Fake News Detection using Sentiment Analysis: A Natural Language Processing Approach

Kevin Singpurwala
School of Computer Science
University College Dublin
Dublin, Ireland
Email: kevin.singpurwala@ucdconnect.ie

Jianhao Feng
School of Computer Science
University College Dublin
Dublin, Ireland
Email: jianhao.feng@ucdconnect.ie

Natasha Suman
School of Computer Science
University College Dublin
Dublin, Ireland
Email: natasha.suman@ucdconnect.ie

Anju Krishna
School of Computer Science
University College Dublin
Dublin, Ireland
Email: anju.krishna@ucdconnect.ie

AhmedFaizel Asharafali
School of Computer Science
University College Dublin
Dublin, Ireland
Email: ahmedfaizel.asharafali@ucdconnect.ie

Madhav Mohan
School of Computer Science
University College Dublin
Dublin, Ireland
Email: madhav.mohan@ucdconnect.ie

***Abstract*—Fake news is the spreading of misinformation or disinformation. The proliferation of fake news on social media makes it difficult for one to procure reliable information online. Our motive is for information to be freely and widely available to all, while maintaining the integrity of said information.**

Ground truth refers to the actual reality or a baseline truth. Existing research focuses on the importance of deciphering the ground truth value and in to a feature-oriented approach. [1] This paper contributes to the existing body of literature by meticulously selecting features and leveraging emotional cues through natural language processing (NLP), to address the issue of fake news detection. This is done by availing of the Truth seeker and Liar data sets [1] [2] . The models used are an ensemble model which is a combination of Naive Bayes, passive aggressive classifier with a layer of deep neural network and Bi-directional LSTM (Long Short-Term Memory) networks with sentiment analysis.

Our findings suggest that sentiment analysis can serve as a valuable tool for enhancing the detection of fake news.

***Index Terms*—Sentiment; Social Media; Classification; Ground Truth; Machine Learning; Fake News; Feature extraction;**

I. Introduction

Misinformation and disinformation are pervasive in the modern digital era. It permeates every layer of society. Fake news deceives people at an ever increasing rate [2].

AI has the capability of generating articles and rendering images that are convincingly realistic [3]. This can create a very large number of fake news articles that can not be manually reviewed [4]. Thus, it is crucial to automate the process of fake news detection. It is of utmost importance for individuals to possess tools that can help them distinguish between what is real and what is not. Knowledge wields power, and information is the gateway to knowledge. Without accurate information, one would be left paralysed, unsure of what to believe and what not to believe. This can be a nuisance in normal times and have dire consequences in times where urgent action is required. Effective solutions are needed that can distinguish between real news articles and fake ones to mitigate the impact of fake news.

The pervasive dissemination of false information can have serious consequences, including triggering public panic during times of crisis, influencing elections, and weakening the public's trust in institutions. The main technical objective of this project is to create a real time sentiment analysis-based fake news detection system that is capable of distinguishing between real news and fake news. In order to gain a better understanding of the context and intention behind news items, sentiment analysis entails the extraction of emotional tones, opinions, and attitudes from text. The system can not only capture textual patterns but also underlying emotional cues. These emotional cues may help in evaluating the accuracy of a news article. In this paper we discuss:

1) The existing solutions to fake news detection.
2) Our proposed solution and how it was evaluated.
3) The role of sentiment as a feature in fake news detection.
4) Future works.

Definition 1 (Fake News) - *Fake news is a news article that is verifiable as false. This includes misinformation and disinformation.*
Definition 2 (Misinformation) - *The unknowing spreading of false information*
Definition 3 (Disinformation) - *The deliberate spreading of false information*
Definition 4 (News Article) - *a piece if text that provides information on current events, issues, or developments*

II. User Stories

The Fake News Detection System addresses a wide range of use cases. It helps users make informed judgements, reduces the spread of false information, enhances the standard of

[1] https://www.unb.ca/cic/datasets/truthseeker-2023.html
[2] https://paperswithcode.com/dataset/liar

academic research, accelerates news reporting and empowers small businesses to make informed decisions by providing users with the means to verify the accuracy of incoming news articles. These benefits are further strengthened by user reviews, utilisation data, and case studies. These all demonstrate the tangible advantages of your product in enhancing users' daily tasks and decision making processes.

