Journal of Information Technology and Digital World (ISSN: 2582-418X)
www.irojournals.com/itdw/

Empirical Analysis for Classification of Fake News through Text Representation

Dr Ilango Krishnamurthi[1], Dr Santhi V[2], Madhumitha N H[3]

[1]Visiting Professor, Department of Computer Science and Engineering, PSG College of Technology, Coimbatore

[2]Professor, Department of Computer Science and Engineering, PSG College of Technology, Coimbatore

[3]PG Scholar, Department of Computer Science and Engineering, PSG College of Technology, Coimbatore

E-mail: [1]ikm.cse@psgtech.ac.in, [2]vsr.cse@psgtech.ac.in, [3]22mz02@psgtech.ac.in

Abstract

Fake news refers to inaccurate or deceptive information that is portrayed as legitimate news. It is intentionally generated and disseminated to mislead the public. Fake news takes on multiple forms, including altered visuals, invented narratives, and misrepresented accounts of actual occurrences, although this work focuses solely on textual content. Initially, the focus of this work is to evaluate various pre-processing techniques involved in fake news detection, such as TF-IDF, GloVe, and Integer Encoding. Each of these techniques has its own way of converting text to numerical format. Despite numerous studies in this field, there is still a research gap regarding the comparative analysis of TF_IDF (Term Frequency Inverse Document Frequency), Integer Encoding, and GloVe (Global Vector for Word Representation) specifically for fake news tasks. This study aims to bridge this gap by evaluating and comparing the performance of these three popular preprocessing techniques. Next, three RNN variants are used in this experiment for the classification task. They are SimpleRNN (Simple Recurrent Neural Network), LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit). The reason behind choosing RNN variants is RNN is capable of capturing long term dependencies.

DOI: https://doi.org/10.36548/jitdw.2024.1.003
Received: 09.01.2024. received in revised form: 04.02.2024, accepted: 22.02.2024, published: 06.03.2024

It is proven to be effective in handling sequential data. It consists of memory that stores the previous important content. GloVe showed high accuracy in GRU model, and it also used only less computational resources, but LSTM took more time and required more computational resources. The results produced by GRU and LSTM for GloVe were better than the rest of the combinations. Integer Encoding also produced good results. But TF-IDF gives poor results when fed to Deep Learning models like RNN, LSTM, and GRU, but when it is fed to Machine Learning Model it gives good accuracy. This is due to sparse matrix generation based on the importance of term frequency. The findings highlight the advantages and limitations of each algorithm, providing valuable guidance for researchers and practitioners in choosing the suitable method for their specific needs. The experimental finding of this work is that GloVe with GRU produces the highest accuracy of 92.15%

Keywords: GloVe, TF-IDF, RNN, GRU, LSTM, NLP

1. Introduction

The spread of inaccurate data poses a significant risk to communities as it undermines trust in the media, distorts individuals' perceptions of the truth, and creates division among societies. The prevalence of social media and internet-based news outlets in recent times has bestowed the spread of fake information and deceptive content. The generation of fake news has been on a steady incline, as evidenced by a study conducted by the NCRB (National Crime Records Bureau), revealing a 214% surge in the generation of fabricated news in 2020. This study, aims to explore and compare three pre-processing techniques, such as TF-IDF, Integer Encoding, and GloVe, involved in NLP tasks for fake news detection, along with three DL Algorithms, such as LSTM, GRU, and RNN. The Fake News Challenge Dataset from Kaggle [21] is widely used in fake news detection and will serve as a pillar for conducting our experiments. This research evaluates various techniques involved in fake news detection, which is crucial for several reasons. First and foremost, it helps researchers and developers understand the performance and limitations of different methods in tackling the complex challenge of fake news detection. By rigorously assessing these techniques, we can gain insights into their strengths and weaknesses, thus paving the way for the improvement and advancement of fake news detection technologies.

The relevance of a term within a document in a collection or corpus can be assessed statistically using the TF-IDF measure [27]. By considering both the term's frequency inside the document (TF) and its occurrence throughout the corpus (IDF), it calculates a word's value in relation to the entire corpus. Each word in the corpus is given a distinct integer as part of the integer encoding process. Frequently, this method is employed as a first step before implementing deep learning models. Deep learning algorithms can process and analyze data more efficiently by turning language into numerical form. By converting textual data into an understandable format for algorithms, integer encoding makes it easier to train models for the detection of fake news. GloVe is a technique for unsupervised learning that constructs word vector representations. These vector representations effectively represent the linguistic associations between words when they consider the overlap of words in a particular corpus [26]. GloVe embeddings can be used to capture the context and underlying semantics of words in the context of fake news detection.

