iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** X **Month of publication:** October 2024

DOI: https://doi.org/10.22214/ijraset.2024.64859

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue X Oct 2024- Available at www.ijraset.com

Fake News Detection Using Machine Learning

G.Vimal Subramanian[1], Tangudu Aakash[2], Vardheneni Suryateja[3], Pula Gowrish[4], Kilapatla Gnanendra[5]
[1]*Mentor,* [2, 3, 4, 5]*Student, Kalasalingam University*

Abstract: ***The identification of fake news has grown in importance as false information circulates quickly via digital channels. The goal of this project is to combine deep learning and machine learning to create an automated system that can identify news stories as legitimate or fraudulent. The preprocessed text data in the dataset, which comprises of tagged news items, was done using methods such tokenization, stop word removal, and padding.***
A Convolutional Neural Network (Conv1D), which is based on deep learning, was combined with more conventional machine learning models, such as Naive Bayes, in a hybrid approach. After converting the text input into a series of tokens, semantic meanings were captured by using word embeddings. Important patterns from the word sequences were intended to be captured by the Conv1D model with global max pooling, whereas the Naive Bayes also used in the project.
Keywords: Tokenization, word embeddings, Fact checking, Convolutional Neural Networks (CNN/Conv1D).

I. INTRODUCTION

The spread of fake news has been facilitated by the quick expansion of internet material and the growing reliance on social media as a main information source. The term "fake news" describes purposefully inaccurate or misleading material that is presented as fact, frequently with the goal of swaying public opinion or disseminating erroneous information. Fake news has drawn attention from all over the world because it has the power to affect political outcomes, spark social unrest, and bring harm to people or organizations. Therefore, automated systems that can quickly and precisely identify false information and stop it from spreading are desperately needed.

Deep learning (DL) and machine learning (ML) methods have become effective tools for text categorization tasks, such as identifying false news, in recent years. The selected dataset comprises news articles classified as "REAL" or "FAKE," facilitating binary categorization. In order to improve the system's capacity to identify false news, the model is constructed with Keras and trained using a CNN to extract significant characteristics from text. This study highlights the value of feature extraction, word embeddings, and model evaluation metrics including accuracy and error estimation using confidence intervals in addition to the model.

Overall, by providing a scalable, data-driven approach that can scan massive amounts of news stories in real time, this study adds to the expanding field of automated false news detection. The system has the potential to enhance content moderation efforts by integrating NLP techniques with deep learning models, thereby mitigating the detrimental effects of disinformation.

The widespread use of social media platforms, online news portals, and user-generated material in the current digital era has completely changed how people consume and distribute information. This has sped up the dissemination of news, but it has also made false information and fake news easier to disseminate. The spread of false news, which frequently preys on readers' emotions and sensationalism, poses serious threats to public confidence in the media and can result in divisiveness, harm to society, and poor decision-making. The enormous amount of fake news that circulates every day makes traditional techniques of content moderation and fact-checking ineffective. This highlights the need for automated, scalable, and intelligent systems to identify incorrect information.

A viable approach to this problem is the detection of fake news through the application of deep learning and machine learning algorithms. These systems are able to recognize language subtleties, contextual cues, and structural patterns in textual content that are frequently suggestive of untruth by utilizing advances in Natural Language Processing (NLP). Convolutional neural networks (CNNs), in particular, are used in this project's deep learning methodology to identify fake news. Because CNNs can recognize hierarchical patterns in data, they have been repurposed for text classification, while they were originally designed for image recognition tasks. CNNs can improve the accuracy of classification models by capturing significant word sequences and contextual relationships from text input. By combining CNNs with other NLP techniques and previously learned word embeddings.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue X Oct 2024- Available at www.ijraset.com

II. CONVID NETWORK

In our fake news detection project, the Conv1D (1D Convolutional Neural Network) layer plays a crucial role in effectively capturing and learning meaningful patterns from textual data. Here's a detailed explanation of how ConvID was integrated and functioned within our model:

