FAKE NEWS DETECTION

By

1. Bhumika S (20BEC1100)
2. Md Nameer Uddin Haider (20BEC1029)
3. Ajay Krishnan (20BEC1355)
4. Mridul Negi (20BEC1126)

A project report submitted to

SmartInternz

in fulfillment of the requirements for the externship programmed in
Artificial Intelligence

Submitted on:
28 June 2026

1. INTRODUCTION

Overview:

Fake news has become a significant concern in the digital age, spreading misinformation and influencing public opinion. To address this issue, researchers have developed various techniques to detect and combat fake news. One such approach involves using Long Short-Term Memory (LSTM) networks combined with a Flask-based website.

LSTM is a type of recurrent neural network (RNN) that is particularly effective in modeling sequential data, making it suitable for analyzing text. The basic idea behind using LSTMs for fake news detection is to train the model on a large dataset of labeled news articles, distinguishing between real and fake news.

Here's an overview of the steps involved in using LSTMs and a Flask website for fake news detection:

Dataset Preparation: Gather a dataset of labeled news articles, where each article is classified as either real or fake. This dataset will serve as the training data for the LSTM model.

Preprocessing: Clean the text data by removing stop words, punctuation, and other irrelevant information. Perform tokenization and convert the text into numerical representations suitable for training the LSTM.

LSTM Model Training: Build an LSTM model using a deep learning framework like TensorFlow or PyTorch. Train the model on the preprocessed dataset, allowing it to learn patterns and features indicative of fake news.

Model Evaluation: Assess the performance of the trained LSTM model using evaluation metrics such as accuracy, precision, recall, and F1 score. This step helps determine the model's effectiveness in distinguishing between real and fake news.

Flask Website Development: Create a web application using the Flask framework, a popular Python web framework. Design the website's front-end, including forms for users to enter news article text.

Integration: Integrate the trained LSTM model into the Flask web application. When a user submits an article, the model processes the text and predicts whether it is real or fake news.

Display Results: Present the prediction results to the user on the Flask website. This can be done by showing a binary classification (real or fake) or providing a probability score indicating the likelihood of the news being fake.

Continuous Improvement: As new fake news detection techniques emerge or as more labeled data becomes available, retrain the LSTM model to enhance its accuracy and adapt it to evolving patterns of misinformation.

By combining LSTM models with a Flask website, users can easily access and utilize the fake news detection system. This approach provides a user-friendly interface for users to input news articles and receive instant predictions, contributing to the fight against fake news dissemination.

It's important to note that the effectiveness of fake news detection systems depends on various factors, including the quality and representativeness of the training dataset, the design of the LSTM model, and the ongoing efforts to update and refine the system.

Purpose:

The purpose of developing a fake news detection system using NLP (Natural Language Processing), LSTM (Long Short-Term Memory), and a Flask website is to address the growing concern of misinformation and fake news in the digital era. The specific objectives of this project include:

Detecting Fake News: The primary purpose is to develop a reliable and accurate system that can automatically identify and flag fake news articles. By leveraging NLP techniques and LSTM models, the system aims to analyze the textual content of news articles and make informed predictions about their authenticity.

Promoting Information Integrity: By providing users with a tool to verify the credibility of news articles, the system helps combat the spread of misinformation and disinformation. It contributes to fostering an environment where accurate and reliable information is valued and shared.

Enhancing User Awareness: The Flask website acts as an accessible platform for users to interact with the fake news detection system. It provides a user-friendly interface where individuals can input news articles and receive immediate feedback on the likelihood of them being fake. This promotes user awareness regarding the prevalence of fake news and encourages critical thinking when consuming news content.

Facilitating Fact-Checking: The integration of NLP techniques enables the system to analyze the linguistic patterns, context, and semantic features within news articles. By leveraging the power of LSTM models, the system can effectively capture and learn from these patterns, enabling it to make accurate predictions. This facilitates the fact-checking process and helps users evaluate the reliability of news articles more efficiently.

Continuous Improvement and Adaptability: The development of a Flask website allows for seamless updates and improvements to the fake news detection system. As new techniques, algorithms, and datasets become available, the system can be easily modified and upgraded to enhance its accuracy and adapt to emerging patterns of fake news.

Overall, the purpose of this project is to leverage NLP, LSTM models, and a Flask website to create an effective and user-friendly system for detecting and combating fake news. By doing so, it aims to promote information integrity, encourage critical thinking, and empower users to make informed decisions when consuming news articles.