A specific kind of neural network called RNNs is made to process sequential data by using feedback loops. [25]. Their capacity to retain information from previous inputs enables them to make informed decisions based on the broader context of the entire input sequence. LSTM, a form of RNN, was introduced to address the gradient descent problem and is capable of recognizing and learning long-term dependencies. LSTMs incorporate a memory cell that facilitates the retention of information over extended periods, making them well-suited for tasks involving the analysis of sequential data. GRUs, on the other hand, represent a simplified version of the conventional RNN architecture, aiming to streamline the design and training process while still retaining the ability to capture temporal dependencies.

In this study, a comparison of the performance of various pre-processing techniques along with various types of RNN models is carried out. The aim is to identify the combination of the most effective RNN variants and pre-processing techniques for fake news detection.

2. Related Works

2.1 Machine Learning Models

The Buzzfeed dataset, which contains the facebook data is used in [1], and it achieves an accuracy of 79%. It estimates the distance that exists among every point in the training

dataset and the object that needs to be classified. K values of 15 and 20 has achieved maximum accuracy. The researchers mainly used the share count, comment count, and like count to predict the fake news. The data related to Pakistan was gathered. Sentimental and lexical analysis is done for machine learning algorithms. Among Machine learning algorithms KNN achieves a highest accuracy of 89% [2]. The studies [4] and [5] use three distinct datasets—the Corpus, Fake News, and Liar datasets—to examine the effectiveness of several ML and DL approaches for the detection of fake news. The study clearly gave insights about feature extraction techniques and the approaches used in fake news detection. Naïve Bayes gives an accuracy of 93% among all the ML algorithms. In [6], in order to identify the appropriate vectorizer for detecting fake news, a score of two vectors is obtained by comparing its counts and term frequency with document frequency. The highest accurate prediction was made by SVM using the TF-IDF, according to the results. In research [7], a Twitter dataset was used, and the accuracy measures of the Decision Tree appear to be 97.67%, which is better than the accuracy of the SVM method with the accuracy of 91.74 %. In [8], a hypothesis was proposed that there may be a relationship between false news and the content of texts published over the internet. World Cloud visualization technique after cleaning the text was applied. PHEME dataset consisting of Twitter data, was used. Latent Semantic Analysis and Latent Dirichlet Allocation were used to calculate the semantic score. Naïve Bayes with lidstone smoothening is used in [9]. A survey of 40 papers was presented in paper [11]. In the paper [14], the dataset used is the Kaggle dataset. TF-IDF technique is used for feature extraction. J48, which is based on C4.5 algorithm, is used, and it achieves 89.11% over a random forest of 84.97% accuracy. It utilizes the Spark cluster to create an ensemble model. The results show that the ensemble model has 92.45% of the F1 score [10].

From [1], they have limited their work to KNN algorithm alone, but the focus of this work, is incorporating various DL algorithms. In [4] and [5] various ML algorithms are compared and a final conclusion was made that Naive Bayes performs well among all the ML algorithms. In [6] and [7] came to know about TF-IDF and word cloud techniques in which TF-IDF is going to be incorporated in this work. Paper [10], gave insights about ensemble model and spark architecture but this doesn’t help this work because Spark architecture is not going to be used in this model.

2.2 Reinforcement Model

In paper [12], the model is trained on one source and targeted on another source. The dataset used in this paper is FakeNewsNet. To generate suitable text content, they adjust the model based on the BERT and HAN networks. Both the fake news model and the domain model are trained. For training agents, the real news classification and domain classification are used as a reward function in the RL setting. It achieves an accuracy of 84% in the Politifact dataset.

From [12], I came to know that a domain-adaptive model can be built, but in this paper, training the model takes place in the general domain, so it doesn't help this work.

2.3 Deep Learning Models

In Pakistani dataset, GloVe and BERT embeddings are used for DL algorithms. Models are compiled with an Adam optimizer that sets the learning rate to 0.001 and binary cross entropy is used for loss functions. LSTM achieves an accuracy of 94% under GloVe feature extraction [2]. Sequential Neural Network is used in [3], and a linguistic model is used to find out the properties of content. It extracts syntactical (word count, title count, word density), sentimental (polarity, subjectivity), grammatical (pos), and readability (complexity of vocabulary). To calculate the readability score, selenium is used as a unit test library. The dataset used is Horne2017_FakeNews data and the accuracy of the model is 86%. Deep neural networks that composed of seven layers and uses Sigmoid activation functions to produce an output that is either true or incorrect is used and ReLU activation function is used by the Dense layers contained [9]. The dataset used in this paper is ISO and FAKES. Word2Vec preprocessing technique is used to convert words into vectors in [13] and [18] where, Covid dataset was used. In order to extract the features of the word vector, the CNN model is used, and the CNN output is fed into the RNN classification model. The accuracy of the model is 99% ISO and 60% in FA-KES dataset. In [15], Bi LSTM allows look for a particular sequence in both front to back and back to front. RNN's performance in short context news articles is excellent. In this paper, an optimization of the Adam algorithm has been used and is achieving a 99% accuracy. In order to detect fake news, this research suggests using a deep convolutional neural network (FNDNet). In order to detect false news, this paper proposes a deep convolutionally based network called theFNDNet. The authors demonstrate that the suggested

