Detection of Machine-Generated Tweets through a Hybrid CNN-LSTM Deep Learning Framework

[1]V Aparana, [1]Muttum Usha Varshini, [2]Pothagani Navya, [3]Potru Charitha, [4]Nagineni Hansika, [5]Pashikanti Niharika

[1]Assistant Professor, [2,3,4,5,6]UG Student, [1,2,3,4,5,6]Department of CSE (Data Science) Malla Reddy Engineering Collge for Women, Secunderabad, India

E-Mail: ushavarshini7@gmail.com

Abstract

The recent breakthrough in the natural language production can be considered another weapon to control the masses opinion in the social media. Moreover, language modelling has substantially enhanced the generative capabilities of deep neural models, giving them more content generation skills. Therefore, text-generative models have developed into more powerful ones that provide the opponents an opportunity to employ these incredible skills to empower social bots, enabling them to produce authentic deep fake posts and manipulate the conversation of the masses. In order to solve this issue, it is significant to develop effective and precise deepfake social media message detecting algorithms. In this light, the modern day studies entail the discovery of the machine generated text in social networks such as Twitter. This project uses a simple deep learning model along with word embeddings to classify the tweets as either human generated or bot-generated tweets based on a publicly available dataset made up of Tweep fake. A standard Convolutional Neural Network (CNN) based on a Long ShortTerm Memory(LSTM) network is developed, to execute the role of detecting deepfake tweets. In order to demonstrate the high quality of the offered method, this project used a number of machine learning models as the baseline methods to compare them. Additionally, the effectiveness of the suggested approach is also contrasted with other deep learning networks like Long short-term memory (LSTM) that demonstrates the efficiency and outlines the benefits of the approach in coping with the given task correctly.

Keywords: CNN, LSTM, tweets, message detecting, deep fake

Introduction

Social media platforms were created for people to via (text, image, audio and video) that may be misleading connect and share their opinions and ideas through texts, Deepfake multimedia's creation and sharing on social media images, audio, and videos. A bot is computer software that manages a fake account on social media by liking, sharing, and uploading posts that may be real or forged using techniques like gap-filling text, search-and-replace, and video editing or deepfake. Deep learning is a part of machine learning that learns feature representation from input data. Deepfake is a combination of "deep learning" and have already created problems in a number of fields such as politics by deceiving viewers into thinking that they were created by humans. Using social media, it is easier and faster to propagate false information with the aim of manipulating people's perceptions and opinions especially to build mistrust in a democratic country. Accounts with varying degrees of humanness like cyborg accounts to sock puppets are used to achieve this goal. On the other hand, fully automated social media accounts also known as social bots mimic human behaviour. Particularly, the widespread use of bots and recent developments in natural language-based generative models, such as the GPT and Grover, give the adversary a means to propagate false information more convincingly. The Net Neutrality case in 2017 serves as an illustrative example: millions of duplicated comments played a significant role in the Commission's decision to repeal. The issue needs to be addressed that simple text manipulation techniques may build false beliefs and what could be the impact of more powerful transformer-based models. Recently, there

have been instances of the use of GPT-2 and GPT-3 : to generate tweets to test the generating skills and automatically make blog articles. A bot based on GPT3 interacted with people on Reddit using the account "/u/thegentlemetre" to post comments to inquiries on /r/AskReddit. Though most of the remarks made by the bot were harmless. Despite the fact that no harm has been done thus far, OpenAI should be concerned about the misuse of GPT-3 due to this occurrence. However, in order to protect genuine information and democracy on social media, it is important to create a sovereign detection system for machine-generated texts, also known as deepfake text. In 2019, a generative model namely GPT-2 displayed enhanced text-generating capabilities which remained unrecognizable by the humans. Deepfake text on social media is mainly written by the GPT model; this may be due to the fact that the GPT model is better than Grover and CTRL at writing short text.

Motivation

Identifying machine-generated tweets is crucial in various fields, including journalism, cybersecurity, and content moderation, to ensure the authenticity and reliability of information. Traditional methods often struggle to distinguish between human and machine-generated content due to increasingly sophisticated AI technologies.