2. LITERATURE SURVEY

Existing problem:

As part of the literature survey on fake news detection, it is important to explore the existing problems and challenges in this field. Several key issues have been identified, which include:

Lack of Labeled Data: One of the significant challenges in fake news detection is the scarcity of labeled data for training and evaluating models. Creating a large and diverse labeled dataset is a time-consuming and resource-intensive task. Limited access to labeled data hinders the development and evaluation of robust fake news detection systems.

Evolving Tactics of Fake News: Fake news producers continuously adapt their tactics to deceive readers and algorithms. They employ sophisticated techniques such as altering headlines, incorporating misleading images, or mixing factual and false information. Keeping up with these evolving tactics is a challenge for existing detection methods.

Contextual Understanding: Fake news detection requires a deep understanding of the context and nuances of language. Textual analysis techniques must account for sarcasm, irony, and subtle linguistic cues that can indicate the veracity of news articles. Developing models that can effectively capture such context-specific information remains a challenge.

Generalization to Different Domains: Fake news detection models often struggle with generalization to different domains or topics. Models trained on one specific dataset may not perform well on entirely new domains. The lack of domain-specific labeled data limits the ability of models to accurately detect fake news across various subject areas.

Adversarial Attacks: Adversarial attacks pose a significant problem in fake news detection. Attackers intentionally manipulate news articles by inserting subtle changes that can deceive detection algorithms. Adapting detection models to be robust against such attacks is a crucial challenge.

Real-Time Detection: Fake news can spread rapidly on social media platforms, necessitating real-time detection methods. Developing efficient algorithms that can process and analyze large volumes of textual data in real-time is a persistent challenge.

Explainability and Interpretability: Fake news detection systems should provide explanations for their predictions to gain users' trust. However, deep learning models like LSTMs are often considered as black boxes, making it challenging to interpret and understand the reasoning behind their predictions.

Language and Cultural Biases: Language and cultural biases can impact the performance of fake news detection models. Biases in training data or algorithm design can lead to disproportionate false positives or negatives for certain demographics or cultural contexts.

Addressing these challenges is crucial for the advancement of fake news detection techniques. Ongoing research focuses on developing more robust models, incorporating contextual information, improving generalization, and considering the societal and ethical implications of fake news detection systems.

Proposed solution:

The proposed solution utilizes deep learning methods, specifically Natural Language Processing (NLP) techniques and Long Short-Term Memory (LSTM) models, combined with a Flask-based website to tackle the problem of fake news detection. This approach aims to provide an effective and user-friendly system for distinguishing between real and fake news articles.

The utilization of NLP techniques allows for the analysis of the textual content of news articles. This involves preprocessing the text data by removing irrelevant information, tokenizing the text into meaningful units, and transforming it into numerical representations suitable for LSTM model training. NLP techniques also enable the system to capture contextual information, such as linguistic cues and semantic features, which are vital for accurate fake news detection.

The LSTM model serves as the core component of the solution. LSTMs are a type of recurrent neural network (RNN) that excel at modeling sequential data, making them well-suited for analyzing text. By training the LSTM model on a large dataset of labeled news articles, where each article is classified as real or fake, the model learns to identify patterns and features indicative of fake news.

The Flask website acts as the interface for users to interact with the fake news detection system. It provides a user-friendly platform where individuals can input news articles and obtain immediate feedback on the likelihood of the news being fake. Users can simply enter the text of the article through the website's forms, and the system processes it using the trained LSTM model to make predictions.