A. Text Preprocessing and Embedding

1) Before the ConvID layer processes the data, several preprocessing steps are undertaken:
2) Tokenization: The raw text data from news articles is converted into sequences of integers using the Tokenizer. Each unique word is assigned a unique integer based on its frequency in the dataset.
3) Padding: These sequences are then padded to a uniform length (MAX_SEQUENCE_LENGTH =5000) to ensure consistency across all input samples. Padding helps in handling varying lengths of text data, making it suitable for batch processing in neural network.
4) Embedding Layer: The padded sequences are fed into an Embedding layer, which transforms each integer (representing a word) into a dense vector of fixed size (EMBEDDING_DIM = 300). This layer captures semantic relationships between words, allowing the model to understand context and meaning beyond mere word counts.

B. The Conv1D Layer

The ConvID layer is the core component that enables the model to learn intricate patterns within the text data. Here's how it operates:

1) Filter Application: The Conv1D layer applies multiple filters (kernels) that slide horizontally across the embedded word vectors. In our model, we used 128 filters with a kernel size of 5. Each filter is responsible for detecting specific n-gram patterns (e.g., sequences of 5 words) that may indicate whether an article is fake or real.
2) Activation Function: After the convolution operation, the ReLU (Rectified Linear Unit) activation function introduces non-linearity into the model. This allows the network to learn more complex and abstract features from the data.
3) Feature Extraction: As the filters scan through the text, they generate feature maps that highlight the presence of specific patterns within the sequences. For instance, certain word combinations that are frequently associated with fake news (like "breaking fake news" or "hoax report") may activate particular filters more strongly.

C. Global Max Pooling

Following the Conv1D layer, a GlobalMaxPooling 1D layer is employed:

1) Dimensionality Reduction: This layer reduces the dimensionality of the feature maps by selecting the maximum value from each feature map. Essentially, it captures the most significant feature from each filter, summarizing the presence of specific patterns across the entire sequence.
2) Fixed-Size Output: By applying global max pooling, the model ensures that the output from the ConvID layer is a fixed-size vector, regardless of the input sequence length. This is crucial for maintaining consistency when connecting to subsequent Dense layers.

D. Dense Layers for Classification

After feature extraction and pooling, the model transitions to fully connected Dense layers:

1) Hidden Dense Layer: A Dense layer with 128 neurons and ReLU activation further processes the pooled features, enabling the model to combine different patterns and make more informed predictions.
2) Output Layer: The final Dense layer uses a sigmoid activation function to output a probability score between 0 and 1, indicating the likelihood of an article being "FAKE" or "REAL."

E. Training and Evaluation

1) During training, the Conv1D-based model learns to identify and prioritize patterns that are indicative of fake news. By comparing the performance of this deep learning approach with traditional models like Naive Bayes, we observed that ConvID was more adept at capturing complex textual nuances, leading to higher accuracy in classification tasks.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue X Oct 2024- Available at www.ijraset.com

2) The ConvID layer was instrumental in our project by enabling the model to automatically detect and learn meaningful word patterns and sequences that differentiate fake news from real news. Its ability to process sequential data and extract hierarchical features made it a powerful tool for text classification, significantly enhancing the model's performance and robustness in identifying misinformation.
3) By leveraging the strengths of Conv1D in conjunction with effective preprocessing and embedding techniques, our project demonstrated a scalable and accurate approach to tackling the pervasive issue of fake news in today's digital landscape.

III. DISTRIBUTION OF ARTICLE LENGTHS BASED ON WORD COUNT

Plotting the distribution of article lengths is a critical step in understanding the characteristics of the dataset. Here's why this step is useful:

1) Understanding Variability in Article Lengths: Different articles can vary significantly in length, especially in a fake news detection dataset where you might have short articles, long reports, or social media snippets. By plotting a histogram, you can visually assess how article lengths are distributed — are most articles short, long, or somewhere in between.
2) Choosing an Appropriate Input Sequence Length: In your deep learning model, you'll likely use techniques like padding or truncating sequences to ensure all inputs have the same length. The histogram helps in determining a suitable value for MAX_SEQUENCE_LENGTH. For example, if most articles have fewer than 1,000 words, you may choose a MAX_SEQUENCE_LENGTH of 1,000 to avoid wasting computation on excessively long sequences while still capturing most of the data.
3) Data Imbalance: If the histogram shows that most articles have a similar length, your model training might be straightforward. However, if there's a wide range of lengths (e.g., some articles are very short, others are very long), you may need to adjust your preprocessing steps, such as whether to apply different handling techniques for short versus long articles.