Daily users frequently find it difficult to distinguish between fake and real news articles. This can cause confusion, ignorance, and sometimes harmful judgements based on incorrect information. By using this system, users can submit news articles to the Fake News Detection System for analysis. The system evaluates the reliability of a news article in real-time and delivers a classification outcome that indicates the likelihood of the content being genuine or fake news. Using reliable information, users may now make informed decisions. For instance, a user can check the accuracy of an article before reading it, preventing the spread of fake medical advice and related health dangers.

On social media, users routinely share information without checking the credibility of the source material. Thereby, accidentally spreading false information. Users can actively stop the spread of false information with the help of our system. One can contribute to a more trustworthy online information environment by preventing the transmission of false information.

Finding trustworthy resources for research and assignments can be difficult for educators and students alike. This can result in mistakes in academic work. Users can confirm the reliability of news stories and sources, improving the quality of data for research projects. Students make academic work of higher quality using reliable sources. This improves grades and learning outcomes. Teachers can also direct students toward reliable sources of information while promoting critical thinking.

News reporting is considerably delayed because journalists and researchers frequently take a long time to confirm the reliability of news sources. Our fake news detection system examines reports and statements fast and provides a credibility rating in real-time. By expediting the verification process, journalists can create news stories more rapidly without compromising accuracy. Researchers can focus more directly on cases of interest as the system can filter and indicate which articles are more likely to be truthful or fake.

- Users must get real time fast responses as to whether the news is real or fake.
- User wants to be able to assert the reliability of a news article.
- User wants to know why the article is true/false.
- User wants to be able to interact with the app seamlessly.
- User wants to detect fake news as they traverse the web.
- User wants to paste URL of the news article and decipher whether it is true or false.
- User wants to be able to tell the sentiment of the given article.
- User wants to be able to tell to detect fake news in their language of choice.

III. Literature Review

The detection of fake news is a pressing challenge in today's digital age. The surge of misinformation has led to significant research efforts aimed at devising effective mechanisms for its detection. One such method is the Rhetorical Approach[5], but the burgeoning potential of machine learning, especially deep learning models like recurrent neural networks, offers promising avenues [6].

Sentiment Analysis in Fake News Detection: Alonso et al. [7] delve into the domain of sentiment analysis as a tool for detecting fake news. Their work likely illuminates the unique sentiment patterns in misinformation, which may exhibit more polarized or biased emotional tones. The utilization of sentiment analysis techniques, perhaps leveraging deep learning, forms a potent tool to discern the authenticity of news.

Deep Learning and Sentiment Classification: The prowess of deep learning, especially recurrent neural networks, has been well-established in sentiment analysis tasks. Sharfuddin et al. present a comprehensive study on the BiLSTM (Bidirectional Long Short-Term Memory) model for sentiment classification. Their work underscores the significance of considering both past and future data in sequences to enhance contextual understanding in sentiment analysis tasks [8].

Versatility of BiLSTM: While the primary focus of our literature review is on fake news detection, it's noteworthy to mention the versatility of BiLSTM models. Abduljabbar et al.'s[9] research on freeway traffic forecasting with BiLSTM models exemplifies the broad applications of these models. Their methodologies and evaluations might offer insights into the robustness and efficacy of BiLSTM in varied scenarios.

Our Approach and Conclusions: Building on the foundational work of Chora et al.[6], we evaluated the veracity of entire articles in our research. We also incorporated ensemble models, as suggested by Shu et al.[1][10]. Our evaluations revealed that the LSTM model outperformed the ensemble models, a finding we confirmed in our implementation[3]. Specifically, a comparison of the receiver operating characteristic (ROC) curves for our LSTM and ensemble models indicated the superiority of the LSTM model. This observation aligns with findings from other studies[1][10]. Given its higher accuracy and enhanced capability to discern real news from fake, we chose the LSTM model for our final application. However, the journey to authenticating news is riddled with challenges. Shu et al.'s[1] premise that fake news is "intentionally and verifiable false" has been subjected to scrutiny, highlighting the inadvertent spread of misinformation. The multifaceted nature of fake news detection requires an amalgamation of techniques, and while Alonso et al.[7] have made strides in multilingual fake

[3]https://github.com/SuperSaiyansUcd/Fake-News-Detection-/tree/main/model

news detection, the challenge of addressing inherent biases in natural language processing remains[7].