The CNN-LSTM architecture offers a powerful solution by combining the strengths of Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs). CNNs excel at capturing local patterns and features in data, making them effective for text classification tasks. Meanwhile, LSTMs are well-suited for capturing sequential dependencies, crucial for understanding the context and coherence of text.

By leveraging the CNN-LSTM architecture, this project aims to enhance the accuracy and efficiency of machine-generated text detection. The CNN component efficiently extracts relevant features from text, while the LSTM component captures the temporal dynamics, allowing for robust detection of subtle patterns indicative of machine-generated content.

Ultimately, this project seeks to contribute to the development of reliable tools for detecting machine-generated texts, thereby safeguarding against misinformation and ensuring the integrity of communication channels in an increasingly AI-driven world.

Objective

CNN-LSTM architecture to accurately identify machine-generated tweets amidst the proliferation of misinformation on social media. Leveraging publicly available datasets, the model aims to distinguish between human-generated and bot-generated tweets with high accuracy. Through rigorous experimentation, the project seeks to showcase the effectiveness of the proposed approach in combating the spread of machine-generated misinformation, thereby safeguarding the integrity of online discourse.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. arXiv preprint arXiv:2301.07597, 2023.

The introduction of ChatGPT has garnered widespread attention in both academic and industrial communities. ChatGPT is able to respond effectively to a wide range of human questions, providing fluent and comprehensive answers that significantly surpass previous public chatbots in terms of security and usefulness. On one hand, people are curious about how ChatGPT is able to achieve such strength and how far it is from human experts. On the other hand, people are starting to worry about the potential negative impacts that large language models (LLMs) like ChatGPT could have on society, such as fake news, plagiarism, and social security issues. In this work, we collected tens of thousands of comparison responses from both human experts and ChatGPT, with questions ranging from open-domain, financial, medical, legal, and psychological areas. We call the collected dataset the Human ChatGPT Comparison Corpus (HC3). Based on the HC3 dataset, we project the characteristics of ChatGPT's responses, the differences and gaps from human experts, and future directions for

LLMs. We conducted comprehensive human evaluations and linguistic analyses of ChatGPT-generated content compared with that of humans, where many interesting results are revealed. After that, we conduct extensive experiments on how to effectively detect whether a certain text is generated by ChatGPT or humans. We build three different detection systems, explore several key factors that influence their effectiveness, and evaluate them in different scenarios

Rexhep Shijaku and Ercan Canhasi. Chatgpt generated text detection.
Generative models, such as ChatGPT, have gained significant attention in recent years for their ability to generate human-like text. However, it is still a challenge to automatically distinguish between text generated by a machine and text written by a human. In this paper, we present a classification model for automatically detecting essays generated by ChatGPT. To train and evaluate our model, we use a dataset consisting of essays written by human writers and ones generated by ChatGPT. Our model is based on XGBoost and we report its performance on two different feature extraction schemas. Our experimental results showthat our model successfully (with 91% accuracy) detects ChatGPT-generated text. Overall, our results demonstrate the feasibility of using machine learning to automatically detect ChatGPT generated text, and provide a valuable rsource for researchers and policymakers interested in understanding and combating the use of ChatGPT for malicious Purposes.

Nick Hajli, Usman Saeed, Mina Tajvidi, and Farid Shirazi. Social bots and the spread of disinformation in social media: the challenges of artificial intelligence. British Journal of Management, 33(3):1238–1253, 2022
Artificial intelligence (AI) is creating a revolution in business and society at large, as well as challenges for organizations. AI-powered social bots can sense, think and act on social media platforms in ways similar to humans. The challenge is that social bots can perform many harmful actions, such as providing wrong information to people, escalating arguments, perpetrating scams and exploiting the stock market. As such, an understanding of different kinds of social bots and their authors' intentions is vital from the management perspective. Drawing from the actornetwork theory (ANT), this project investigates human and non-human actors' roles in social media, particularly Twitter. We use text mining and machine learning techniques, and after applying different pre-processing techniques, we applied the bag of words model to a dataset of 30,000
English-language tweets. The present research is among the few studies to use a theory-based focus to look, through experimental research, at the role of social bots and the spread of disinformation in social media. Firms can use our tool for the early detection of harmful social bots before they can spread misinformation on social media about their organizations.