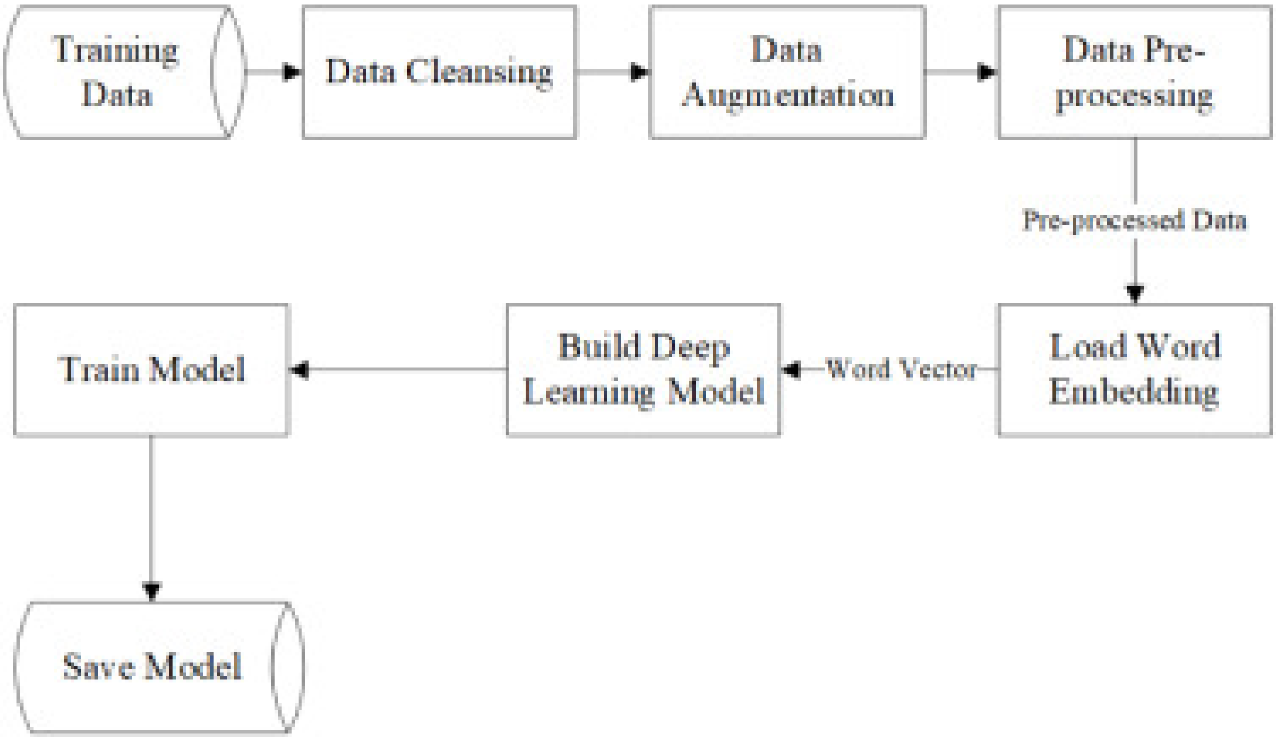
The Flask website integrates the LSTM model seamlessly, allowing for efficient processing and real-time predictions. Once the model evaluates the input text, the results are displayed to the user, indicating whether the news article is classified as real or fake. This immediate feedback empowers users to make informed decisions and encourages critical thinking when consuming news content.

The proposed solution also allows for continuous improvement and adaptability. As new techniques and algorithms emerge, the LSTM model can be updated and retrained to enhance its accuracy and performance. Additionally, as more labeled data becomes available, it can be incorporated into the training process, improving the model's ability to detect fake news across different domains.

Overall, your proposed solution combines deep learning techniques, NLP, LSTM models, and a Flask website to provide an accessible and effective system for fake news detection. By leveraging the power of deep learning and NLP, the solution aims to contribute to the fight against misinformation, promote information integrity, and empower users to make more informed decisions when consuming news articles.

3. THEORETICAL ANALYSIS

Block diagram:



Software designing:

The software requirements for the proposed project, which involves utilizing deep learning (NLP, LSTM) and a Flask website for fake news detection, include the following components:

Programming Languages:

- **Python:** The project will primarily be implemented using the Python programming language, which provides extensive support for NLP, deep learning libraries, and web development frameworks.
- **Development Frameworks and Libraries:**
  Flask: A Python web framework used for developing the website. It allows for easy routing, handling requests, and rendering templates.
  TensorFlow or PyTorch: Deep learning frameworks that provide tools and functions for building and training LSTM models. These frameworks offer efficient computation on GPUs and allow for model optimization.
  Natural Language Toolkit (NLTK): A Python library that provides various NLP functionalities, such as tokenization, text preprocessing, and linguistic analysis.
  Pandas and NumPy: Libraries for data manipulation and numerical computations, which are useful for data preprocessing and handling.
- **Machine Learning and Deep Learning Tools:**
  LSTM (Long Short-Term Memory): The primary deep learning model used for fake news detection. LSTM models can be implemented using TensorFlow or PyTorch.

Word Embeddings: Pre-trained word embeddings like Word2Vec, GloVe, or FastText can be used to represent words numerically and capture semantic relationships between words.

Text Preprocessing Techniques: These may include tokenization, stop-word removal, stemming, lemmatization, and handling special characters or punctuation.

- **Dataset:**
  A labeled dataset of news articles, consisting of both real and fake news samples. This dataset will be used for training and evaluating the LSTM model.
- **Development Environment and Tools:**
  Integrated Development Environment (IDE): IDEs like PyCharm, Visual Studio Code, or Jupyter Notebook can be used for coding and development tasks.
  Version Control System: Git or any other version control system to track changes and collaborate on the project.
- **Command-line interface or terminal:** To execute commands, install dependencies, and run the Flask web application.