In conclusion, the landscape of fake news detection is vast, with various methodologies showing promise [11]. Our research leans towards the LSTM and ensemble models, resonating with prevailing research trends and the promise they hold in this domain [1] [10].

IV. Implementation

This section describes the implementation of the fake news detection application. The application mainly consists of three parts: the front-end, the back-end and the model. [4].

For the front-end we looked in to React and Angular. The reason we went with React is that React is a library where as Angular is a framework. React is a JavaScript library known for its efficiency and flexibility in building user interfaces. Angular forces you to do things a certain "Angular" way where as React is more flexible. This flexibility helps us since not all of us on the team are familiar with Angular. Furthermore, we considered implementing million js. However, to benefit from the performance increase of million js, you need to follow a specific set of rules using blocks. If this is not done, the speed will revert back to that of regular react. Also, the heavy load of our system is on the model. The front-end does not have to be that performant. [5]

A. Data-Set

We selected two data-sets for our research: "Truth Seeker" [12] and the "LIAR" data-set [13].

a) Training: The models were primarily trained on the "Truth Seeker" data-set. An essential aspect of any fake news detection data-set is the binary "True" or "False" labels assigned to news items or statements. In the case of "Truth Seeker", labeling was conducted manually. Such labeled data facilitates the training of machine learning models, enabling them to distinguish between authentic and misleading news. These models are adept at recognizing patterns, terms, and other unique characteristics of fake news. Furthermore, the credibility of the sources within the data-set can be leveraged to ascertain the accuracy of the ground truth labels. A balanced data-set, comprising an equal mix of genuine and fake news articles, provides a solid foundation for evaluating model performance, including metrics such as accuracy, recall, precision, and F1-score.

b) Testing: For testing, we turned to the "LIAR" data-set. Our choice was influenced by the comprehensive nature of LIAR, which boasts over 12,800 statements vetted for their truthfulness by PolitiFact editors. Given its widespread recognition in the research community, the LIAR data-set stands as a benchmark for fake news detection algorithms. Model efficacy is often gauged by performance on this data-set. However, it's noteworthy that we refrained from using this data-set for training due to its unbalanced nature, which poses a risk of overfitting.

c) Feature Engineering: We undertook rigorous pre-processing of the data-set. Text data was streamlined for easy processing using tokenization. Additionally, we employed the Term Frequency-Inverse Document Frequency (TF-IDF) method to numerically vectorize the text, ensuring that the semantic essence of the content is encapsulated in its numerical representation. TF-IDF assigns weights to words based on their frequency in a given document relative to their occurrence across multiple documents, spotlighting particularly significant terms.

Sentiment analysis was another tool in our arsenal. This technique reveals the underlying sentiment or emotional undertone of a text string, offering insights into the perspectives of users, consumers, or any textual feedback. For this, we employed a gamut of tools: NLTK's Sentiment Intensity Analyzer (VADER), TextBlob, Flair's Text Classifier, Afinn, Pattern's sentiment module, and Transformers with BERT. Notably, BERT, a leading-edge transformer-based model, excels at grasping the contextual meaning of words in a text by examining their surroundings in the text.

B. The Front-end

To provide an intuitive and responsive user experience, the front-end of our Fake News Detection App has been designed using React. This report will discuss the orchestration of React components and techniques that make the web application.

We also built the web extension with React so that it can be easily integrated in to the main app in future releases. The web app is light weight and serves as a means of quickly highlighting and pasting content to the main web application.

a) Modular Design: Our app's user interface has been broken down into reusable components such as Line Spectrum, Radar Chart and Bar Chart. This modular approach makes it easy to organise different components, aids in maintainability and allows for flexible modifications.

b) State Management: Using React's useState hook, we've maintained a consistent state for the app. This allows data to be preserved when a user goes to a different page, where data needs to be stored for future reference by the user. The data is only removed when a user refreshes the home page. Then a confirmation modal window is presented that asks the user confirm this action. Thus, data is saved or removed in accordance with the user's desire.

c) User Feedback - Interactive Input Form: The web application has been designed to accept textual news input from users, or a link to the desired news article. Validation checks are implemented to ensure legitimate entries and to protect the back-end and model against attacks, such as cross site scripting; At the same time, it ensures the model receives meaningful inputs, inputs such as blank spaces will be warned as invalid inputs, users will be informed by the changing of the colour at edge of the text box, from black to red as well as a corresponding label informing them of what to add.