David Dukic, Dominik Ke ´ ca, and Dominik Stipi ˇ c. Are you human? ´detecting bots on twitter using bert. In 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), pages 631–636. IEEE, 2020.
Dissemination of fake news on Twitter is a rapidly growing problem, mostly due to the increasing number of bots. Hence, automatic bot detection is becoming an important area of research. In this work, we present the BERT-based bot detection model along with exploratory data analysis of tweets written by bots and humans. We statistically prove that including additional features alongside contextualized embeddings boosts model performance. Furthermore, we develop a gender prediction model using derived features and compare the difficulties of the two tasks. Finally, we demonstrate how Logistic Regression outperforms Deep Neural Network on both tasks.

Sneha Kudugunta and Emilio Ferrara. Deep neural networks for bot detection. Information Sciences, 467:312–322, 2018.

The problem of detecting bots, automated social media accounts governed by software but disguising as human users, has strong implications. For example, bots have been used to sway political elections by distorting online discourse, to manipulate the stock market, or to push antivaccine conspiracy theories that may have caused health epidemics.Most techniques proposed to date detect bots at the account level, by processing large amounts of social media posts, and leveraging information from network structure, temporal dynamics, sentiment analysis, etc. In this paper, we propose a deep neural network based on contextual long short-term memory (LSTM) architecture that exploits both content and metadata to detect bots at the tweet level: contextual features are extracted from user metadata and fed as auxiliary input to LSTM deep nets processing the tweet text. Another contribution that we make is proposing a technique based on synthetic minority oversampling to generate a large labeled dataset, suitable for deep nets training, from a minimal amount of labeled data (roughly 3000 examples of sophisticated Twitter bots). We demonstrate that, from just one single tweet, our architecture can achieve high classification accuracy in separating bots from humans.

Proposed System

CNN-LSTM

The CNN-LSTM architecture, a hybrid model combining Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) layers, offers a powerful solution for analyzing sequential data like text. CNNs are adept at extracting local features, while LSTMs excel at capturing longrange dependencies. In this architecture, CNN layers are employed to extract relevant features from the input data, capturing local patterns and features within the text. The outputs from CNN layers are then fed into LSTM layers, which model the sequential nature of the data and learn dependencies over time. This combination enables the model to effectively capture both local and sequential information, making it well-suited for tasks such as sentiment analysis, sequence labeling, and machine translation. By leveraging the complementary strengths of CNNs and LSTMs, the CNN-LSTM architecture provides a robust framework for tasks requiring understanding and analysis of sequential data.

Convolutional Neural Network (CNN):

CNNs are effective for extracting spatial and temporal patterns from input data, commonly used in image processing but also applicable to sequential data.

In the context of text analysis, CNNs can learn local features from the text, such as word combinations and phrases.

Long Short-Term Memory (LSTM):

LSTMs are a type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data.

They consist of memory cells that can maintain information over time, allowing them to remember and process information from previous time steps.

Hybrid Architecture:

The CNN-LSTM architecture combines the strengths of both CNNs and LSTMs for sequential data processing.

CNNs are used for feature extraction, capturing local patterns and features from the input data.

LSTMs are employed for sequence modeling, capturing long-range dependencies and contextual information from the sequential data.

Feature Extraction with CNN:
The CNN component of the architecture processes the input data, extracting relevant features.
Convolutional layers apply filters across the input data, capturing local patterns and features.
Pooling layers reduce the dimensionality of the extracted features, retaining the most relevant information.

Sequence Modeling with LSTM:
The LSTM component of the architecture processes the features extracted by the CNN.
LSTM layers model the sequential nature of the data, capturing dependencies over time.
They maintain a memory state that can store and update information over multiple time steps, facilitating long-term learning.

Combining CNN and LSTM Layers:
The outputs of the CNN layers are fed into the LSTM layers, combining local and sequential information.
This integration allows the model to capture both local features and long-range dependencies, improving its ability to understand and analyze sequential data effectively.

