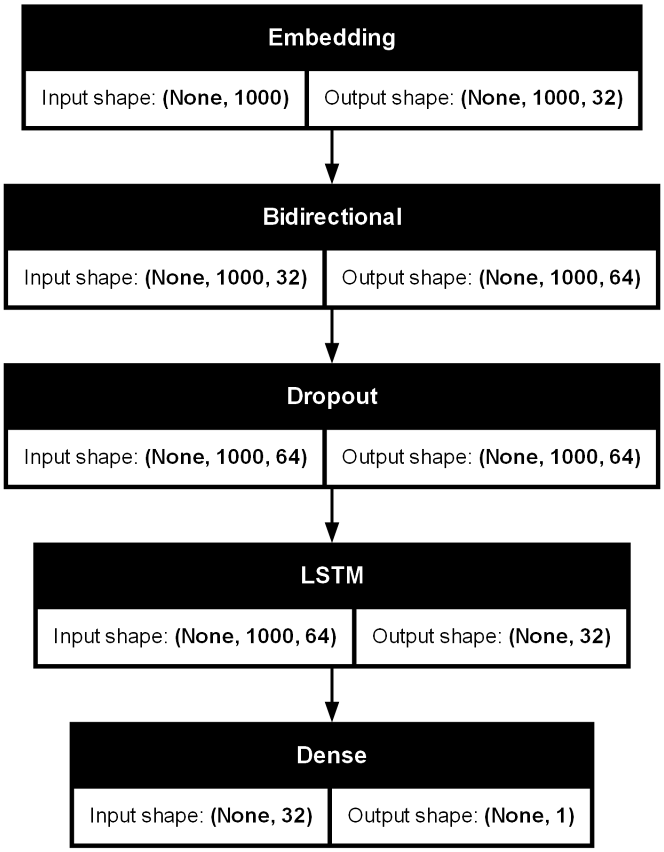

Fig. 2. Architecture for the LSTM model used in our project

trained separately to best fit the training data and their accuracy score was calculated using the test data. For the deep learning models, callback and L2 regularization were applied to prevent overfitting. The early stopping mechanism was used to prevent unnecessary epochs from running due to having the same value loss. The simple RNN model ran for four epochs and the LSTM model ran for twelve epochs and stopped before completing the assigned one hundred epochs due to early stopping.

III. Result Analysis

A. Experimental Setup

The experiments of this report were conducted on a desktop with RAM 32 GB, processor Intel Core i5 13500 and GPU AMD RX 6800, and the software used was VS Code and Python3 with Jupyter Notebook. The Streamlit Python library was used for developing the user interface of our app (via a website).

B. Evaluation

Now, we are going to evaluate and analyze some results of our project. At first, we will do some Quantitative Evaluation and after that, we will do some Qualitative Evaluation

1) Quantitative Evaluation: We have trained our model using machine learning and also on deep learning using feature extraction techniques of TF-IDF and NLP word embeddings from tokenized text sequences respectively. In Table I, we have shown some evaluation values of different machine learning models and deep learning models. We have shown Feature, Accuracy (A), Precision (P), Recall (R) and F1 score. Synthesis, Mean squared error in that table.

Model	**Feature Extraction Technique Used**	**A (%)**	**P (%)**	**R (%)**	**F1 (%)**
LR	TF-IDF	98.59	98.70	98.62	98.66
MNB	TF-IDF	95.30	96.65	94.35	95.48
LSVM	TF-IDF	99.50	99.56	99.45	99.52
RF	TF-IDF	99.20	99.55	98.94	99.24
SRNN	Word embeddings from tokenized text sequences	94.27	93.24	95.43	94.32
LSTM	Word embeddings from tokenized text sequences	99.84	99.91	99.77	99.84

TABLE I
Some Quantitative Evaluation metrics of different models used in our project

2) Qualitative Evaluation: Here, in Figure 3, the graph shows the average accuracy of all trained machine learning and deep learning algorithms. Overall, the algorithm that showed the highest accuracy was the LSTM model (accuracy 99.84%), whereas the algorithm that showed the least accuracy was the simple RNN model (accuracy of 94.27%). However, only considering accuracy measurement alone can't give us the full picture to evaluate the performance of a model. In addition to calculating accuracy, we also calculated recall, precision, and F1-score for each of the training models as well.