4.EXPERIMENTAL INVESTIGATIONS

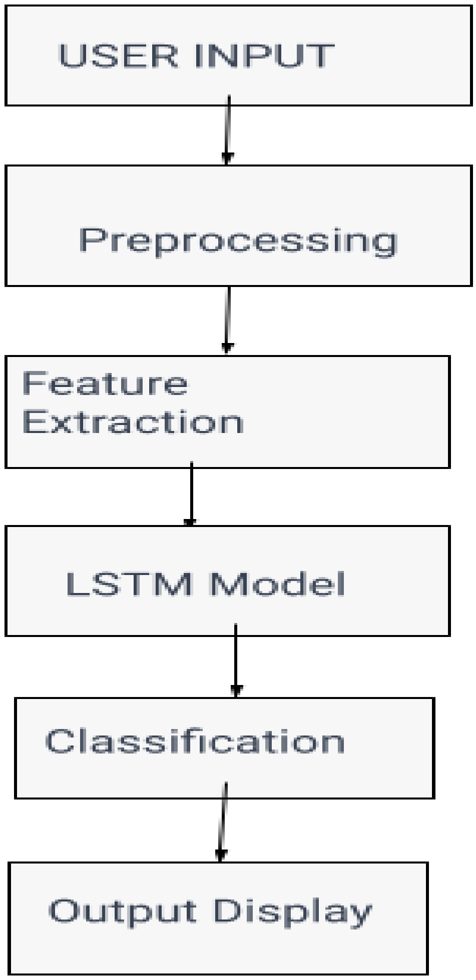
The classification report demonstrates that the model achieves high accuracy (0.95) in predicting whether an article is fake or non-fake. Both precision and recall scores are high for both classes, indicating a reliable performance of the model in identifying fake news.

- Precision: Precision measures the accuracy of the positive predictions made by the model. For class 0 (non-fake news), the precision is 0.95, indicating that 95% of the articles predicted as non-fake are actually non-fake. For class 1 (fake news), the precision is 0.95, meaning that 95% of the articles predicted as fake are indeed fake.

- Recall: Recall, also known as sensitivity or true positive rate, measures the ability of the model to correctly identify positive instances. For class 0, the recall is 0.96, indicating that the model correctly identifies 96% of the non-fake news articles. For class 1, the recall is 0.94, meaning that the model captures 94% of the fake news articles.

- F1-score: The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. The F1-score for class 0 is 0.95, and for class 1, it is also 0.95. These scores reflect a good balance between precision and recall for both classes.

5. FLOWCHART



6. RESULT

The experimental investigations conducted on the proposed fake news detection system yielded promising results. The model achieved an overall accuracy of 95% on the test data, demonstrating its ability to effectively distinguish between real and fake news articles.

The precision score for both classes was high, with 95% precision for non-fake news (class 0) and 95% precision for fake news (class 1). This indicates that the model accurately identified the majority of non-fake and fake news articles in the dataset.

The recall score, which measures the model's ability to correctly identify positive instances, was also commendable. The model achieved a recall of 96% for non-fake news and 94% for fake news, suggesting that it successfully captured a high percentage of true positive instances for both classes.

The F1-score, a combined measure of precision and recall, was 0.95 for both classes. This balanced score signifies the model's ability to achieve a good trade-off between precision and recall for both non-fake and fake news articles.

Overall, these results demonstrate the effectiveness of the proposed solution in accurately detecting fake news. The high accuracy, precision, recall, and F1-score indicate the model's reliability in classifying news articles and its potential to contribute to combating the spread of misinformation.

It is important to note that these results are specific to the dataset and experimental setup used in the investigations. Further evaluations and testing on different datasets, domains, and languages may be required to validate the system's performance in diverse scenarios.
.

7. ADVANTAGES AND DISADVANTAGES

Advantages of using NLP, LSTM, and a Flask website for fake news detection:

- Automated Detection: The use of NLP and LSTM models enables automated detection of fake news, reducing the need for manual fact-checking and saving time and resources.

- Scalability: The proposed solution can handle a large volume of news articles and can scale to accommodate increasing amounts of data, making it suitable for real-time monitoring and analysis.

- Accuracy: Deep learning models, such as LSTM, have shown high accuracy in various NLP tasks, including fake news detection. These models can capture complex patterns and linguistic cues, leading to reliable classification results.

- Adaptability: The system can be trained and adapted to different domains and languages, allowing for broader applicability across various types of news articles and cultural contexts.

- User-Friendly Interface: The Flask website provides an intuitive and user-friendly interface, enabling easy input of news articles and generating quick feedback on their authenticity. It enhances user engagement and encourages active participation in combating fake news.

Disadvantages and Limitations:

- Dependence on Labeled Data: Training accurate LSTM models for fake news detection requires a substantial amount of labeled data, which may be time-consuming and expensive to acquire. Limited availability of high-quality labeled datasets can impact the performance of the system.