Training and Optimization:
The CNN-LSTM architecture is trained using supervised learning on labeled sequential data.
During training, the model learns to extract relevant features and capture dependencies from the input data.
Optimization techniques such as backpropagation and gradient descent are used to adjust the model parameters and minimize the loss function.

Application Areas:
The CNN-LSTM architecture is widely used in various applications such as natural language processing (NLP), time series analysis, and video processing.
It is particularly effective for tasks involving sequential data, where capturing both local patterns and long-term dependencies is essentiality.

System Architecture

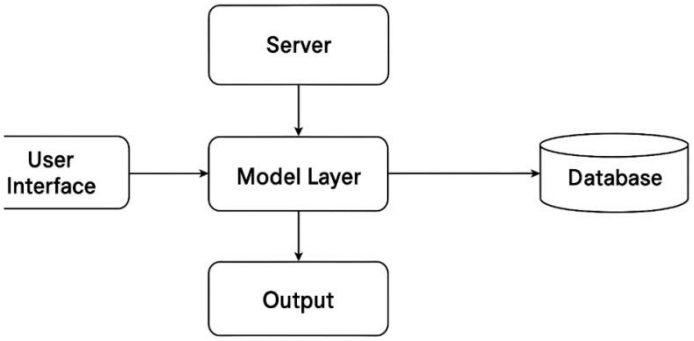

Fig.1. Architecture of proposed framework for deepfake tweet classification

The system architecture methodology for identifying machine-generated texts through a CNNLSTM architecture involves a multi-step process aimed at leveraging the strengths of both convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. Initially, a diverse dataset comprising human-generated and machine-generated texts is collected and preprocessed to remove noise and standardize formats. The preprocessed textual data is then transformed into numerical vectors using word embeddings like Word2Vec or GloVe. Subsequently, CNN layers are employed to extract relevant features from the text, capturing local

patterns and features. These features are then fed into LSTM layers, which model the sequential nature of the text data and learn long-range dependencies. The outputs of the CNN and LSTM layers are combined to create a hybrid architecture, enabling the model to capture both local and sequential information effectively. The model is then trained on a labeled dataset, validated for generalization, and evaluated on a separate test dataset to assess its performance in accurately identifying machine-generated texts. Upon satisfactory performance, the model is deployed into production, integrated with existing systems, and continuously monitored for improvements to adapt to new types of machine-generated texts and evolving techniques used by malicious actors. Through this architecture, the system can effectively distinguish between human-generated and machine-generated texts, contributing to the mitigation of misinformation and fraudulent content spread.

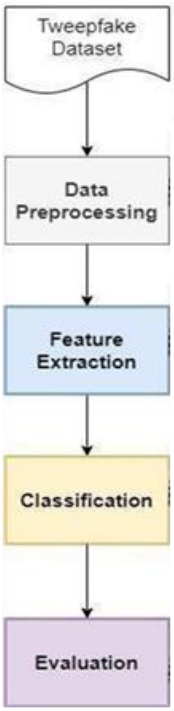

Fig.2. Architecture of methodologies adopted for deepfake tweet classification

1.Data Collection and Prep rocessing:

Data collection involves gathering a diverse dataset containing both human-generated and machine-generated texts from various sources such as social media platforms or online forums. Preprocessing steps include cleaning the text data by removing noise, such as special characters, punctuation, and HTML tags, and standardizing the text format.

2.Word Embeddings:

Word embeddings transform the preprocessed textual data into numerical vectors, capturing the semantic relationships between words. Methods like Word2Vec or GloVe are commonly used to generate dense representations of words in a continuous vector space, preserving contextual information.

3.CNN Feature Extraction:

Convolutional Neural Networks (CNNs) are employed to extract relevant features from the text data. CNN layers apply convolutional filters across the input text, capturing local patterns and features. These filters help in identifying meaningful patterns within the text, which are crucial for subsequent classification tasks.

4.LSTM Sequence Modeling:
Long Short-Term Memory (LSTM) networks are utilized to model the sequential nature of the text data. LSTMs are capable of learning long-range dependencies and retaining information over extended periods, making them suitable for analysing sequences of data such as sentences or paragraphs.