model has an accuracy of 98.36% on the test data of [16] and [17] respectively by comparing its performance to a number of baseline models. The paper uses the RumorEval19 dataset, composed of social media conversations on rumors. Dual emotion features are extracted from news content and user comments. The attention-weight update mechanism is used to learn the importance of each dual emotion feature. In [19], Genetic search is utilized to initialize the population, select, recombine, and evaluate the fitness function in order to determine the optimal set of hyper parameters. The dataset used is available in Kaggle. Word cloud, n grams are used for feature extraction. Four-layer DNN is used in this research. Adam is used as an optimization algorithm to reduce the error in the model. The accuracy of the model is 74.1%. In paper [20], the dataset used is WelFake dataset and a hybrid CNN and Bi LSTM is used to classify the news and achieves an accuracy of 97%.

From [2], came to know about GloVe and BERT pre trained word embeddings model from which GloVe technique will be incorporated as feature extraction technique in the DL models. In [9], [13] and [16] more complex neural network can be used to achieve a better performance and in [15], LSTM was used which was very efficient in terms of detecting the fake news so thought of incorporating LSTM in this work. In [17], the authors gave importance to dual emotion which act as good parameter so sentimental analysis is going to be incorporate in this work. In [20], they have limited their work to the WelFake dataset but in this work, the dataset going to be used is the dataset from [19].

3. System Design

The Models to be involved in this detection of fake news system is shown in Fig 1. The dataset used in this work is Kaggle Fake News Challenge Dataset. The news articles in the dataset is cleaned. The cleaning process involves removal of null rows, special characters, stopwords and lemmatizing the words to convert the word to its base form. This is done with the help of NLTK package. Section 4 presents the description in detail.

Next comes the pre-processing technique such as TF-IDF, Glove and Integer Encoding. These techniques are used to convert the text into numerical format based on certain criteria. A detailed working of these techniques is provided in Section 4. Each of these techniques has its own advantages and disadvantages. The last stage involves providing these numerical values as input to different Deep Learning methods, including RNN, LSTM, and

GRU. The reason behind choosing RNN variants is RNN is capable of capturing long term dependencies. It is proven to be effective in handling sequential data. It consists of memory that stores the previous important content.

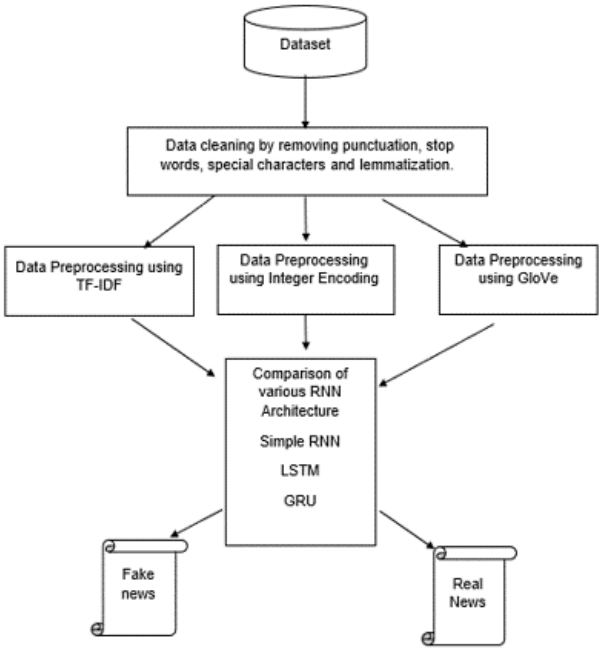

Figure 1. Flow of Proposed Design

Fake news detection often involves analyzing text data, which can be sequential in nature. Text context and meaning can be greatly enhanced by recognizing long-range dependencies and patterns in sequential data, which can be effectively captured by RNN variations. RNN variants are effective at automatically learning and extracting relevant features from textual data. They can capture semantic relationships and context within the text, which is essential for differentiating between real and fake news. Fake news detection often requires understanding complex linguistic structures and context. RNN variants can capture intricate relationships between words and phrases, which can aid in identifying patterns associated with fake news propagation. The architecture and working of these algorithms are explained in Section 5. The activation function used is sigmoid and the result will be produced in range 0 to 1. If the result is 0 then the news falls under real news and if the result is 1 then the news comes under fake news.