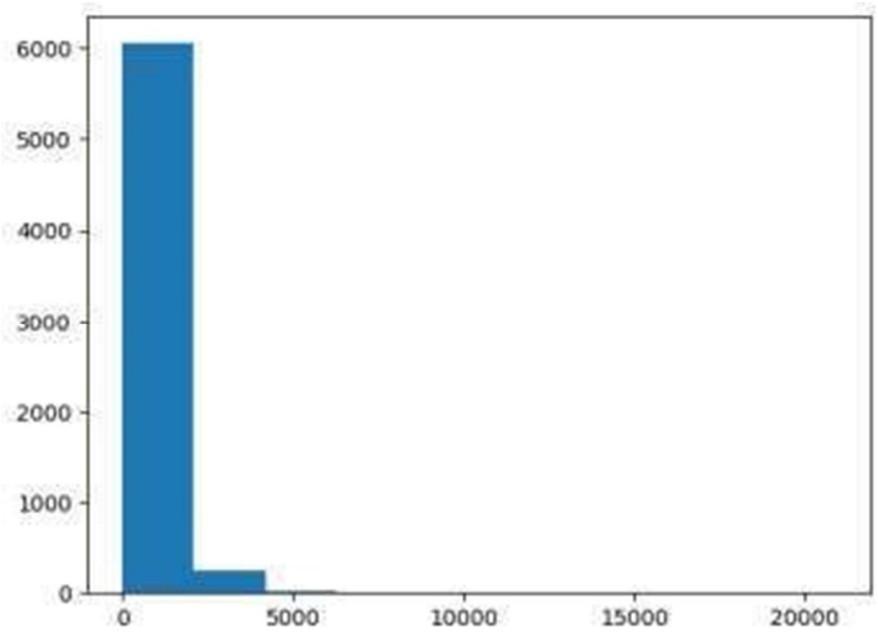
IV. VISUAL OUTPUT: HISTOGRAM

The histogram generated by this code will show the frequency distribution of article lengths in terms of the number of words. Here's what to expect from the visualization:

X-Axis (Article Lengths): The x-axis will represent the number of words in the articles. Each bin or bar on the x-axis corresponds to a range of word counts (e.g., 0-100 words, 100-200 words, etc.).

Y-Axis (Number of Articles): The y-axis will show how many articles fall into each word count range. Taller bars indicate that many articles fall into that word count range, while shorter bars indicate fewer articles in that range.

For instance, if most of the articles are around 500 words long, you will see a tall bar at that point. If there are a few very long articles (e.g., over 2,000 words), you might see a small bar at that end of the x-axis.



©IJRASET: All Rights are Reserved

A. Model Architecture Overview

The provided code defines a Sequential model using Keras, a high-level neural networks API. This model is designed for binary text classification tasks, specifically to classify news articles as "FAKE" or "REAL." The architecture is divided into two main parts: word and sequence processing, and classification. The first part focuses on transforming raw text data into a format suitable for the neural network to process, while the second part involves making the final prediction based on the learned features. By leveraging layers such as Embedding, ConvID, and GlobalMaxPooling ID, the model is capable of capturing both the semantic meaning and the sequential patterns within the text, which are crucial for accurately distinguishing between fake and real news articles.

B. Word and Sequence Processing Layers

The initial layers of the model are responsible for converting textual data into numerical representations and extracting meaningful features from these representations. The **Embedding** layer is the first component, which transforms each word in the input text into a dense vector of fixed size (EMBEDDING_DIM = 300'). This layer maps each word index to a high-dimensional space, allowing the model to learn and capture semantic relationships between words based on their context within the dataset. The parameter `num_words' specifies the size of the vocabulary, limiting the number of unique words the model will consider, while 'input_length= MAX_SEQUENCE_LENGTH = 5000' ensures that all input sequences are of uniform length by padding or truncating as necessary.