5.Combination of CNN and LSTM Layers:
The outputs of the CNN layers are combined with the LSTM layers to create a hybrid architecture.
This combination leverages both the local feature extraction capabilities.

6.Training and Validation:
The combined CNN-LSTM model is trained on a labeled dataset containing examples of both human-generated and machine-generated texts. During training, the model learns to distinguish between the two classes of texts. Validation is performed on a separate dataset to assess the model's performance and tune hyperparameters to improve generalization.

7.Evaluation and Testing:
The trained model is evaluated on a separate test dataset to measure its ability to accurately identify machine-generated texts. Evaluation metrics such as accuracy, precision, recall, and F1-score are used to quantify the model's performance and assess its effectiveness in distinguishing between human-generated and machine-generated texts.

8.Deployment and Integration:
Once the model demonstrates satisfactory performance, it is deployed into production for realtime identification of machine-generated texts. Integration with existing systems may be necessary to enable seamless operation within the desired environment.

9.Continuous Monitoring and Improvement:
The deployed model is continuously monitored to ensure its effectiveness in identifying machinegenerated texts. Feedback mechanisms and retraining strategies are implemented to adapt the model to new types of machine-generated texts and evolving techniques used by malicious actors, thereby improving its accuracy and robustness over time.

DATASET
This project utilizes TweepFake dataset containing 25835 tweets in total.
(TWEETS) Via the Twitter REST API, the timelines of both deep-fake accounts and their corresponding human profile were downloaded. To get a balanced dataset on both categories (human' and 'bot'), tweets were randomly sampled from each account pair (human and bot/s) based on the less productive. For example, if the bot (human) account had X tweets and the corresponding human (bot) account N tweets (with N>X), X tweets were sampled from the set of N tweets to get the same amount of data. In total, 25,835 tweets were collected (half-human, halfbot).
(DATASET SPLITS) Stratified sampling was applied w.r.t. the screen_name of the account, which means that both train, validation and test sets have got tweets from each account. First, train and test sets were extracted from the whole dataset, assigning 90% of the total tweets to the train set and the remaining 10% to the test set. Then, 10% of the training tweets were pulled out from the train set to compose the validation set.

SPLIT	#BOT TWEETS	#HUMAN TWEETS	TOTAL
TRAINING SET	10354	10358	20712
VALIDATION SET	1152	1150	2302
TEST SET	1280	127	2558

SYSTEM MODULES

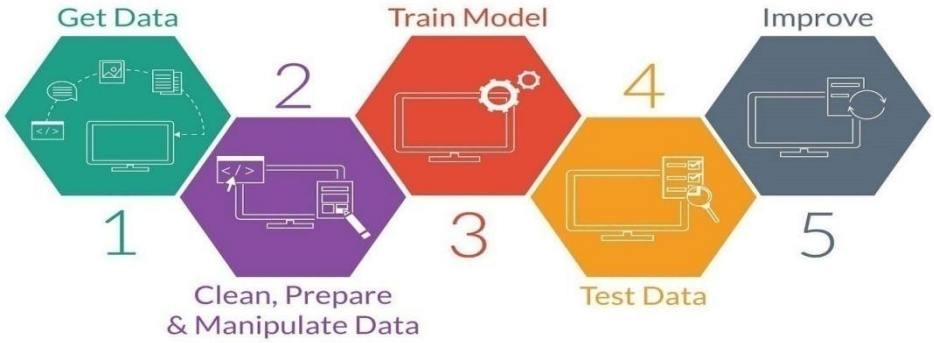

Fig.3. System Modules

Data Preparation:

Collect data: Use Twitter APIs or web scraping techniques to collect tweets from various sources.

Gather both real tweets from legitimate users and fake tweets generated by bots.

Preprocess Data: Data preprocessing involves several steps such as cleaning the data to handle missing values, duplicates, and outliers, transforming categorical variables into numerical representations, and preprocessing text data by tokenizing, lowercasing, and removing stopwords.

Model Training:

Model training is the process of teaching a algorithm to recognize patterns and make predictions based on input data. It involves providing the algorithm with a labeled dataset, consisting of input features and corresponding target labels. During training, the algorithm iteratively updates its parameters using optimization techniques such as gradient descent to minimize a predefined loss function.