[4]Find our implementation here https://github.com/SuperSaiyansUcd/Fake-News-Detection-.

[5]million.js https://million.dev/, angular, https://angular.io/, react, https://react.dev/

d) Integration with Back-end Services: Utilising Axios, a JavaScript library, we make the front-end seamlessly communicate with our back-end services. Once a user submits a news article for verification, an asynchronous HTTP request is made thereafter. The request is sent to a REST endpoint to communicate with the back-end services and model.

e) Error Handling: In case of any discrepancies or server down-times, users are presented with clear error messages. This ensures transparency and builds user trust.

f) Dynamic Result Display: A Boolean flag represents a label using binary classification to display to the user whether the model evaluates the given news article as real news or fake news. The Line Spectrum, Radar Chart and Bar Chart components showcase the results. An arrow below the Line Spectrum points out whether the general overall sentiment of the news article. The Radar Chart and Bar Chart shows the sentiment of the news article. These displays change according to the output of the model.

g) User-centric Design: Burger Menus are displayed on our web pages to ensure users are always aware of their location on the application, and they can jump to their desired page by clicking the corresponding buttons without hassle.

h) Responsive Design: Leveraging CSS, our user interface adapts across a variety of devices - be it desktop, tablet, or mobile.

In conclusion, the front-end of our Fake News Detection App, powered by React, has been designed to offer a seamless and intuitive user experience. The modular design, paired with React's dynamic rendering capabilities, ensures that users can swiftly verify the legitimacy of news articles. With user-centric design, our front-end implementation stands as a testament to the modern standards of web applications.

C. The Back-end

The Back-end of our Fake News Detection App has been designed using Flask, a framework for developing web applications using python. This report will discuss the orchestration of Flask components and techniques that make the web application.

The choice between Flask and Django was a tough one to make. Flask has support for concurrent requests using its "@app.route" decorator. We picked flask due to its minimal and easy to use design while Django is more bulky [14]. Flask also maintains security as Flask's Jinja2 templates automatically escape cross site scripting attempts.

1) Flask Framework: The application is built on the Flask framework, which provides a robust foundation for handling HTTP requests and responses.
2) Sentiment Score Calculation: The back-end utilises multiple sentiment analysis libraries to compute sentiment scores for the content pasted by the user. These scores are used as a feature for the model. The sentiment analysis libraries include:
   - Afinn: Utilises a pre-built word list to calculate a sentiment score.
   - Pattern: Provides sentiment analysis based on patterns in the text.
   - NLTK's VADER: Analyses sentiment by considering the intensity of positive, negative, and neutral words.
   - TextBlob: calculates sentiment polarity based on the content.
3) Emotion Prediction Model: A pre-trained emotion prediction model is integrated to predicts Ekman's 6 basic emotions in addition to a neutral class present in the content [6].
4) Saved Model: The back-end includes a saved fake news detection model created using a combination of Bi-directional LSTM layers and dense layers. This model accepts content text, emotion features, and sentiment scores as inputs to predict the authenticity of news articles.
5) API Endpoints: The back-end defines two API endpoints: Web-Scrap API: Accepts a URL and extracts the title and main text content using Beautiful Soup. Submit-API: Accepts title and content, performs sentiment analysis, emotion prediction, and fake news detection, and returns analysis results as JSON.

The following steps outline the workflow of the Fake News Detection Web App back-end:

1) User submits a URL or textual content which is then sent to the appropriate API endpoint.
2) Sentiment analysis libraries compute sentiment scores for the content.
3) The sentiment model detects emotions present in the news article.
4) The fake news detection model combines sentiment scores, emotion features, and content text to determine the authenticity of a news article.
5) The results, including sentiment scores, emotion predictions, authenticity prediction, and various metrics, are sent in JSON format to the front-end.

D. The Model

This section outlines the implementation of a fake news detection model using a combination of Bi-directional LSTM (Long Short-Term Memory) networks and sentiment analysis. The model is designed to predict the authenticity of news articles based on textual content, emotions, and sentiment scores.

The first step is Data Preparation and pre-processing of the truthseeker data-set [7], which is the largest ground truth fake news analysis data-set for real and fake news content in relation to social media posts [12], containing information about news articles along with emotional and sentiment features. The following steps are done in order to create the ideal data-set for the project.