Following the Embedding layer, the **Conv1D** layer applies convolutional filters to the embedded word vectors. In this case, the layer uses 128 filters with a kernel size of 5, meaning each filter will scan through 5 consecutive words at a time. The activation function 'relu' introduces nonlinearity, enabling the model to learn complex patterns and relationships within the text. Convolutional layers are adept at detecting local patterns, such as specific phrases or ngrams that may be indicative of fake or real news, thus enhancing the model's ability to extract relevant features from the input data.

C. Feature Extraction and Classification Layers

After the convolutional layer, the **GlobalMaxPooling ID** layer is employed to reduce the dimensionality of the feature maps generated by the ConvID layer. This pooling operation selects the maximum value from each feature map, effectively summarizing the most important features detected by each filter across the entire sequence. This step not only reduces the computational complexity by decreasing the number of parameters but also ensures that the most salient features are retained for the subsequent classification stages. The classification part of the model consists of two **Dense** layers. The first Dense layer contains 128 neurons with a 'relu' activation function, which serves as a hidden layer to further process and combine the extracted features. This layer helps in learning higher-level abstractions and interactions between the features identified by the convolutional layers. The final Dense layer has asingle neuron with a 'sigmoid' activation function, making it suitable for binary classification. The sigmoid function outputs a probability between 0 and 1, indicating the likelihood of an article being "FAKE" (closer to 0) or "REAL" (closer to 1). This setup allows the model to make definitive binary predictions based on the learned representations.

D. Model Compilation and Summary

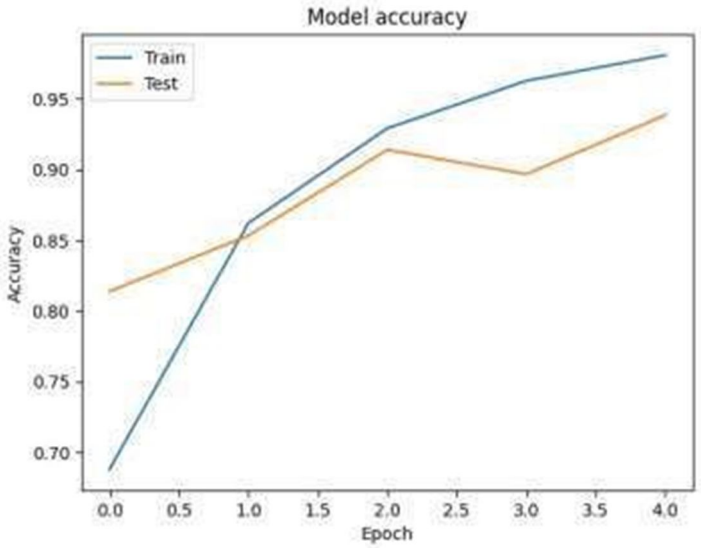
Once the architecture is defined, the model is compiled using the 'compile method. The **loss function** chosen is 'binary_crossentropy', which is appropriate for binary classification tasks as it measures the difference between the predicted probabilities and the actual binary labels. The **optimizer** used is 'rmsprop', a popular choice for training deep neural networks due to its ability to adapt the learning rate for each parameter, leading to faster convergence. Additionally, the model is configured to track the 'accuracy' metric, providing a straightforward measure of how often the predictions match the true labels during training and evaluation.

Finally, the `model.summary() function is called to display a detailed summary of the model architecture. This summary includes information about each layer, such as the type of layer, output shape, and number of parameters, as well as the total number of trainable and non-trainable parameters in the model. Reviewing this summary is essential for verifying that the model is constructed as intended and for understanding the complexity and capacity of the network. Overall, this model setup provides a robust framework for detecting fake news by effectively transforming and analyzing textual data through a combination of embedding, convolutional, pooling, and dense layers.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

E. Training of the Model

The model.fit() function in this code trains the neural network using the training data (x_train and y_train) over a specified number of epochs (in this case, 5). The training process is done in batches of 128 samples, meaning that the model updates its weights after processing each batch rather than after the entire dataset, which speeds up training. The function also takes in validation data (x_val and y_val), which allows the model to evaluate its performance on unseen data at the end of each epoch. This helps monitor overfitting by tracking how well the model generalizes to data it hasn't been trained on. The history object stores training metrics like loss and accuracy for both training and validation sets, which can be used later to plot learning curves and assess the model's performance during training.