4. Data Preprocessing

From the system design it is very clear that the initial step involved in fake news classification is data pre-processing. The news articles cannot be fed as direct input into the deep learning models. These news articles have to be cleaned and converted into numerical format with the help of NLP techniques. The following are the steps involved in data cleaning.

4.1 Data Cleaning

Step 1: Convert the News Article to Lowercase

This step converts all text in the 'text' column to lowercase. Lowercasing ensures that text is treated consistently, as it makes all words in lowercase.

Step 2: Special Character Removal

This step uses regular expressions ('regex=True') and removes the special characters and non-alphabetical characters from the text. This effectively removes special characters, punctuation, and numbers, leaving only alphabetic characters, and spaces.

Step 3: Tokenization

This step converts the words into small chunks called tokens. This is done for performing stopwords removal and lemmatization. Tokenisation is very crucial and they are done with the help of NLTK packages.

Step 4: Stop word Removal

This step define a set of English stop words using NLTK's 'stopwords.words('english')'. Stopwords are common words (e.g., "the," "is," "in") that are often removed from text data because they don't carry significant meaning.

Step 5: Lemmatization

In this step the words are reduced to their base forms, which helps in standardizing the text and reducing dimensionality. Instead of stemming, lemmatization is chosen because in stemming the words are truncated and results in meaningless words.

Step 6: Handling Missing Values

Dealing with missing data points is critical to prevent biases and ensure that the dataset is complete. So text column with empty or null values are identified and removed.

These data cleaning steps are applied to the text column of our dataset. After data cleaning, TF-IDF, Integer Encoding, GloVe Techniques are used for numerical conversion of news articles. The text column is converted into numerical vectors which are fed into RNN variants.

4.2 TF-IDF Technique

This is a pre-processing approach for converting text into numerical form, based on the calculation of the importance of the term in the document and in the entire document collection. The words with a high number of TF-IDF are strongly linked to the document it refers to. Term Frequency measures the frequency of a term within a document relative to the total number of terms in that document. Inverse Document Frequency measures the rareness of a particular word throughout the corpus.

$$w_d = f_{w,d} * \log\left(\frac{|D|}{f_{w,D}}\right) \quad (1)$$

where;

w_d is the TF_IDF w is term weight in document d. $f_{w,d}$ is the frequency of the word in the document, $f_{w,D}$ is the frequency of the word in the entire document corpus, |D| is the total number of document in the corpus. TF-IDF can help identify words that are distinctive to either real or fake news articles, enabling the classification of articles based on the uniqueness of their content.

4.3 Integer Encoding Technique

Integer encoding is a procedure that assigns unique integer values to words or tokens within a text corpus, serving as a foundational step in text data preparation for machine learning and deep learning applications. Despite its simplicity and effectiveness, integer encoding alone does not capture the intricate semantic associations between words or the contextual nuances embedded within the text. For tasks that require a more profound comprehension of language,

approaches like word embeddings and deep learning architectures are often employed to capture the complex linguistic relationships and meanings within textual data. Integer encoding is a simple and effective method for preparing textual data for tasks such as natural language processing, sentiment analysis, and fake news detection.

4.4 GloVe Technique

Word embeddings, also known as dense vector representations of words within a corpus, are generated using the unsupervised learning algorithm GloVe (Global Vectors for Word Representation). Unlike certain other methods of word embedding, GloVe considers the global statistics of the corpus and the co-occurrence of words, aiming to capture both the overall word co-occurrence information and the local context. GloVe constructs a low-dimensional vector space where words sharing similar context and usage are positioned closer to each other, leveraging analysis of the word co-occurrence matrix. GloVe embeddings have proven their utility across various natural language processing applications, particularly in text analysis. Traditional text representations like one-hot encoding, TF-IDF are high-dimensional and lack semantic meaning. The 100d GloVe file developed by Stanford university is used in this work.

5. Proposed Methodology

After pre-processing and converting the text data into numerical format three popular Deep Learning Techniques have been used namely RNN, LSTM and GRU for classifying fake news articles.