[6] https://huggingface.co/j-hartmann/emotion-english-distilroberta-base
[7] https://www.unb.ca/cic/datasets/truthseeker-2023.html

Sentiment scores are computed using various sentiment analysis libraries, including TextBlob, Afinn, and VADER. The relevant columns for the model are selected, including 'statement', 'tweet', 'BinaryNumTarget', emotion features ('anger', 'disgust', 'fear', 'joy', 'neutral', 'sadness', 'surprise'), and sentiment scores ('afinn', 'pattern', 'vader', 'textblob'). The emotions are calculated using the same hugging face transformer that we used on the back-end [8].

The next step is the model architecture which is composed of three main branches:

1) Textual Input Branch: The tweet and the statement of tweets are combined and processed using an embedding layer followed by a Bidirectional LSTM layer to capture sequential information.
2) Emotion Input Branch: Emotion features are passed through a Dense layer with 64 neurons.
3) Sentiment Scores Input Branch: Sentiment scores are passed through another Dense layer with 64 neurons. The outputs of the three branches are concatenated and fed through a Dense layer with 64 neurons, followed by the final output layer with a sigmoid activation function for binary classification (fake or real).

The final step is the model training and saving phase. The compiled 'best model' is trained using prepared data, optimising its ability to distinguish between fake and genuine news. After training, the model's learned weights are saved as a ".h5" file, ensuring that the trained model can be easily reused or deployed for future predictions without needing to retrain from scratch. The h5 file facilitates the deployment of the trained model as part of the web application's back-end. The system can then make predictions in real-time based on the input provided by users.

E. Dev-Ops

We deployed using the Microsoft Azure portal. This was done by availing of docker and the azure web-app server alongside a virtual machine (VM).

Using the docker app leads to more efficient use of resources, while the virtual machine(VM) was faster in overall speed. Using our CI/CD, pipeline the docker set-up was set to fetch from origin/main every hour. In this way we get continuous deployment. This is useful when someone on the team wants to look at a branch without loading up the code. One can go to the build of said branch which is deployed. Then one can test the deployed application for bugs. This is useful as there may be bugs during deployment that are not present on the deployed code. The VM was of Standard_B1ms size. The code was perpetually run using process manager 2(PM2). The network security settings on Azure makes sure that only certain ports can be accessed. This prevents unauthorised access and nefarious activity on non-secure ports.

To enable the real-time response of the model, both the ensemble model and LSTM model were pre-loaded as .h5 files. This is a highly compatible file format. Ideal for ease of collaboration. It is also quite compact and efficient [15]. The pre-loading enables us to give users real time responses with only seconds of delay(while the model is processing the news article). The concurrent ability of Flask allows us to give real time responses to multiple users simultaneously. This highly improves the user experience.

The Continuous Integration/Continuous Deployment (CI/CD) pipeline was set up on GitHub. Inside said pipeline, it runs the build, ensuring that deployment is successful. It also ensures that we have the Azure key credentials, that the front-end tests run and pass, that all the formatting is correct, that the system works integration testing wise, and that the back-end builds successfully.

V. Evaluation/results

A. The Front-end

This part outlines the evaluations conducted for the front-end of the Fake News Detection Application. The evaluations encompass user experience through surveys, functionality testing via Jest, and stress testing using JMeter.

1) *User Experience Evaluation:*
A survey was distributed to a diverse group of users to evaluate the user-friendliness of the user interface(UI). Through the survey result, we find that 90% of the users find the user interface good, while 10% gave neutral responses. 78% of the users are very clear about the functionality of this application without previous knowledge, while the other 22% gave neutral responses. They also gave valuable suggestions, such as enhance the contrast between text and background etc. In conclusion, the user application is user-friendly, and with just a few improvements, it meets the standards of an excellent user interface.
Dependabot keeps track of outdated dependencies that are a security concern. It creates a pull request to resolve these dependency vulnerabilities. If the pull request passes the CI/CD pipeline, we can merge it to main after a review and ensure our application remains secure.
2) *Functionality Testing:*
The Jest framework was employed to test the functionality of the application, focusing on unit tests for React components and integration tests for service calls. Every page is tested to show that they display and response in the expected way. The passing rate for functionality testing is 100%.
3) *Stress Testing:*
JMeter [9] was used to simulate multiple users accessing the application simultaneously to understand its scalability and identify potential performance bottlenecks. Through this investigation, we found that the application remained responsive and stable with up to 50 concurrent users. The application cannot take too much stress. The main reason being so is the limited bandwidth of the virtual machine, the only way to increase the bandwidth

[8] https://huggingface.co/j-hartmann/emotion-english-distilroberta-base

[9] Learn more about this system at https://jmeter.apache.org/.

is to pay. Optimising the model and pre-possessing steps can also help alleviate the strain on bandwidth.