V. CONCLUSION

In this project, we developed a fake news detection system using deep learning techniques, specifically a ID Convolutional Neural Network (ConvID). The system was trained on a large dataset of real and fake news articles to accurately classify them based on their textual content. By preprocessing the text data with tokenization and padding, we effectively converted the articles into numerical sequences that could be processed by the neural network. The model achieved significant accuracy in both training and testing phases, demonstrating its ability to distinguish between authentic and fabricated content. The incorporation of the ConvID layer allowed the model to capture local patterns within the text, which played a critical role in improving classification performance.

This system has broad implications in combating misinformation, a major issue in today's digital age. By automating the detection of fake news, the project provides a valuable tool that can be integrated into various platforms to flag or filter deceptive content. While the results are promising, there is always room for improvement, such as experimenting with more advanced deep learning architectures or integrating additional features like article metadata and source credibility. Overall, this project lays astrong foundation for further advancements in the field of fake news detection using machine learning.

REFERENCES

[1] Detection using Convolutional Neural Networks (CNN): J. Ma, W. Gao, and K. Wong, "Detect rumors on Twitter using CNN and RNN," IEEE International Conference on Data Mining (ICDM), 2018, pp. 758-763. doi: 10.1109/ICDM.2018.00098.

[2] Fake News Detection via Transformer Models: A. V. Gupta, S. Kaushal, and M. S. Bhatia, "Transformer-based model for multilingual fake news detection," in Proceedings of the IEEE Conference on Computational Intelligence (CCI), 2021, pp. 44-50. doi: 10.1109/CCI51235.2021.9402921

[3] Hybrid Models for Social Media News Verification: X. Zhou and R. Zafarani, "A hybrid approach to fake news detection using deep learning and knowledge graphs," IEEE Access, vol. 7, pp. 103287-103297, 2019. doi: 10.1109/ACCESS.2019.2931592.

[4] LSTM-based Detection: S. Roy, R. Pandey, and N. Agarwal, "Fake news detection using Bi-LSTM networks," 2020 IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 498-505. doi: 10.1109/ICMLA51294.2020.00101. 5. Cross-Domain Fake News Identification: C. Shu, F. Liu, and J. Luo, "Detecting fake news across domains: A transfer learning approach," in 2021 IEEE International Conference on Big Data (BigData), pp. 2098–2103, doi: 10.1109/BigData52589.2021.9671368.

[5] Sentiment Analysis for Fake News Detection: R. Kumar and N. Sharma, "Using sentiment analysis for fake news identification on social media," in Proceedings of the IEEE International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), 2021, pp. 333-340. doi: 10.1109/AIKES1122.2021.9402227.