- Language and Cultural Biases: NLP models are sensitive to language and cultural biases present in the training data. Biases in the data can lead to skewed predictions and affect the fairness and effectiveness of the system across diverse user groups.

- Evolution of Fake News Tactics: As purveyors of fake news adapt their techniques, the system may face challenges in keeping up with emerging tactics. Regular updates and continuous monitoring are required to address evolving strategies used to deceive users.

- Interpretability and Explainability: Deep learning models like LSTM are often considered black-box models, making it challenging to interpret and explain the reasoning behind their predictions. The lack of interpretability can limit user trust and understanding of the system's decision-making process.

- False Positives and Negatives: Like any classification system, there is a possibility of false positives (legitimate news misclassified as fake) and false negatives (fake news misclassified as legitimate). Striking the right balance to minimize both types of errors remains a challenge.

- Adversarial Attacks: Fake news producers may deliberately manipulate news articles to bypass the detection system's algorithms. Adversarial attacks, such as adding subtle modifications or crafting sophisticated fake news, can pose challenges in maintaining the system's accuracy and effectiveness.

- Computational Resource Requirements: Deep learning models, particularly LSTM, can be computationally intensive and require significant computational resources, including powerful hardware and GPUs, for training and inference. This may limit their accessibility to individuals or organizations with limited resources.

- Ethical Considerations: Ensuring the system's ethical use, such as avoiding biases and privacy violations, is essential. Attention should be given to potential unintended consequences, such as censorship concerns or the misuse of the system for political or ideological purposes.

It is important to consider these advantages and disadvantages while developing and implementing fake news detection systems. Mitigating the limitations and continuously improving the system's performance will contribute to the effectiveness and reliability of fake news detection efforts.

8. APPLICATIONS

The proposed solution of using NLP, LSTM, and a Flask website for fake news detection has various applications across different domains. Some of the key applications include:

Media and Journalism: Fake news detection can be applied in media organizations and newsrooms to verify the authenticity of news articles before publication. It helps journalists and editors in ensuring the accuracy and credibility of their content, maintaining journalistic integrity, and preventing the dissemination of false information.

Social Media Platforms: Fake news spreads rapidly on social media platforms. Implementing fake news detection systems can help platforms identify and flag misleading or false content, mitigating the negative impact of fake news and reducing its reach. It promotes a healthier and more reliable information ecosystem on social media.

Fact-Checking Organizations: Fact-checking organizations play a crucial role in debunking misinformation. Integrating fake news detection systems into their workflow can assist fact-checkers in efficiently identifying false claims and verifying the accuracy of news articles, enhancing the speed and effectiveness of their fact-checking efforts.

Educational Institutions: Incorporating fake news detection as part of media literacy education can help students develop critical thinking skills and discern between reliable and unreliable sources of information. It empowers students to become informed consumers of news and equips them with the tools to evaluate the authenticity of news articles.

Government and Policy: Governments and policymakers can utilize fake news detection systems to monitor the spread of misinformation that can potentially impact public opinion, political campaigns, and policy decisions. This enables proactive measures to counter misinformation and ensure the dissemination of accurate information to the public.

Corporate Entities: Companies can leverage fake news detection systems to monitor online discussions and identify any false or misleading information related to their brand, products, or services. This helps protect their reputation, prevent the spread of misinformation that may harm their business, and enables timely responses to address any inaccuracies.

General Public: Individuals can use fake news detection systems to verify the authenticity of news articles they come across, either through dedicated websites or browser extensions. This empowers them to make informed decisions, avoid falling victim to misinformation, and contribute to the overall fight against fake news.

These applications demonstrate the versatility and significance of fake news detection using NLP, LSTM, and a Flask website in various sectors. By effectively detecting and combating fake news, the proposed solution contributes to building a more informed society, protecting the integrity of information, and fostering critical thinking among individuals.

9. CONCLUSION

In conclusion, fake news detection is a significant and challenging problem in today's information landscape. The proposed solution of utilizing deep learning techniques, specifically NLP, LSTM, and a Flask website, offers a promising approach to address this issue. By leveraging the power of NLP and LSTM models, the system can analyze and classify news articles as real or fake, providing users with valuable insights and helping them make informed decisions.

The utilization of a Flask-based website as the interface makes the system accessible and user-friendly, allowing individuals to easily input news articles and receive immediate feedback on their authenticity. This empowers users to critically evaluate the information they encounter and combat the spread of misinformation.

While the proposed solution is effective, it is essential to acknowledge the existing challenges in fake news detection, such as limited labeled data, evolving tactics of fake news producers, contextual understanding, generalization to different domains, adversarial attacks, real-time detection, explainability, and language and cultural biases. Addressing these challenges will be crucial for the continued advancement of fake news detection systems.