5.1 RNN

It sequentially processes the input data and the recurrent propagation of information through time. At each time step, the RNN takes an input vector and combines it with the information from the previous time step to generate an output and update its internal state. This recurrent structure allows RNN to consider the entire input sequence and learn from the historical context within the sequence. RNN are able to extract dynamic patterns from data by using a range of learning parameters, which can be used for the adaptive processing of periodic information. The formula to the hidden and output layer is given in equation (2) and (3).

$$h_t = act\_func[(W_{hh}h_{t-1} + W_{xh}X_t) + bias \quad (2)$$

$$y_t = W_{hy} h_t \quad (3)$$

h_t is current state, h_{t-1} is previous state, X_t is input state, W_{hh} - weight of recurrent neural network, W_{xh} - weight of input neuron, W_{hy} - weight of output layer, y_t- output layer, act_func - activation function (tanh, Sigmoid, ReLu).

5.2 LSTM

Information is transferred across multiple interconnected memory cells, each of which has three major gates—an input gate, an output gate, and a forget gate—during the functioning of the LSTM Network. These gates decide which information should be output, forgotten, or controlled in terms of information flow within the memory cell. The input gate decides how much new data should be fed to the memory cell at each step, and the LSTM uses the input as well as the memory cell's prior state as input. The output gate will decide which information should be sent on to the next phase, while the forget gate will decide which information from the previous memory cell state should be erased. Equations (4), (5), (6), and (7) provide the formulas for the new information, cell state, output gate, and hidden state, respectively.

$$g_t = \tanh\left[\left(W_{gh} h_{t-1} + W_{gx} x_t\right) + bias\right] \quad (4)$$

$$c_t = c_{t-1} * f_t + i_t * g_t \quad (5)$$

$$o_t = \sigma[(W_{oh} h_{t-1} + W_{ox} x_t) + bias] \quad (6)$$

$$h_t = \tanh(c_t) * o_t \quad (7)$$

where;

h_{t-1} – previous hidden state, h_t – hidden state, $W_{fh}, W_{ih}, W_{gh}, W_{oh}$ - weight associated with hidden state, o_t – output gate, c_t – current cell state, c_{t-1} – previous cell state, x_t – current input, $W_{fx}, W_{ox}, W_{ix}, W_{gx}$ – weight associated with input, i_t – input gate, g_t – new information.

5.3 GRU

The working of a GRU involves the iterative processing of sequential data through a series of interconnected units, each of which contains a reset gate and an update gate. These gates regulate the flow of information, and at each step in time they check that there is an

update to the internal state. The reset gate determines how much previous information should be forgotten at each step, while the update gate decides how much new information is to be included in the internal state. GRUs are designed to efficiently capture and retain relevant information over extended sequences, making them adept at learning and representing complex temporal patterns and dependencies within the data. The formula behind the working of reset and update gate is given in equation (8), and (9) respectively.

$$r_t = \sigma(X_t W_{rx} + h_{t-1} W_{rh} \quad (8)$$

$$u_t = \sigma(X_t W_{ux} + h_{t-1} W_{uh}) \quad (9)$$

where;

r_t – reset gate, X_t – current input, h_{t-1} – previous hidden state, W_{rx}, W_{ux} – weight associated with input, W_{rh}, W_{uh} - weight associated with hidden state.

6. Dataset Details

The dataset used is Fake News Challenge dataset. This dataset is available in Kaggle [21]. Its size is 98.63MB. It has 20,799 news articles. The count of false and original news is 10,413 and 10,387 respectively. Since the count of fake and real news articles is close enough the dataset is not biased.

The dataset contains 5 columns they are; Id (Serial number), Title (news headlines), Author (writer of the news), Text (about the news), Label (1-fake news, 0-real news). The dataset for Fake News Detection is shown in Fig 2. and Fig 3.

id	title	author	text	label
0 – 20.8k	[null] 3% The Dark Agenda B... 0% Other (20237) 97%	nan 9% Pam Key 1% Other (18600) 89%	20387 unique values	0 – 1
0	House Dem Aide: We Didn't Even See Comey's Letter Until Jason Chaffetz Tweeted It	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Letter Until Jason Chaffetz Tweeted It By Darrell Lucus o...	1
1	FLYNN: Hillary Clinton, Big Woman on Campus - Breitbart	Daniel J. Flynn	Ever get the feeling your life circles the roundabout rather than heads in a straight line toward th...	0
2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, 2016 The tension between intelligence analysts and po...	1