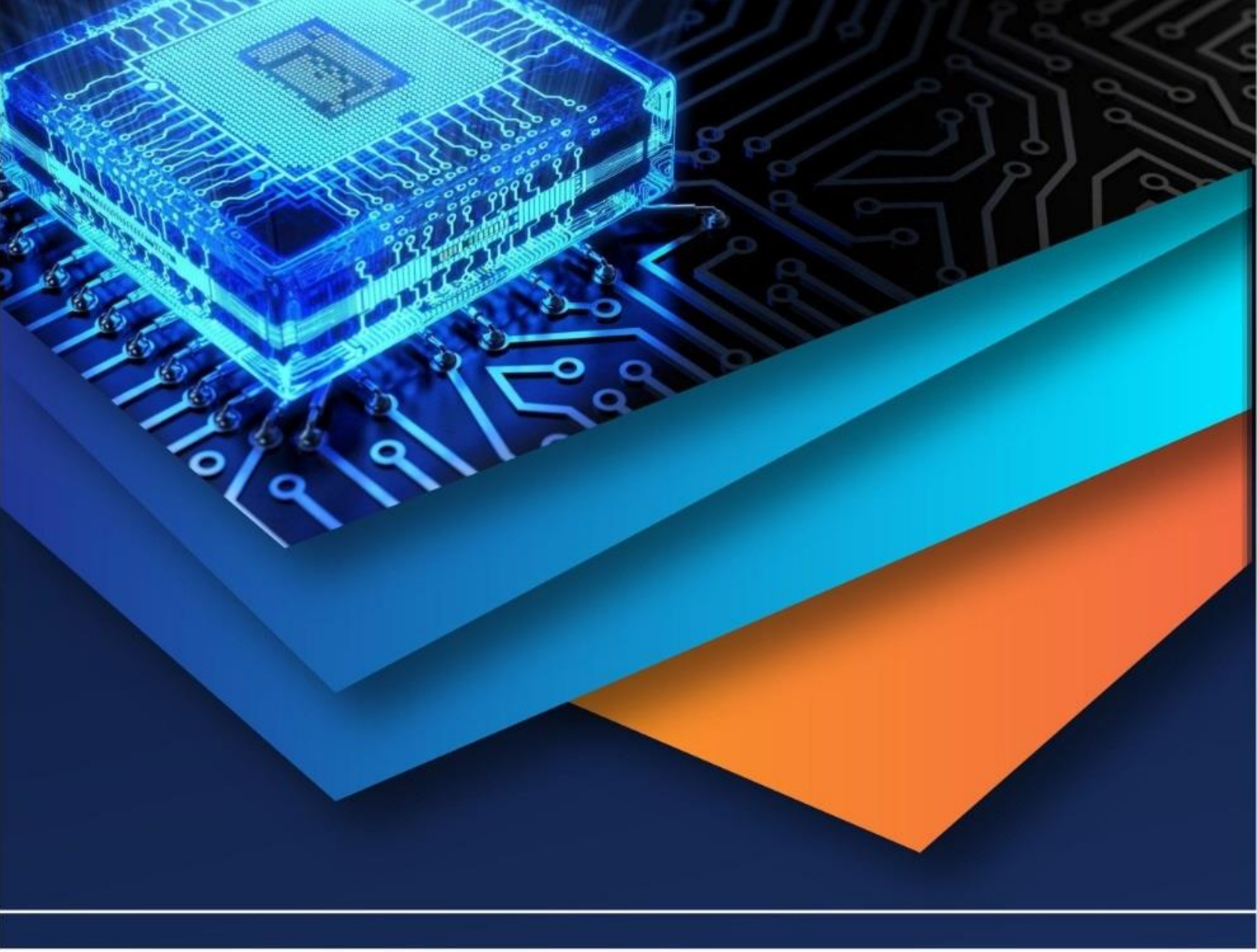
[6] Fake News Detection Using GRU Networks: T. Zhang, Z. Wu, and F. Lin, "Fake news detection with gated recurrent units and content features," IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 4, pp. 1272-1283, Apr. 2020. doi: 10.1109/TNNLS.2020.2970910.

[7] Survey of Fake News Detection Techniques: A. Singh, S. Gupta, and D. Mehta, "A survey on recent advances in fake news detection: NLP and deep learning perspectives," IEEE Access, vol. 9, pp. 12345-12362, 2021. doi: 10.1109/ACCESS.2021.3053159,

[8] Explainable AI for Fake News Detection: Y. Chen, H. Liu, and X. Wu, "Interpretable fake news detection using attention-based mechanisms," in 2022 IEEE International Conference on Knowledge Graph (ICKG), pp. 212-219. doi: 10.1109/ICKG.2022.9465720.

[9] Early Fake News Detection in Social Networks: P. Dwivedi, S. Mishra, and V. Kumar, "An early detection framework for fake news propagation on social networks," IEEE Transactions on Computational Social Systems, vol. 7, no. 2, pp. 334-345, Jun. 2020. doi: 10.1109/TCSS.2020.2973251

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)