CNN-LSTM: This pre-trained model is used to identify patterns in the data and gives the predicted output. In this the perplexity score is generated and if it is greater than 0.34 then it is said to be as human generated tweet and if it is less than 0.34 then is said as bot generated tweet. It classifies and gives the predictions 0(Bot) or 1(Human).

Testing:

Use the trained model to make predictions on the test data. Feed the input features from the test dataset into the model and obtain predicted outputs. For each data point in the test dataset, the model generates predictions based on the learned patterns from the training data.

First, train and test sets were extracted from the whole dataset, assigning 90% of the total tweets to the train set and the remaining 10% to the test set. Then, 10% of the training tweets were pulled out from the train set to compose the validation set.

Evaluation:

Evaluation refers to the process of assessing the performance and effectiveness of a predictive model or algorithm. Evaluation is crucial for determining how well the model generalizes to new, unseen data and whether it meets the desired objectives of the task at hand. Calculate metrics like accuracy, precision, recall, and F1-score to assess the model's performance on unseen data

RESULTS

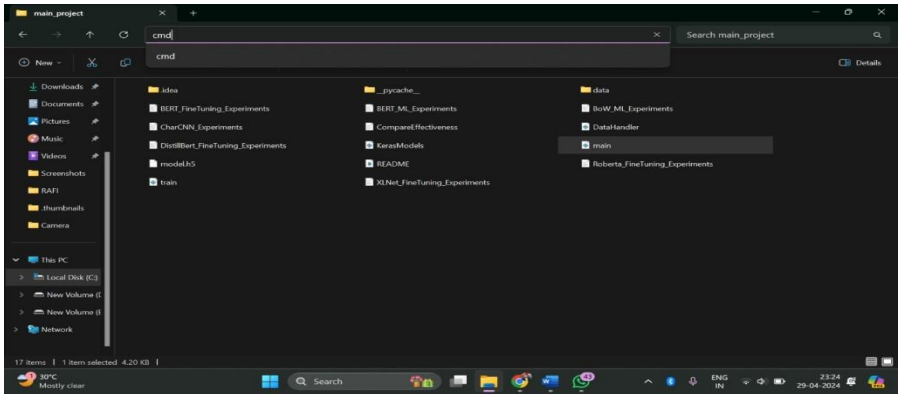

Fig.4. Open the location of your code and redirect to command prompt.

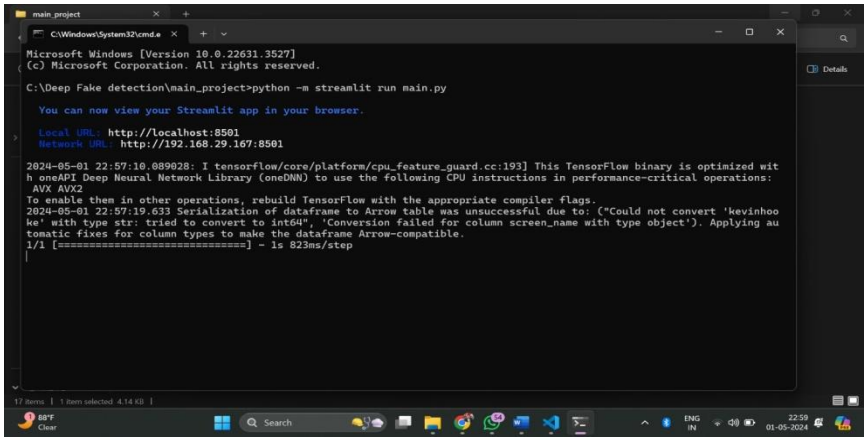

Fig.5. Enter the command to run the code.

Fig.6. It redirects to the browser when code is successfully executed. Appears Abstract page.