Figure 2. Dataset for Fake News Detection

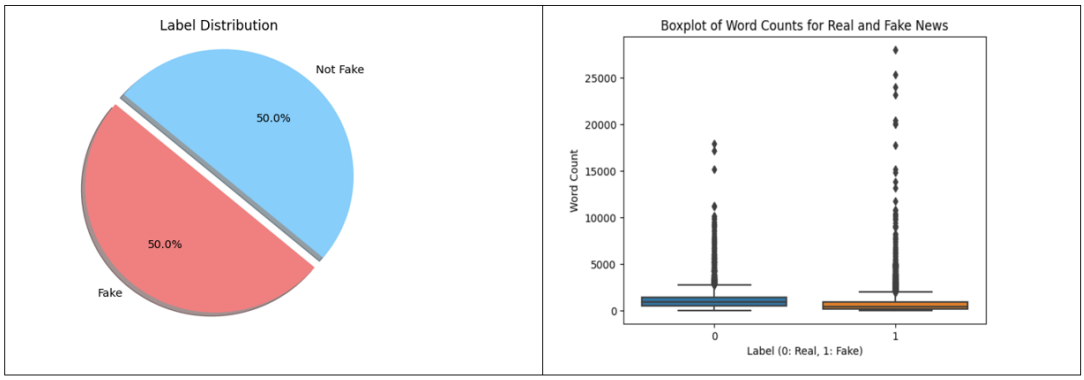

Figure 3. Distribution and Word Count for Real and Fake News Articles

7. Implementation and Results

The dataset is read with the help of pandas and tensorflow consist of integer encoding library. Once the numerical format is stored in list, we are padding it upto 100 words and storing it as embedded_docs. This embedded_docs is fed into the Embedding layer and the result is fed into SimpleRNN with 128 neurons.

The text column in the Fake News Challenge dataset is converted into a numerical format based on the frequency of the term in the document and in the entire document corpus. The max number of features chosen is 1000. “Scikit-learn is a well-known open-source library for the Python programming language. It consists of tfidfvectorizer which is used to convert

the text into numerical format. It is converted into array since the compressed sparse matrix cannot be fed directly into the SimpleRNN model".

The text column in the dataset is converted into numerical with the help of tokenizer. It is padded to a length of 100. Tensorflow.keras.preprocessing consist of tokenizer and pad_sequences function. From the tensorflow.keras.models sequential model can be imported and tensorflow.keras.layers RNN can be imported. GloVe 100d is used as a weight to the RNN model. The words present in the dataset is extracted and stored in embedding_matrix. A sequential model with embedding layer, RNN layer and a dense layer is used. Activation function used is sigmoid and Adam Optimizer is used. The same parameters were used except Simple RNN, LSTM and GRU model were used.

The accuracy, F2 score and Mean Average Precision comparison for the three RNN variants such as Simple RNN, LSTM, and GRU is shown in Fig 4.

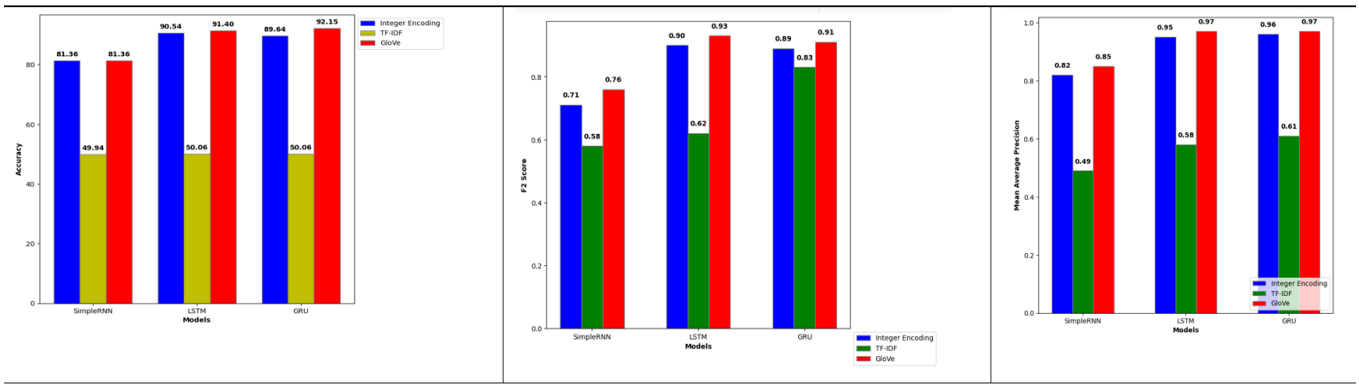

Figure 4. Accuracy, F2 Score and Mean Average Precision Comparison for RNN Variants along with Three Pre-Processing Techniques

The results of all the three preprocessing techniques along with the three deep learning models is shown in the below Table 1.