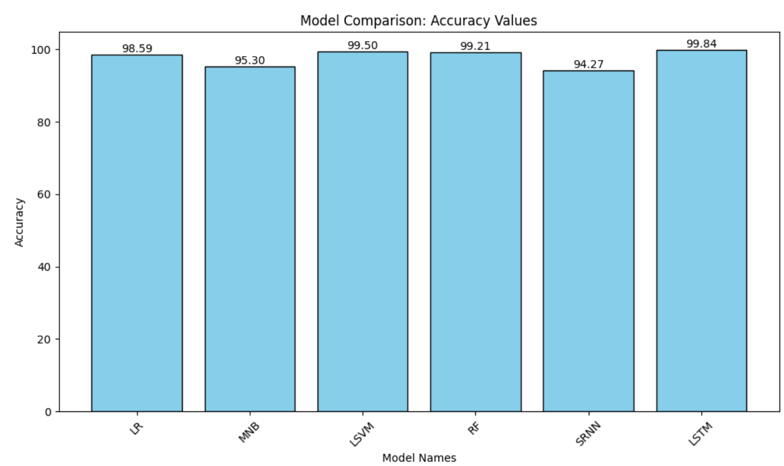

Fig. 3. Accuracy Comparison of all models

Therefore, comparing all the values we calculated for all of our models, among machine learning models, the Linear SVM model has shown better output. But with neural network architecture and using NLP, among all the models, the LSTM model has the best performance, also showing better results in accuracy, precision, recall as well as F1 score. So, from our trained models, the LSTM model can be used to detect fake news more accurately for our used dataset.

C. Learning Curves

A learning curve is a plot that shows how a machine learning model's performance changes over time or experience (epochs, number of training samples). There are two types of learning curves- optimization learning curve and performance learning curve.

Figure 4 shows the performance learning curve of our deep learning models. The graph shows accuracy as the number of samples increases. We see from the graph that for logistic regression both for training and testing, the accuracy rises as the sample number increases. However, the test accuracy is always lower than the training accuracy as it should be. For the random forest model, we see training performing almost the same as the training sample increases and for the testing the accuracy increases steadily as the number of samples increases. Looking at the naive Bayes model, we see an interesting scenario where for training accuracy decreases as training samples increase, and for testing, we see the accuracy rising as sample count increases. We see for training of linear SVM as sample count increases at first accuracy drops but rises again as more samples are needed and for test, we also see the trend of accuracy increasing as sample count increases.

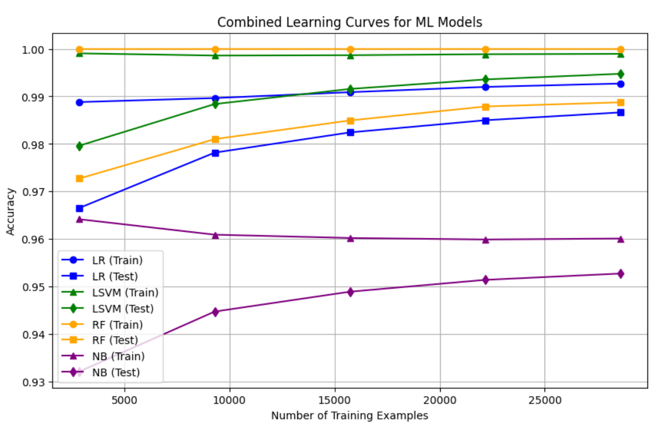

Fig. 4. This graph shows the learning curves for the machine learning models

Figure-5 shows the performance learning curve of our deep learning models. The graph shows the increase of accuracy with the number of epochs conducted. The simple RNN model ran for twelve epochs and shows the ideal scenario where the training accuracy is higher than the test accuracy most of the time. The LSTM model only ran for three epochs and also shows that at first test accuracy was higher than training accuracy but it reached the ideal scenario from the second epoch.