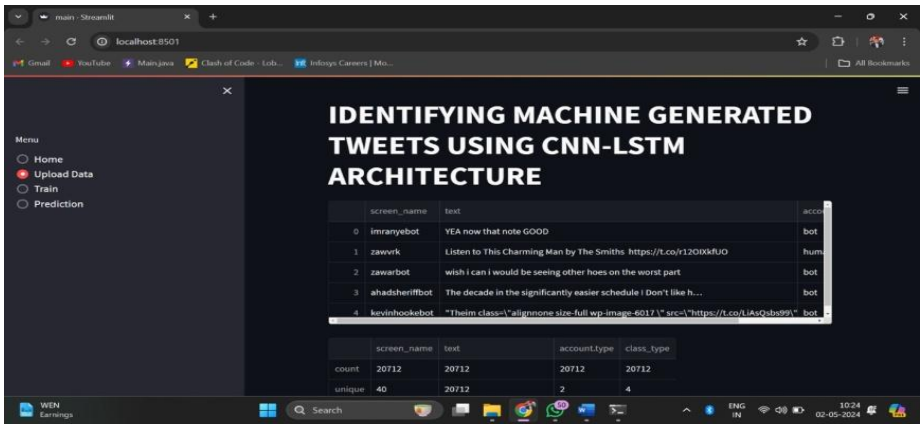

Fig.7. This is an existing uploaded data which was trained in a separate Excel file.

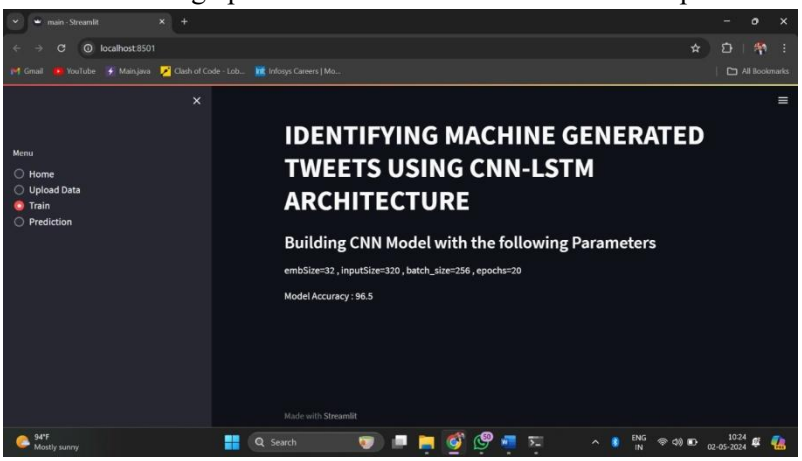

Fig.8. Accuracy of the Model

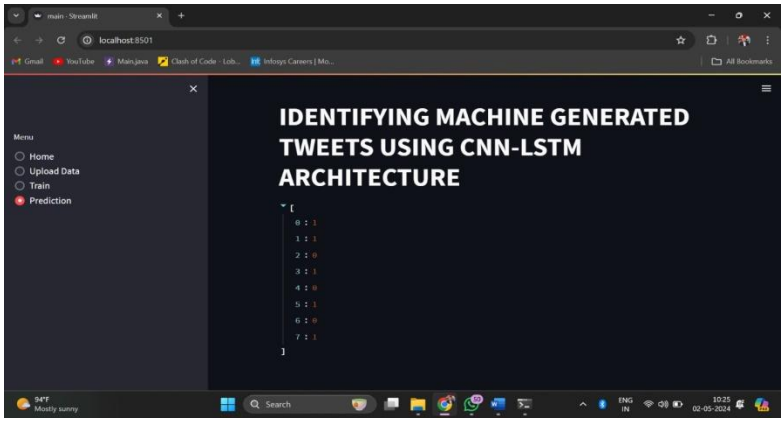

Fig.9. Prediction of a Bot or Human.