Table 1. Results of Various RNN Variants along with Pre-processing Technique

S.NO	Pre-Processing Technique	Model	Accuracy	F2 Score	Mean Average Precision
1	Integer Encoding	Simple RNN	81.36%	0.71	0.82
2	Integer Encoding	LSTM	90.54%	0.90	0.95
3	Integer Encoding	GRU	89.64%	0.89	0.96
4	TF-IDF	Simple RNN	49.94%	0.58	0.49
5	TF-IDF	LSTM	50.06%	0.62	0.58
6	TF-IDF	GRU	50.06%	0.83	0.61
7	GloVe	Simple RNN	81.36%	0.76	0.85
8	GloVe	LSTM	91.40%	0.93	0.97
9	GloVe	GRU	92.15%	0.91	0.97

From Table. 1 it is obvious that GloVe has achieved high accuracy due to its ability to capture semantic relationship between the words. Among RNN variants, GRU achieved high accuracy and computationally effective. TF-IDF gives less accuracy because it gives compressed sparse matrix as output which cannot be fed directly into Deep Learning model, so the CSR matrix is converted into a dense array. But the results were low in all the three Deep Learning models. Lastly, Integer Encoding, which represents words as unique integers, was the simplest technique but delivered competitive results, highlighting its efficiency in computational terms.

In order to tune the hyperparameters in this project different test sizes of ratio 0.2, 0.3, 0.4, and 0.5 were chosen and model accuracy was measured. A graph indicating the model accuracy along with the test size is shown in Fig 5.

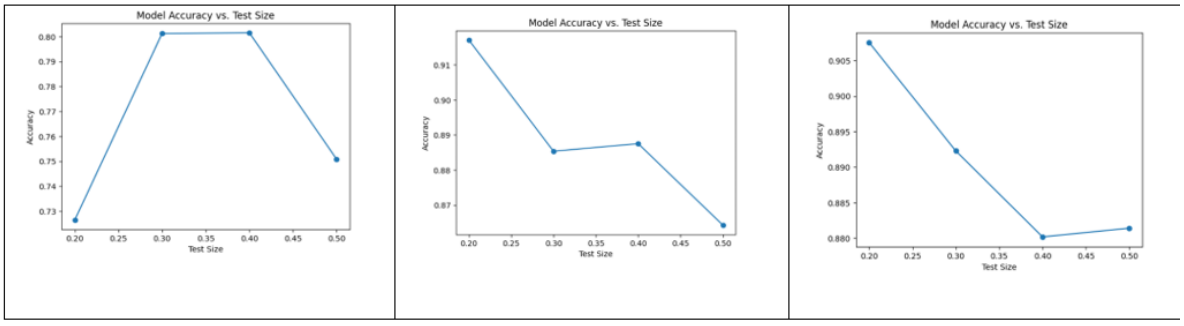

Figure 5. Model Accuracy Vs Test Sizes of Ratio 0.2, 0.3, 0.4, and 0.5 for Simple RNN, LSTM and GRU

In each test size, 10 epoch and a batch size of 64 was chosen in this project. The epoch count and batch size can be varied. From Fig 5 it is very clear that a test_size of 0.2 gives high accuracy and as the test size decreases the accuracy falls down to a greater extent.

8. Conclusion and Future Work

In this work, various preprocessing techniques such as TF-IDF, GloVe, and Integer Encoding have been compared along with different DL architectures such as LSTM, RNN, and GRU. The results obtained show that GRU is computationally effective by achieving a good accuracy of 92.15% under the best computational time under T4 GPU runtime environment. This is because, in contrast to LSTM, which uses three gates—input, forget, and output—GRU only requires two gates—the update and reset gates. Among the preprocessing techniques, GloVe shows the highest accuracy due to its ability to calculate semantic relationship between the words. In order to capitalize on their capacity to identify false news, this study will further explore the integration of transformer-based models, such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) in the future.

References

[1] Kesarwani, Ankit, Sudakar Singh Chauhan, and Anil Ramachandran Nair. "Fake news detection on social media using k-nearest neighbor classifier." international conference on advances in computing and communication engineering (ICACCE), pp. 1-4. IEEE, 2020.

[2] Kishwar, Azka, and Adeel Zafar. "Fake news detection on Pakistani news using machine learning and deep learning." Expert Systems with Applications, Vol.211, pp. 118558, 2023

[3] Choudhary, Anshika, and Anuja Arora. "Linguistic feature based learning model for fake news detection and classification." Expert Systems with Applications, Vol. 169, pp. 114171, 2021.

[4] Mishra, Shubha, Piyush Shukla, and Ratish Agarwal. "Analyzing machine learning enabled fake news detection techniques for diversified datasets." Wireless Communications and Mobile Computing, pp. 1-18, 2022.

