DISASTER TWEETS BINARY CLASSIFICATION TASK USING TRADITIONAL MACHINE LEARNING AND DEEP LEARNING MODELS

1.0 INTRODUCTION AND DEFINITION

Natural disasters are extreme events caused by nature that can seriously disrupt lives and communities. Data about disasters collected through field surveys often isn't available right away. Yet, key information like the location, size, and extent of damage to buildings and infrastructure is vital for responding to disaster and management (Hao and Wang, 2020).

During disasters or emergencies, information shared on social media can be a reliable source of information to assess disasters, provide swift response, and other humanitarian efforts (Basit et al., 2023). Twitter is one of the biggest social networks that generates a huge amount of unstructured data from a plethora of sources (Udanor, Aneke and Ogbuokiri, 2016). By tracking tweets and hashtags, we can see what topics are trending and understand public conversations happening around the world. Therefore, it is often essential for first responders to rely on data from social media, as many users share information in various formats like images, text, and audio.

A major challenge in Natural Language Processing (NLP) is Ambiguity, which has been a tough challenge for NLP researchers for decades. While there has been some progress in resolving it, many important issues in this area remain unsolved (Alfawareh and Jusoh, 2011). Figuring out how to automatically sort through social media posts during natural disasters is also a problem. When something major happens, platforms like Twitter get flooded with messages - some are useful, others are not. Teaching a computer to understand which tweets are about the disaster and which ones are just noise can be tasking. Solving this could assist first responders and aid organizations get the right information faster when every second counts. Furthermore, organizations that provide humanitarian services depend on information from Twitter during disasters for situational awareness (Madichetty, Muthukumarasamy and Jayadev, 2021).

2.0 SCOPE

The project aims to build and test a model that can predict whether a tweet is related to a disaster, using a labelled dataset. This is in the domain of NLP because it deals with understanding and analysing human language, specifically tweets. It focuses on classifying whether a tweet is about a disaster or not. Since tweets are often messy, unstructured, and written in different styles, using NLP to clean and process this data is a solid approach to build models that can support real-time disaster response. Using advanced models like BiLSTM and GRU helps understand the context and meaning of words to tackle ambiguity in text.

3.0 IMPORTANCE

From 2000 to 2019, about 7,348 major disasters occured, which tragically claimed approximately 1.23 million lives and impacted 4.2 billion individuals. These events also caused economic losses of $2.97 trillion globally (UNDRR, 2020). Averagely 6,000 tweets are posted every second on Twitter that adds up to more than 350,000 tweets per minute, half a billion per day, and roughly quarter of a billion tweets each year (Internet Live Stats, 2024). These tweets can be noisy and filled with irrelevant content, when posted in real time (Alam

et al., 2018). How can we automatically classify tweets during a disaster to separate relevant posts from irrelevant ones?

By combining the fast-paced nature of social media with advancements in Machine Learning especially in Natural Language Processing (NLP) this project can tackle this challenge as a binary classification task.

The Qatar Computing Research Institute (QCRI) has developed several AI tools to tackle the challenges of using social media during disasters and support relief efforts. One of their key innovations is "Artificial Intelligence for Digital Response (AIDR)," which was created to help organizations quickly sort through social media posts and find useful information when responding to crises (Imran et al., 2014).

While sentiment analysis on social media has been widely explored, this topic is underexplored and there's been much less focus on applying it specifically within the context of disaster management (Ningsih and Hadiana, 2021)

4.0 BACKGROUND REVIEW

Researchers have previously used machine learning models like BERT and Logistic regression, among others, to help classify tweets related to disasters.

1. **Iparraguirre-Villanueva et al. (2023)** – The authors classified tweets about real natural disasters using machine learning models on posts tagged with "#NaturalDisasters". They used six algorithms- Random Forest (RF), K-Nearest Neighbours (KNN), Decision Tree (DT),Bernoulli Naive Bayes (BNB), Multinomial Naive Bayes (MNB). They followed a structured process: data import, exploration, cleaning, training, and testing. BNB, MNB, LR, and KNN achieved 87% accuracy, DT scored 82%, and RF got 75%. BNB, MNB, and LR performed best in identifying disaster-related tweets.

   Pros- The dataset contains tweets from recent disaster events with geolocation data, this is valuable because this information is often difficult to gather in the early stages of a disaster.

   Limitation- The study's reliance on the hashtag **#NaturalDisasters** may limit data diversity, excluding relevant tweets without the tag and leading to a potentially biased dataset.