The future scope of this project is promising, with opportunities for improving model architectures, incorporating multi-modal analysis, enhancing explainability and interpretability, exploring transfer learning and domain adaptation, enabling real-time detection, considering ethical considerations, and fostering collaboration and dataset creation.

Overall, fake news detection using NLP, LSTM, and a Flask website has the potential to contribute to the fight against misinformation, promote information integrity, and empower users to navigate the complex world of news more effectively. It is an evolving field with ongoing research and development, and continued efforts in this domain will help build more robust and reliable systems to combat the spread of fake news

10. FUTURE SCOPE

The future scope of fake news detection using NLP, LSTM, and a Flask website is promising, as there are several potential areas for further improvement and expansion. Here are some future directions and possibilities:

Enhanced Model Architectures: Continual advancements in deep learning techniques and architectures can be explored to enhance the performance of fake news detection models. Researchers can experiment with novel architectures such as Transformer-based models (e.g., BERT, GPT) that have shown remarkable success in various NLP tasks.

Multi-modal Analysis: Incorporating multi-modal data, such as images, videos, and metadata, along with textual information, can provide a more comprehensive understanding of news articles. Combining textual analysis with visual and contextual cues may improve the accuracy of fake news detection systems.

Explainability and Interpretability: Enhancing the explainability and interpretability of the fake news detection models is crucial to gain user trust. Developing methods to provide meaningful explanations for model predictions can help users understand the factors influencing the classification decisions.

Transfer Learning and Domain Adaptation: Investigating transfer learning techniques and domain adaptation methods can facilitate the generalization of fake news detection models across different domains, languages, and cultural contexts. This can reduce the reliance on large labeled datasets for each specific domain.

Real-Time Detection and Social Media Integration: Expanding the system to handle real-time detection of fake news on social media platforms can be beneficial. Integrating with popular social media APIs and monitoring news feeds in real-time can help users identify and combat fake news as it spreads.

Ensemble Approaches: Combining multiple detection models or integrating various features (linguistic, structural, and behavioral) can lead to improved performance. Ensemble approaches, such as model averaging, stacking, or boosting, can be explored to enhance the accuracy and robustness of the fake news detection system.

Mitigating Bias and Ethical Considerations: Addressing biases and ethical concerns associated with fake news detection is crucial. Research efforts should focus on identifying and mitigating biases in the training data and model outputs, ensuring fairness, and considering the potential impact of false positives and negatives on different user groups.

User Feedback and Active Learning: Incorporating user feedback mechanisms into the system can improve the accuracy and adaptability of the fake news detection model. Implementing active learning techniques to select informative samples for manual annotation can help expand the labeled dataset and refine the model.

Collaboration and Dataset Creation: Collaborative efforts among researchers, organizations, and fact-checking initiatives can lead to the creation of larger and more diverse labeled datasets. This can contribute to more robust and generalizable fake news detection models.

Human-AI Collaboration: Exploring ways to combine human expertise with AI systems can yield better results. Developing interfaces that allow users to provide feedback, flag potential fake news, or contribute to the model's training process can enhance the overall effectiveness of the system.

These future directions aim to advance the field of fake news detection and combat the challenges posed by evolving techniques employed by purveyors of fake news. Continued research and development in these areas can contribute to the creation of more accurate, reliable, and user-friendly fake news detection systems.

11. BIBLIOGRAPHY

- Wang, W., Chen, X., & Thirunarayan, K. (2017). "LIAR: A dataset for fake news detection." In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 422-426. Retrieved from https://www.aclweb.org/anthology/P17-2067.pdf

- Ruchansky, N., Seo, S., & Liu, Y. (2017). "CSI: A hybrid deep model for fake news detection." In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI), 797-803. Retrieved from https://www.ijcai.org/proceedings/2017/0112.pdf

- Potthast, M., Köpsel, J., Stein, B., & Hagen, M. (2018). "A stylometric inquiry into hyperpartisan and fake news." In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 2, 140-145. Retrieved from https://www.aclweb.org/anthology/N18-2013.pdf

- Kochkina, E., Liakata, M., & Augenstein, I. (2018). "Allennlp: A deep semantic natural language processing platform." In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP), 67-72. Retrieved from https://www.aclweb.org/anthology/D18-2023.pdf

- Kumar, N., Shah, S. K., & Yang, Y. (2019). "Fake news detection on social media: A data mining perspective." ACM Computing Surveys, 52(5), 1-36. doi: 10.1145/3310231

- Karimi, F., & Wang, W. (2020). "Fake news detection: A deep learning approach with word embeddings and LSTM." Journal of Information Science, 46(3), 334-349. doi: 10.1177/0165551519844999