[5] Albahr, Abdulaziz, and Marwan Albahar. "An empirical comparison of fake news detection using different machine learning algorithms." International Journal of Advanced Computer Science and Applications, Vol. 11, pp. 9, 2020.

[6] Poddar, Karishnu, and K. S. Umadevi. "Comparison of various machine learning models for accurate detection of fake news." In 2019 Innovations in Power and Advanced Computing Technologies (i-PACT), vol. 1, pp. 1-5. IEEE, 2019.

[7] Krishna, N. Leela Siva Rama, and M. Adimoolam. "Fake News Detection system using Decision Tree algorithm and compare textual property with Support Vector Machine algorithm." In 2022 International Conference on Business Analytics for Technology and Security (ICBATS), pp. 1-6. IEEE, 2022.

[8] Ajao, Oluwaseun, Deepayan Bhowmik, and Shahrzad Zargari. "Sentiment aware fake news detection on online social networks." In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2507-2511. IEEE, 2019.

[9] Mandical, Rahul R., N. Mamatha, N. Shivakumar, R. Monica, and A. N. Krishna. "Identification of fake news using machine learning." In 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), pp. 1-6. IEEE, 2020.

[10] Altheneyan, Alaa, and Aseel Alhadlaq. "Big data ML-based fake news detection using distributed learning." IEEE Access, Vol. 11, pp. 29447-29463, 2023.

[11] Capuano, Nicola, Giuseppe Fenza, Vincenzo Loia, and Francesco David Nota. "Content Based Fake News Detection with machine and deep learning: a systematic review." Neurocomputing, 2023.

[12] Mosallanezhad, Ahmadreza, Mansooreh Karami, Kai Shu, Michelle V. Mancenido, and Huan Liu. "Domain adaptive fake news detection via reinforcement learning." In Proceedings of the ACM Web Conference, pp. 3632-3640, 2022.

[13] Nasir, Jamal Abdul, Osama Subhani Khan, and Iraklis Varlamis. "Fake news detection: A hybrid CNN-RNN based deep learning approach." International Journal of Information Management Data Insights 1, no. 1 (2021): 100007.

[14] Jehad, Reham, and Suhad A. Yousif. "Fake news classification using random forest and decision tree (j48)." Al-Nahrain Journal of Science, Vol. 4, pp. 49-55, 2020.

[15] Bahad, Pritika, Preeti Saxena, and Raj Kamal. "Fake news detection using bi-directional LSTM-recurrent neural network." Procedia Computer Science, Vol.165, pp. 74-82, 2019.

[16] Kaliyar, Rohit Kumar, Anurag Goswami, Pratik Narang, and Soumendu Sinha. "FNDNet–a deep convolutional neural network for fake news detection." Cognitive Systems Research, Vol. 61, pp. 32-44, 2020.

[17] Luvembe, Alex Munyole, Weimin Li, Shaohua Li, Fangfang Liu, and Guiqiong Xu. "Dual emotion based fake news detection: A deep attention-weight update approach." Information Processing & Management 60, Vol.4, pp. 103354, 2023.

[18] Iwendi, Celestine, Senthilkumar Mohan, Ebuka Ibeke, Ali Ahmadian, and Tiziana Ciano. "Covid-19 fake news sentiment analysis." Computers and electrical engineering, Vol. 101, pp. 107967, 2022.

[19] Okunoye, Olusoji B., and Ayei E. Ibor. "Hybrid fake news detection technique with genetic search and deep learning." Computers and Electrical Engineering, Vol. 103, pp. 108344, 2022.

[20] Ouassil, Mohamed-Amine, Bouchaib Cherradi, Soufiane Hamida, Mouaad Errami, Oussama EL Gannour, and Abdelhadi Raihani. "A Fake News Detection System based on Combination of Word Embedded Techniques and Hybrid Deep Learning Model." International Journal of Advanced Computer Science and Applications 13, Vol. 10, 2022.

[21] https://www.kaggle.com/competitions/fake-news/overview

[22] https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

[23] https://www.analyticsvidhya.com/blog/2021/03/introduction-to-gated-recurrent-unit-gru/

[24] https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/

[25] Josh Patterson, Adam Gibson, Deep Learning: A Practitioner's Approach, USA, O'Reilly, 2019.

[26] Michael Walker, Introduction to Natural Language Processing, New York, AI Sciences LLC, 2018.

[27] C. D. Manning, P. Raghavan, and H. Schutze, Introduction to Information Retrieval, Cambridge University Press, 2008.