CONCLUSION

Deepfake text detection is a critical and challenging task in the era of misinformation and manipulated content. This project aimed to address this challenge by proposing an approach for deepfake text detection and evaluating its effectiveness. A dataset containing tweets of bots and humans is used for analysis by applying several machine learning and deep learning models along with feature engineering techniques. Well-known

feature extraction techniques: TF and TF-IDF and word embedding techniques: Fast Text and Fast Text sub words are used. By leveraging a combination of techniques such as CNN-LSTM, the proposed approach demonstrated promising results with a 0.96 accuracy score in accurately identifying deepfake text. Furthermore, the results of the proposed approach is compared with other state-of-the-art transfer learning models from previous literature. Overall, the adoption of a CNN-LSTM model structure in this project shows its superiority in terms of simplicity, computational efficiency, and handling out-of-vocabulary terms. These advantages make the proposed approach a compelling option for text detection tasks, demonstrating that sophisticated performance can be achieved without the need for complex and time-consuming transfer learning models. The findings of this project contribute to advancing the field of deepfake detection and provide valuable insights for future research and practical applications. As social media continues to play a significant role in shaping public opinion, the development of robust deepfake text detection techniques is imperative to safeguard genuine information and preserve the integrity of democratic processes. In future research, the quantum NLP and other cutting-edge methodologies will be applied for more sophisticated and efficient detection systems, to fight against the spread of misinformation and deceptive content on social media platform.

FUTURE WORK

In the realm of deepfake detection on social media, the future holds promising avenues for advancement and innovation. By combining CNN-LSTM presents a robust foundation for identifying machine-generated tweets, yet there are several areas ripe for further exploration. One such area is the development of multi-modal approaches that integrate various data sources such as text, images, and metadata to enhance detection accuracy and reliability. Additionally, there is a growing need to prioritize research efforts towards enhancing the robustness and generalization capabilities of detection models, ensuring their efficacy across diverse types of deepfakes and resilience against adversarial attacks. Real-time detection systems that can swiftly flag suspicious content as it surfaces on social media platforms represent another crucial frontier, demanding optimization for efficiency and scalability. Moreover, the creation of user-friendly tools and plugins, alongside educational initiatives to empower users in recognizing and reporting deepfake content, will play pivotal roles in combating misinformation. Collaboration with social media platforms to integrate detection mechanisms directly into their systems and the development of legal and policy frameworks to address deepfake proliferation are also essential steps towards fostering a safer digital environment. By prioritizing these avenues of research and development, the field of deepfake detection on social media can continue to evolve, providing effective solutions to mitigate the harmful impacts of synthetic media manipulation

References

[1] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. arXiv preprint arXiv:2301.07597, 2023.

[2] Rexhep Shijaku and Ercan Canhasi. Chatgpt generated text detection.

[3] Nick Hajli, Usman Saeed, Mina Tajvidi, and Farid Shirazi. Social bots and the spread of disinformation in social media: the challenges of artificial intelligence. British Journal of Management, 33(3):1238–1253, 2022.

[4] David Dukic, Dominik Ke ´ ca, and Dominik Stipi ˇ c. Are you human? ´ detecting bots on twitter using bert. In 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), pages 631– 636. IEEE, 2020.

[5] Sneha Kudugunta and Emilio Ferrara. Deep neural networks for bot detection. Information Sciences, 467:312–322, 2018.

[6] Yongqiang Ma, Jiawei Liu, and Fan Yi. Is this abstract generated by ai? a research for the gap between ai-generated scientific text and humanwritten scientific text. arXiv preprint arXiv:2301.10416, 2023.

[7] Sandra Mitrovic, Davide Andreoletti, and Omran Ayoub. Chatgpt ´ or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text. arXiv preprint arXiv:2301.13852, 2023.

[8] Maryam Heidari and James H Jones Jr. Bert model for social media bot detection. 2022.

[9] Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic detection of generated text is easiest when humans are fooled. arXiv preprint arXiv:1911.00650, 2019.

[10] Leo Breiman. Random forests. Machine learning, 45:5–32, 2001.

[11] Hailun Xie, Li Zhang, and Chee Peng Lim. Evolving cnn-lstm models for time series prediction using enhanced grey wolf optimizer. IEEE access, 8:161519–161541, 2020.

[12] Hailun Xie, Li Zhang, and Chee Peng Lim. Evolving cnn-lstm models for time series prediction using enhanced grey wolf optimizer. IEEE access, 8:161519–161541, 2020.

[13] S Selva Birunda and R Kanniga Devi. A review on word embedding techniques for text classification. Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2020, pages 267–281, 2021. Maryam Heidari and James H Jones Jr. Bert model for social media bot detection. 2022.