2. **Singh and Saumya, 2019**- The authors aimed at classifying disaster-related tweets using a mix of deep learning (DL) techniques and traditional machine learning (ML). They experimented with seven ML algorithms— RF,Support Vector Machine, LR, KNN, NB, Gradient Boosting, and Decision Tree as well as five DL models—CNN, LSTM, GRU, Bi-GRU, and GRU-CNN. The tweets, collected from real disaster events like hurricanes, earthquakes, flood and wildfire were grouped and pre-processed. The ML models utilized different N-gram TF-IDF features, the DL models relied on word embeddings from GloVe and Crisis. Among the ML models, Gradient Boosting consistently performed best achieving an F1-score of 0.80

0.79, 0.67 and 0.70, for Earthquake, Hurricane, wildfire and Flood events respectively. However, across all disaster types, DL models outperformed the ML models overall, with different architectures excelling depending on the event.

Pros: The study compared a broad range of ML and DL models, giving a well-rounded understanding of their performance on disaster tweet classification.

Limitations: The dataset only includes tweets in English, which restricts its usefulness for multilingual or non-English-speaking regions.

3. **Nair, Ramya and Sivakumar, 2017-** The authors analyzed tweets using the hashtag #chennaiflood, from November 2015 to March 2016, and extracted 17 features per tweet. They utilized surrogate models to classify the content into five labels: Complaints , Relief Measures, Need for Help, Express Gratitude, Complaints, and Other. The classification was performed using DT, RF, and NB in the Weka tool. Among the models, RF achieved the best results, with a recall of 0.997, outperforming DT **(**0.979**)** and NB **(**0.688**).**

   Pros: The time-based analysis approach looked at changes in tweet activity before, during, and after the disaster, providing insight into how people communicate during different stages of a crisis.

   Limitations: While the study focused on tweets using the hashtag #chennaiflood only, it may have missed other relevant tweets.

4. **van den Bulk et al., 2022-** The authors applied eight ML models to classify articles as relevant or not in two food safety review cases—cereals and leafy greens. Models included SVM, NB, LR, and BERT. Results showed LR performed best on the cereals test set, while SVM and Naive Bayes led on future data and in the leafy greens case. They also tested 247 ensemble combinations, with the SVM + Naive Bayes ensemble achieving the highest average F1-score of 86.3%**,** reducing article screening workload by over 54% and irrelevant articles by 88%**.**

   Pros: The model can learn from each new batch of human-reviewed articles, improving performance over time.

   Limitations: Articles marked as “maybe relevant” weren’t used in training, limiting the model's exposure to borderline cases.

5. **Zhao and Sun, 2022** -The authors developed a machine learning model using BERT (Bidirectional Encoder Representations from Transformers) to predict review scores based on text descriptions from the Amazon Fine Food Reviews dataset. They focused on the growing importance of online reviews in consumer decision-making, especially in the eCommerce and food industries. They trained a fine-tuned BERT model for multi-class classification. An accuracy of 79.8% was achieved in the final model.

- **Pros:** The BERT model deployed captures the meaning behind user reviews.

- **Limitations:** The study did not address fake reviews, which could skew results. They acknowledged that filtering these out would likely improve accuracy.

5.0 SMART OBJECTIVES

The project approach met the SMART objectives:

- **Specific –** The study aimed to build a model that can identify accurately if a tweet is disaster related or not, using deep learning and machine learning approaches.
- **Measurable –** The goal was to reach at least 82% accuracy, evaluated through key metrics like F1-score, precision, recall with a baseline accuracy of at least 87% based on prior studies (Iparraguirre-Villanueva et al., 2023)
- **Achievable –** To meet the target, labelled datasets were used, NLP preprocessing was applied, class imbalance was handled with SMOTE, and models such as Naive Bayes, Logistic Regression, BiLSTM, and GRU were trained.
- **Relevant –** This is a relevant study because organizations that provide humanitarian services depend on information from Twitter during disasters for situational awareness (Madichetty, Muthukumarasamy and Jayadev, 2021)
- **Time-bound –** The entire project from data preprocessing to final evaluation and documentation was completed before the deadline May 1st, 2025.

6.0 DATASET

Source: The Disaster Tweets dataset was gotten from Kaggle Disaster Tweet Dataset NLP Task , each tweet was labelled as disaster-related (1) or not (0).