Overall, the front-end of the Fake News Detection App exhibits a high degree of user-friendliness, functionality, integrity, and reasonable robustness under stress for a very small experimental application. This evaluation has provided valuable insights and clear pathways for further optimisation. Continuous evaluations in the future will ensure that the app remains at the forefront of usability and performance standards.

B. The Model

We have used evaluation methods using a classification methodology. These are F1 score, Precision, Recall and Accuracy. They provide a rigorous assessment of our model's effectiveness and its real world implications. We test the scalability of our solution to ensure it can handle larger data-sets and increasing user traffic. By using TensorFlow, our Bi-Directional LSTM model demonstrates efficient resource utilisation. We test the model on multiple data-sets of different sizes before finalising the truthseeker data-set. Our model demonstrated the ability to scale seamlessly without compromising performance. We conducted user surveys, gathered feedback, and refined our model based on user requirements and feedback. The user feedback pointed out the necessity of a user friendly interface and a simple solution on the result page without displaying overly technical outputs.

The main concern was finding the optimal threshold for the truthfulness score on the flask-server. We conducted a manual analysis where we knew the ground truth of the news article by using another data-set and testing with the system and we found that the optimal threshold for the truthfulness score is 0.64 which made the classification of real or fake news more effective. [10] The Evaluation approach is reasonable as we used traditional NLP evaluation metrics (precision, recall, accuracy, and F1-score) to quantitatively assess the model's performance on a test data-set. A stratified k-fold cross-validation was also used which provided robustness in estimating the model's generalisation of the classification. The results of our evaluation reveals the effectiveness of our fake news detection solution. Our LSTM model, augmented with emotion features and sentiment scores, achieved an accuracy of 0.98, outperforming baseline methods. The Evaluation metrics are given below.

Precision:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{1}$$

Recall:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2}$$

F1 Score:

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

Support: *the number of occurrences in the data-set*

TABLE I
Classification Report
LSTM Model With Sentiment

	Precision	Recall	F1 Score	Support
Fake News	0.99	0.97	0.98	13 053
Real News	0.97	0.99	0.98	13 786
Accuracy	-	-	0.98	26 839
Macro Average	0.98	0.98	0.98	26 839
Weighted Average	0.98	0.98	0.98	26 839

Precision: Precision measures the accuracy of positive predictions, as seen in equation 1. For class "Fake News", the precision is 0.99, which means that 99% of instances predicted as class "Fake News", were actually class "Fake News". For class "Real News", the precision is 0.97, indicating that 97% of instances predicted as class "Real News" were actually class "Real News".

Recall: Recall measures the ability of the model to correctly identify positive instances 2. For class "Fake News", the recall is 0.97, meaning that the model correctly identified 97% of the actual class "Fake News" instances. For class "Real News", the recall is 0.99, indicating that the model correctly identified 99% of the actual class "Real News" instances.

F1-Score: The F1-score is the harmonic mean of precision and recall 3. It provides a balanced measure of the model's performance. For both classes, the F1-scores are 0.98, indicating a good balance between precision and recall.

Initially, we thought this was an over-fitting issue. We attempted to resolve this by performing further feature extraction and k-fold cross-validation. Later on, in the work of [10] there was an experiment conducted where sentiment and emotions are used as features on multiple models and data-sets which resulted in similarly high results in accuracy. It is seen that LSTM models tend to have a high accuracy as observed by [6].

We also tested the model against another Bi-directional LSTM model which does not use Sentiment Analysis or Emotions as features and the following are the evaluation metrics. Although the model using sentiment analysis and emotions as features performs 1% better than the base model, we can say that use sentiment analysis and emotions is better. we can prove this by looking into other evaluation metrics like

TABLE II
Classification Report
LSTM Model Without Sentiment

	Precision	Recall	F1 Score	Support
Fake News	0.98	0.97	0.97	13 076
Real News	0.97	0.98	0.97	13 764
Accuracy	-	-	0.97	26 840
Macro Average	0.97	0.97	0.97	26 840
Weighted Average	0.97	0.97	0.97	26 840

[10] https://docs.google.com/spreadsheets/d/1g-8YPkHwzBwVx7bnDEkYgt_3zfkhTfqZT-9-bz1pfpA/edit?usp=sharing

calibration curve and confusion matrix The confusion matrix demonstrated the model's balanced performance in identifying both fake and real news on the model that uses Sentiment Analysis.