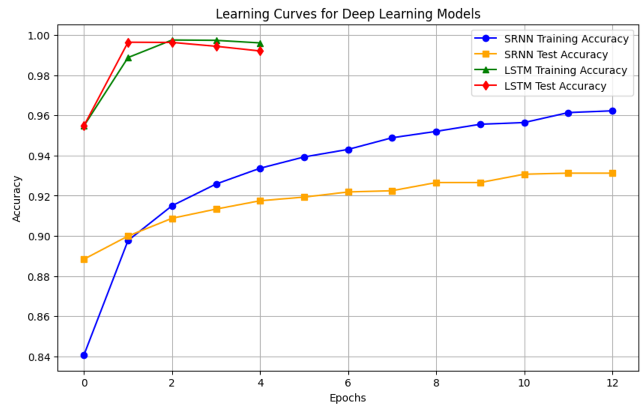

Fig. 5. This graph shows the learning curves for the deep learning models

D. Receiver operating characteristic (ROC) Curve

The term Receiver Operating Characteristic (ROC) curve refers to a graphical representation that illustrates the performance of a binary classification model (also applicable for multi-class classification) across different probability thresholds. The ROC curve is the graphical representation of the true positive rate (TPR) against the false positive rate (FPR) at each threshold value.The AUC represents the area under the ROC curve. AUC values range from 0.5 (random guessing) to 1.0 (perfect classification).

In Figure-6, we see the ROC AUC curve for the machine learning models. We see that for all of the models except naive Bayes, the ideal case where the curve is touching the top left corner indicates a high true positive rate (TPR) and a low false negative rate (FPR) is visible. We can also see that the best ROC curve order is: Linear SVM, random forest, logistic regression, and naive Bayes respectively due to their relative position at the FPR = 0 points. As for the area under the curve (AUC) values, all models give perfect classification (AUC = 1) except the naive Bayes model with AUC = 0.99 which is close to 1 indication of good classification (less FPR).

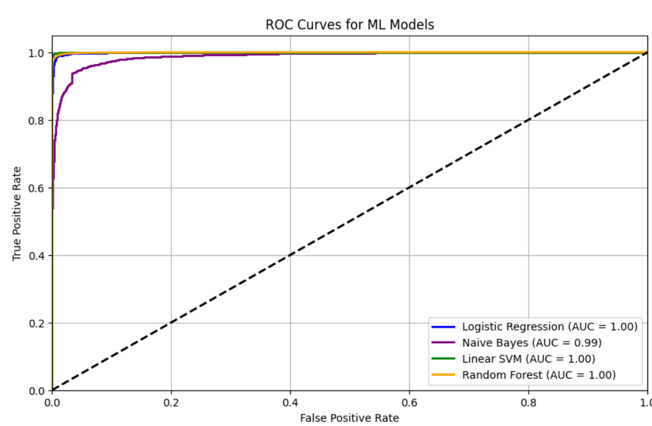

Fig. 6. This graph shows the ROC AUC curves for the machine learning models

In Figure 7, we see the ROC AUC curve for the deep learning models. For the simple RNN model, as FPR increases, the TPR rises steeply. At very low FPR, SRNN achieves a TPR close to 1, indicating excellent model performance. The AUC for SRNN is 0.98. Compared to SRNN, the LSTM model exhibits a sharp rise in TPR as FPR increases. However, LSTM's FPR is slightly higher than SRNN's. The AUC for LSTM is also 0.98. Both the SRNN and LSTM significantly outperform randomness close to perfect classification (AUC = 1). So, both models show high-class discriminatory power but, LSTM is more accurate quantitatively as seen in Table I.