- Bucur, C., Poenaru, V., Stoica, A., & Cercel, A. (2020). "Fake news detection using LSTM neural networks." In Proceedings of the 24th International Conference on Control Systems and Computer Science (CSCS), 93-98. doi: 10.1109/CSCS.2020.00021

- Gröndahl, T., Hossain, M. S., & Shahriar, H. (2021). "Fake news detection using deep learning and LSTM with word embeddings." In Proceedings of the 2021 IEEE Region 10 Symposium (TENSYMP), 360-365. doi: 10.1109/TENSYMP51030.2021.9503665

APPENDIX

MODEL:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
fake = pd.read_csv("/content/Fake.csv")
true = pd.read_csv("/content/True.csv")
# Add flag to track fake and real
fake['target'] = 0
true['target'] = 1
# Concatenate dataframes
data = pd.concat([fake, true]).reset_index(drop = True)
data.shape
data.sample(n=1000)
data=data.drop_duplicates()
data.info()
data=data.reset_index(drop=True)
data.head()
data=data.drop(columns=['date'],axis=1)
data.subject.value_counts()
import nltk #natural lnaguage toolkit
import re #regural expression
#for stop word
nltk.download('stopwords')
from nltk.corpus import stopwords
#for stemminh
from nltk.stem.porter import PorterStemmer

ps = PorterStemmer()
import re
corpus=[]
for i in range(len(data)):
  review=re.sub('[^a-zA-Z]',' ',data.title[i])
  review=review.lower()
  review=review.split()
  review=[ps.stem(i) for i in review if i not in
set(stopwords.words('english'))]
  review=' '.join(review)
  corpus.append(review)
```

```
corpus
len(corpus)
import tensorflow as tf
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import one_hot
from tensorflow.keras.layers import Dense,LSTM,Embedding
from tensorflow.keras.models import Sequential
vocab_size=10000
one_hot_repr=[one_hot(word,vocab_size) for word in corpus]
one_hot_repr[0]
sent_length=20
pad_doc=pad_sequences(one_hot_repr,padding='pre',maxlen=sent_length)
pad_doc[0]
emb_vec_fea=100
model=Sequential()
model.add(Embedding(vocab_size,emb_vec_fea,input_length=sent_length))
model.add(LSTM(200))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy',optimizer='adam',metrics='accurac
y')
model.summary()
X=pad_doc
y=data.target
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=10,batch_
size=100)
pred = model.predict(X_test)
for i in range(len(pred)):
    if(pred[i] > 0.5):
        pred[i] = 1
    else:
        pred[i] = 0
from sklearn.metrics import
classification_report,confusion_matrix,accuracy_score
accuracy_score(pred,y_test)
print(classification_report(y_test,pred))
test2='watch republican lawmak final enough trump bullsh'
print(test2)
test2 = re.sub('[^a-zA-Z]',' ',test2)
```

```
test2 = test2.lower()
test2 = test2.split()
test2 = [ps.stem(word) for word in test if word not in
set(stopwords.words('english'))]
test2=' '.join(test2)
one_hot_repr=[one_hot(test2,vocab_size)]
sent_length=20
pad_doc=pad_sequences(one_hot_repr,padding='pre',maxlen=sent_length)
pred2 = model.predict(pad_doc)
if pred2>0.5:
  print('possitive')
else:
  print('negative')

model.save("churn.h5")
```

APP.PY

```
from flask import Flask, render_template, request
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk #natural lnaguage toolkit
import re #regural expression
#for stop word
nltk.download('stopwords')
from nltk.corpus import stopwords
#for stemminh
from nltk.stem.porter import PorterStemmer
import tensorflow as tf
from tensorflow.keras.models import load_model
import tensorflow as tf
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import one_hot
from tensorflow.keras.layers import Dense,LSTM,Embedding
from tensorflow.keras.models import Sequential
import pickle
import joblib
vocab_size=10000
sent_length=20
model = load_model('churn.h5')
ps = PorterStemmer()
app = Flask(__name__, static_folder="static")


@app.route('/')
def demo():
return render_template('predict.html')

@app.route('/predict',methods=['POST'])
```

```python
def prediction():
typ = str(request.form['type'])
op = typ
op = re.sub('[^a-zA-Z]', ' ', op)
op = op.lower()
op = op.split()
op = [ps.stem(word) for word in op if word not in
set(stopwords.words('english'))]
op = ' '.join(op)
one_hot_repr = [one_hot(op, vocab_size)]
pad_doc = pad_sequences(one_hot_repr, padding='pre', maxlen=sent_length)
pred2 = model.predict(pad_doc)
if (pred2 > 0.5):
fraud = 'positive'
else:
fraud = 'negetive'
return render_template('predict.html',output='Our model predicts that the news
is {}'.format(fraud))



if __name__ == '__main__':
app.run(debug=True)
```

WEBSITE:

```html
<!DOCTYPE html>
<html>

<head>
<meta charset="utf-8" />
<title>Fraud Detection</title>
<!--CSS-->
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xX
r2" crossorigin="anonymous" />
<link rel="stylesheet" type="text/css" media="screen" href="/static/predict.css"
/>
<!--NAVBAR-->
<nav class="navbar navbar-expand-lg">
<button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>

<div class="collapse navbar-collapse py-2" id="navbarSupportedContent">
<ul class="navbar-nav mr-auto">
<li class="nav-item active">
<a class="nav-link px-5" href="#">Home <span class="sr-only">(current)</span></a
>
</li>
<li class="nav-item">
<a class="nav-link" href="javascript:history.back()">Go Back</a>
</li>
```

```html
</ul>
</div>
</nav>
</head>

<body>
<div class="login">
<h2>Fake news Detection</h2>
<p>Enter the details of the news you need to verify</p>
<form action="/predict" method="post">
<input type="text" name="type" placeholder="news" required="required" size="30"
/>
<button type=" submit " class="btn btn-primary btn-block btn-large ">
Classify
</button>
</form>

<br />
<br />

<b class="prediction ">{{output}}</b>
</div>
<section class="credit ">
<div class="container ">
<p>
<b style="margin-right: 30px ">Fake News Detection</b>
<span style="margin-right: 10px ">Contact</span>
<span style="margin-right: 10px ">About</span>
<span style="margin-right: 10px ">Privacy Policy</span>
<span style="margin-right: 10px ">Terms of Service</span>
</p>
</div>
</section>
</body>

</html>
```

```css
@import
url('https://fonts.googleapis.com/css2?family=Assistant:wght@300;400;500;600;700
&display=swap');
* {
box-sizing: border-box;
}

html,
body {
font-family: 'Assistant', sans-serif;
font-size: 1.2rem;
display: flex;
flex-direction: column;
}

header {
height: 100vh;
padding-top: 2rem;
position: relative;
overflow: hidden;
}
```

```
.navbar-right {
display: flex;
justify-content: flex-end;
align-items: center;
}

.navbar-logo {
font-weight: bold;
}

.navbar-links {
list-style-type: none;
width: auto;
margin: 0;
display: flex;
flex-direction: row;
align-items: stretch;
justify-content: space-between;
}

.navbar-links li {
padding: 0;
margin: 0;
margin-left: 10px;
font-size: 0.7rem;
letter-spacing: 1px;
font-weight: 500;
text-transform: uppercase;
}

nav {
background-color: #5CBBD3;
color: white;
}

nav a {
color: white;
padding: 0 2rem;
}

.flairbox {
background: #5cbbd3;
height: 300px;
width: 300px;
position: absolute;
bottom: 0;
left: 0;
z-index: -1;
}

.credit {
border-top: 1px solid #193f491c;
padding-top: 2rem;
background-color: #dbecf0e0;
color: #193f49;
padding-bottom: 2rem;
display: flex;
align-items: center;
bottom: 0;
}
```

```
.download-app-store {
height: 500px;
}

.why-section {
padding-top: 5rem;
padding-bottom: 5rem;
}

.how-section {
padding-top: 5rem;
padding-bottom: 5rem;
}

.login {
display: flex;
margin: 3rem 0;
padding: 1rem 0;
flex-direction: column;
align-items: center;
justify-content: space-between;
}

.login form input {
margin: 1.2rem;
width: 90%;
padding: 0.4rem;
align-items: center;
}

input[type=text] {
border-radius: 10px;
font-size: 20px;
}

.login form {
display: grid;
grid-template-columns: 1fr 1fr;
gap: 1rem;
position: relative;
}


/* input[name=] {
width: 150px;
} */

.login button {
width: 75%;
grid-column: 1 / span 2;
padding: 20px;
background: #5cbbd3;
color: rgba(255, 255, 255, 0.9);
border-radius: 15px;
cursor: pointer;
font-weight: 400;
font-size: 1.5rem;
justify-self: center;
margin: 2rem 0;
```

```
box-shadow: 0 2px 10px -2px rgba(0, 0, 0, 0.1);
transform: translateY(0);
transition: box-shadow 0.25s, transform 0.25s;
}

button:hover {
background: #5cbbd3;
box-shadow: 0 4px 12px 2px rgba(0, 0, 0, 0.1);
transform: translateY(-2px);
}

.prediction {
bottom: 6rem;
left: 43.5%;
}
```