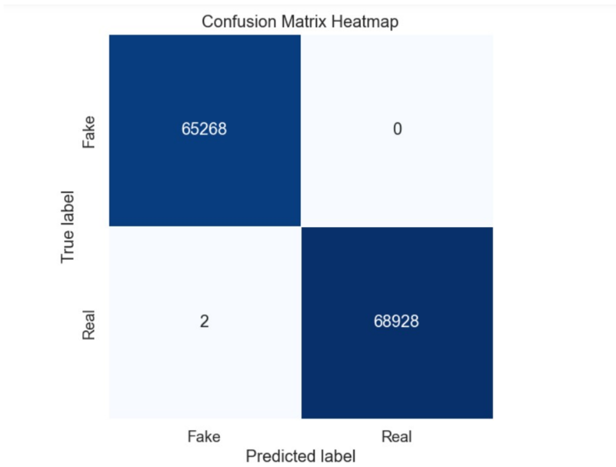

Fig. 1. Confusion Matrix SA model

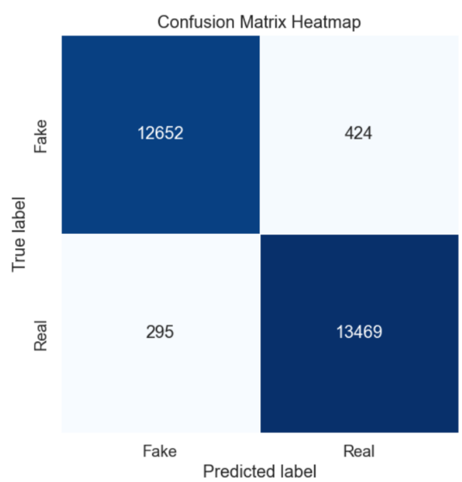

Fig. 2. Confusion Matrix BASE Model

The calibration curve showed a close alignment between predicted and true probabilities, validating the reliability of our model's ability to predict whether the news is real or fake on the sentiment analysis model.

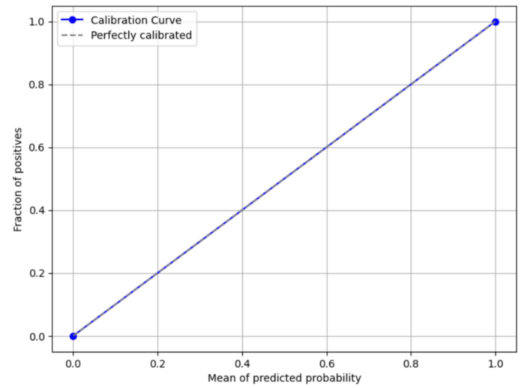

Fig. 3. Calibration Curve SA model

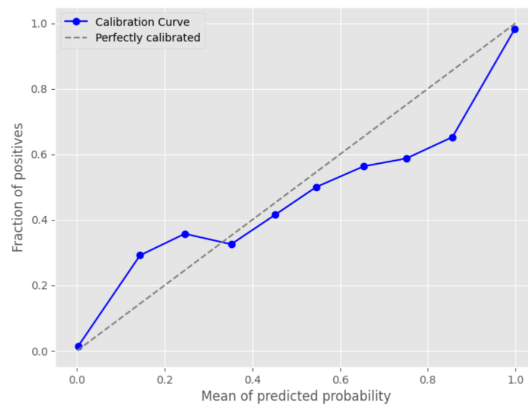

Fig. 4. Calibration Curve BASE model

In conclusion, we can say that using sentiment analysis as a feature can enhance the performance and reliability of the fake news detection model.

VI. Future Work

One future improvement would be to evaluate the truthfulness of each line and provide annotations as to why that statement is true or false. Currently the system evaluates based on all the content. Multi-language support is lacking in our current implementation. Our training data focuses on detecting fake news in the English language.

One big improvement that can be made is that of the data sources. We only avail of 2 data sets and this is not sufficient for general purpose fake news detection. Currently our system is only trained for the detection of fake news on social media.