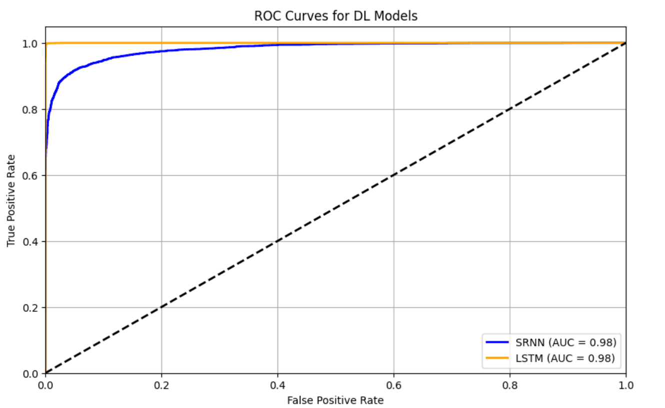

Fig. 7. This graph show the ROC AUC curves for the deep learning models

IV. Prototype

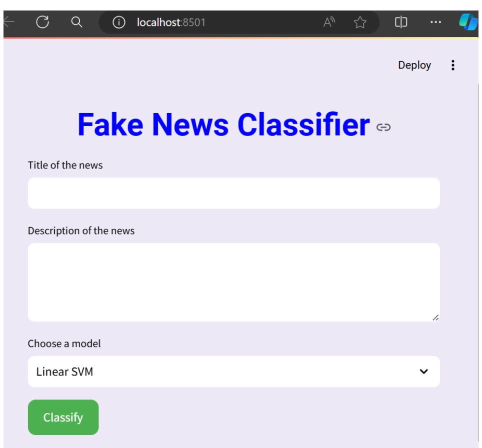

Fig. 8. UI of our application

Here, figure 8 shows the user interface for our fake news classifier application. For making the prototype of our app, we have used Steamlit library. Streamlit is an open-source python framework used for data science, AI, or ML to create interactive apps in a few minutes. If we change any input or something is updated on the screen, Streamlit reruns the whole project automatically without having to run the application from the terminal. So, for this reason, we have used this Python-supporting library in order to make our user interface.

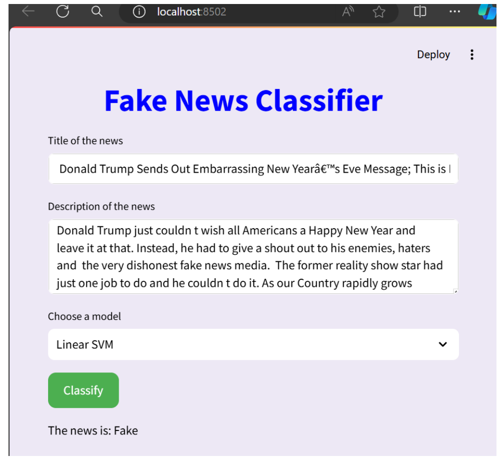

Fig. 9. Output prototype of our application

In figure 9, we have given input from our dataset, after that, the output shows if the news is true or fake.

We need to give both the title and description as inputs, otherwise, it will show an error and request to give both as inputs shown in figure-10.

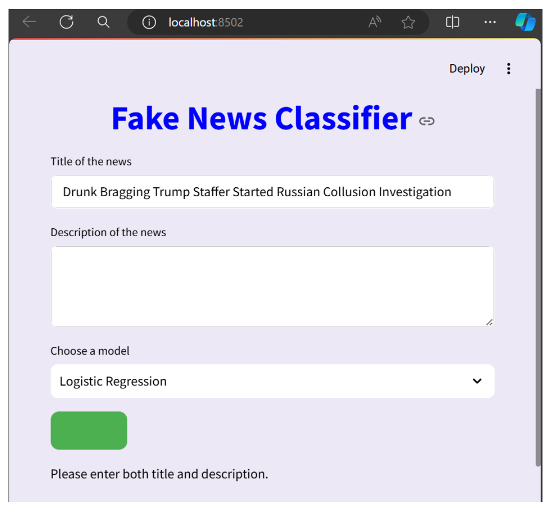

Fig. 10. Error shown for incorrect inputs

Now, we have also used a CSS file to style our web page and to provide some custom font colors, styles, and font families. For more customization, we also changed the theme in the configuration of Streamlit library. In this way, we tried to display our web page in a unique style different from the usual streamlit apps' designs.

V. Conclusion and Future Work

In conclusion, we tried to use both machine learning and deep learning models effectively to classify the news from our dataset. Here, we have used a previously compiled dataset to train our model. After analyzing all the models we trained, Linear SVM among the machine learning model performed the best whereas considering every metric among all the models the best was the LSTM model.

Fake news has now become a serious issue which has attracted many researchers to do their studies on this topic to stop people from being misinformed. Our model has been trained with limited algorithms as well as on only a particular dataset. The best deep learning model's efficiency in processing news titles makes it suitable for real-time applications, where quick user responses are crucial. However, for applications where high accuracy is necessary, models trained with comprehensive news content are preferable. For future work on our app, we are willing to fine-tune model parameters and explore advanced architectures for further improvements. Additionally, we are planning to incorporate data in different formats, such as images and videos, and exploring new datasets and also expanding the dataset to include multiple languages will enhance the robustness and applicability of fake news detection systems. Besides, we want to set up our application on the mobile platform as well to make it more accessible and easy to use for users.