The model can be improved by further refining what features to extract.

While limitations existed, the project's outcomes and insights provide a solid foundation for future improvements and extensions. [16] [1]

VII. Conclusions

Our project successfully united a diverse teams for the creation of the web application. At a high level, our solution involved creating a web application that detects fake news using sentiment analysis. Users can input news articles, and the system assesses the sentiment to determine the likelihood of authenticity. The collaboration among team members allowed us to seamlessly integrate front-end and back-end components, implementing multiple sentiment analysis methods for improved accuracy.With more time,we would further explore advanced sentiment analysis techniques and use Cross-functional training which could streamline our development process. Key successes include achieving a functional product within the project's scope and timeline. Effective communication and a flexible mindset ensured smooth coordination, underpinning the project's achievements. Overall, the Super-Saiyans showed the strength of teamwork, adaptability, and effective communication in achieving a successful partial solution to the fake news detection challenge as sentiment analysis cannot be the standalone solution.

VIII. Acknowledgements

Special thanks to our mentors at University College Dublin, namely Dr. Pavel Gladyshev, Dr. Timilehin Aderinola, Dr. Honghu Du and Dr. Simon Caton.

Also special thanks to our Microsoft mentors for providing us access to Azure which we used to deploy our web application.

References

[1] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD explorations newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[2] N. M. Brashier and D. L. Schacter, "Aging in an era of fake news," *Current directions in psychological science*, vol. 29, no. 3, pp. 316–323, 2020.

[3] N. C. Köbis, B. Doležalová, and I. Soraperra, "Fooled twice: People cannot detect deepfakes but think they can," *Iscience*, vol. 24, no. 11, 2021.

[4] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[5] N. K. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proceedings of the association for information science and technology*, vol. 52, no. 1, pp. 1–4, 2015.

[6] M. Choraś, A. Giełczyk, K. Demestichas, D. Puchalski, and R. Kozik, "Pattern recognition solutions for fake news detection," in *Computer Information Systems and Industrial Management: 17th International Conference, CISIM 2018, Olomouc, Czech Republic, September 27-29, 2018, Proceedings 17*, Springer, pp. 130–139.

[7] M. A. Alonso, D. Vilares, C. Gómez-Rodríguez, and J. Vilares, "Sentiment analysis for fake news detection," *Electronics*, vol. 10, no. 11, 2021, ISSN: 2079-9292. DOI: 10.3390/electronics10111348. [Online]. Available: https://www.mdpi.com/2079-9292/10/11/1348.

[8] A. Aziz Sharfuddin, M. Nafis Tihami, and M. Saiful Islam, "A deep recurrent neural network with bilstm model for sentiment classification," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, 2018, pp. 1–4. DOI: 10.1109/ICBSLP.2018.8554396.

[9] D. Abduljabbar Rusul L and P.-W. Hussein Tsai, "Development and evaluation of bidirectional lstm freeway traffic forecasting models using simulation data," *Scientific Reports*, 2021. DOI: 10.1038/s41598-021-03282-z. [Online]. Available: https://doi.org/10.1038/s41598-021-03282-z.

[10] S. K. Hamed, M. J. Ab Aziz, and M. R. Yaakub, "Fake news detection model on social media by leveraging sentiment analysis of news content and emotion analysis of users' comments," *Sensors*, vol. 23, no. 4, p. 1748, 2023.

[11] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–40, 2020.

[12] S. Dadkhah, X. Zhang, A. G. Weismann, A. Firouzi, and A. A. Ghorbani, "Truthseeker: The largest social media ground-truth dataset for real/fake content," 2023.

[13] W. Y. Wang, "" liar, liar pants on fire": A new benchmark dataset for fake news detection," *arXiv preprint arXiv:1705.00648*, 2017.

[14] D. Ghimire, "Comparative study on python web frameworks: Flask and django," 2020.

[15] M. Folk, G. Heber, Q. Koziol, E. Pourmal, and D. Robinson, "An overview of the hdf5 technology suite and its applications," in *Proceedings of the EDBT/ICDT 2011 workshop on array databases*, 2011, pp. 36–47.

[16] M. Fernández-López and M. Perea, "Language does not modulate fake news credibility, but emotion does," *Psicológica Journal*, vol. 41, no. 2, pp. 84–102, 2020.