References

[1] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake news detection using machine learning ensemble methods," *Complexity*, vol. 2020, no. 1, p. 8885861, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2020/8885861

[2] N. K. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015. [Online]. Available: https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/pra2.2015.145052010082

[3] H. Ahmed, I. Traore, and S. Saad, "Detection of online fake news using n-gram analysis and machine learning techniques," in *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, I. Traore, I. Woungang, and A. Awad, Eds. Cham: Springer International Publishing, 2017, pp. 127–138. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-69155-8_9

[4] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, "Fake news detection on social media using geometric deep learning," *ArXiv*, vol. abs/1902.06673, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:62841478

[5] J. A. Nasir, O. S. Khan, and I. Varlamis, "Fake news detection: A hybrid cnn-rnn based deep learning approach," *International Journal of Information Management Data Insights*, vol. 1, no. 1, p. 100007, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667096820300070

[6] A. Thota, P. Tilak, S. Ahluwalia, and N. Lohia, "Fake news detection: A deep learning approach," 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:68219159

[7] A. Abdulrahman and M. Baykara, "Fake news detection using machine learning and deep learning algorithms," in *2020 International Conference on Advanced Science and Engineering (ICOASE)*, 2020, pp. 18–23. [Online]. Available: https://ieeexplore.ieee.org/document/9436605

[8] J. Alghamdi, Y. Lin, and S. Luo, "A comparative study of machine learning and deep learning techniques for fake news detection," *Information*, vol. 13, no. 12, 2022. [Online]. Available: https://www.mdpi.com/2078-2489/13/12/576

[9] W. H. Bangyal, R. Qasim, N. u. Rehman, Z. Ahmad, H. Dar, L. Rukhsar, Z. Aman, and J. Ahmad, "Detection of fake news text classification on covid-19 using deep learning approaches," *Computational and Mathematical Methods in Medicine*, vol. 2021, no. 1, p. 5514220, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/5514220

[10] S. Kumar, R. Asthana, S. Upadhyay, N. Upreti, and M. Akbar, "Fake news detection using deep learning models: A novel approach," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 2, p. e3767, 2020, e3767 ETT-19-0216.R1. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3767

[11] D. Saha, A. Das, T. C. Nath, S. Saha, and R. Das, *Detection of Fake News and Rumors in Social Media Using Machine Learning Techniques With Semantic Attributes*. John Wiley Sons, Ltd, 2022, ch. 4, pp. 85–97. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119857686.ch4

[12] M. Madani, H. Motameni, and H. Mohamadi, "Fake news detection using deep learning integrating feature extraction, natural language processing, and statistical descriptors," *SECURITY AND PRIVACY*, vol. 5, no. 6, p. e264, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/spy2.264

[13] S. Mishra, P. Shukla, and R. Agarwal, "Analyzing machine learning enabled fake news detection techniques for diversified datasets," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 1575365, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/1575365

[14] T. Li, J. Yu, and H. Zhang, "Web of things based social media fake news classification with feature extraction using pre-trained convoluted recurrent network with deep fuzzy learning," *Theoretical Computer Science*, vol. 931, pp. 65–77, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304397522004467

[15] S. K. Hamed, M. J. Ab Aziz, and M. R. Yaakub, "A review of fake news detection approaches: A critical analysis of relevant studies and highlighting key challenges associated with the dataset, feature representation, and data fusion," *Heliyon*, vol. 9, no. 10, p. e20382, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405844023075904

[16] L. Huang, "Deep learning for fake news detection: Theories and models," in *Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering*, ser. EITCE '22. New York, NY, USA: Association for Computing Machinery, 2023, p. 1322–1326. [Online]. Available: https://doi.org/10.1145/3573428.3573663

[17] A. Seyam, A. Bou Nassif, M. Abu Talib, Q. Nasir, and B. Al Blooshi, "Deep learning models to detect online false information: A systematic literature review," in *The 7th Annual International Conference on Arab Women in Computing in Conjunction with the 2nd Forum of Women in Research*, ser. ArabWIC 2021. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3485557.3485580

[18] N. F. Baarir and A. Djeffal, "Fake news detection using machine learning," in *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)*, 2021, pp. 125–130.

[19] Z. Khanam, B. N. Alwasel, H. Sirafi, and M. Rashid, "Fake news detection using machine learning approaches," *IOP Conference Series: Materials Science and Engineering*, vol. 1099, no. 1, p. 012040, mar 2021. [Online]. Available: https://dx.doi.org/10.1088/1757-899X/1099/1/012040

[20] S. Pandey, S. Prabhakaran, N. V. S. Reddy, and D. Acharya, "Fake news detection from online media using machine learning classifiers," *Journal of Physics: Conference Series*, vol. 2161, no. 1, p. 012027, jan 2022. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/2161/1/012027