Shape:

	Training Set	Test Set
Shape	(7613, 5)	(3263, 4)
Memory Usage	0.20 MB	0.08 MB

Table 1: Dataset (Train and Test) Shape

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7613 entries, 0 to 7612
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   id        7613 non-null   int16
 1   keyword   7552 non-null   object
 2   location  5080 non-null   object
 3   text      7613 non-null   object
 4   target    7613 non-null   int8
dtypes: int16(1), int8(1), object(3)
memory usage: 200.9+ KB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3263 entries, 0 to 3262
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   id        3263 non-null   int16
 1   keyword   3237 non-null   object
 2   location  2158 non-null   object
 3   text      3263 non-null   object
dtypes: int16(1), object(3)
memory usage: 83.0+ KB
```

Figure 1: Dataset (Train and Test) Description

6.1 DATA PREPROCESSING

Tweets are often short, messy, and written informally, it is important to clean and structure the text before it is fed into a machine learning model. This project, which aims to classify tweets related to disasters, uses several key preprocessing techniques to handle the unstructured nature of social media data:

```
Null values in training set:
id             0
keyword       61
location    2533
text           0
target         0
dtype: int64

Null values in test set:
id             0
keyword       26
location    1105
text           0
dtype: int64
```

Figure 2: Missing Values in Dataset

6.1.1 DATA CLEANING - The columns with missing values; Location *and* Keyword aren't needed for this analysis, so no value was imputed**.** The tweet text was standardized as a cleaning technique before analysis. This removes URLs, emojis, HTML tags, and punctuation (while temporarily preserving hashtags), then converts the text to lowercase for consistency. It also removes words containing numbers to reduce noise and finally strips out the hashtag symbol itself while keeping the associated word (e.g., #earthquake becomes earthquake). This helps ensure the text is in a cleaner, more uniform format, making it easier for a machine learning model to extract meaningful patterns.

To prepare the tweets for classification, several key NLP preprocessing steps were carried out to clean and structure the text data in a way that models could understand:

- **Tokenization** -This technique was used to break each tweet into smaller parts called tokens—usually words or subwords. For example, a tweet like *"heard about earthquake"* was split into ["Heard", "about", "earthquake"]. This allowed the model to analyze the text word by word, which is essential when trying to detect patterns that indicate a disaster.
- **Stop Words Removal** – Many tweets contained common words like *"is"*, *"about"* and *"the"*, which don't contribute much meaning in the context of disaster detection. This stopwords was removed for noise reduction in the data, to allow the model focus more on important keywords like *"earthquake"* or *"fire"*.
- **Lemmatization** was used to set the words to their base form. Tweets with different variations of the same word, such as *"lightning"* were transformed to *"light"*, helping

the model group similar terms and better generalize across tweets with different wording but similar meaning.

- **Vectorization** was the final step before modelling. Machine learning models require numerical input, this converted the cleaned text into numbers by utilizing **TF-IDF** and **word embeddings**. These methods captured the importance and context of each word, turning tweets into vectors that the models could learn from to predict whether the tweet was disaster related or not.

6.1.2 DATA AUGMENTATION- SMOTE

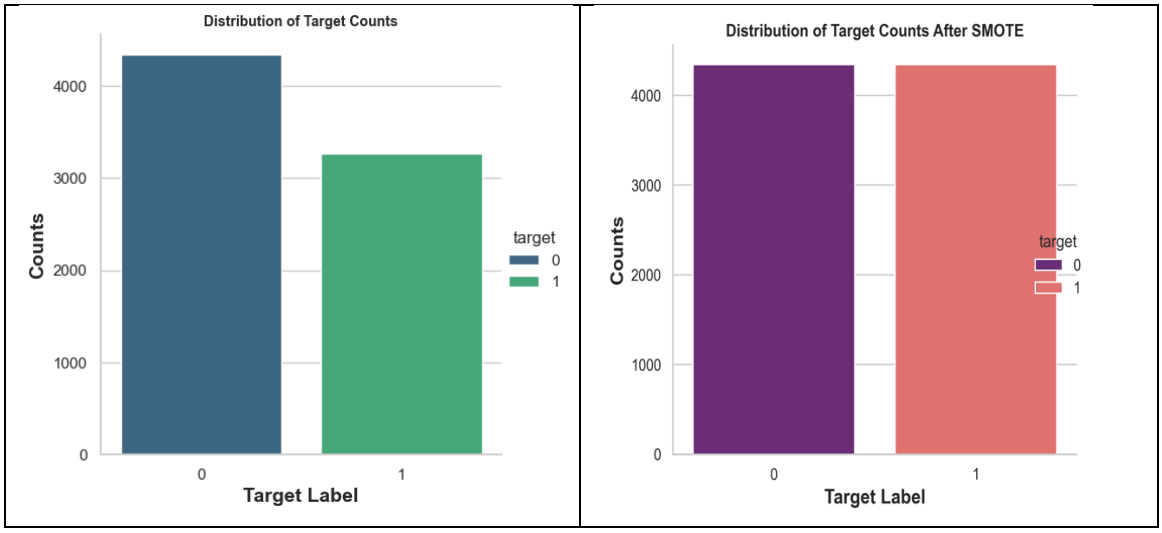

Figure 3: Distribution of Target Count before and after SMOTE

7.0 EXPLORATORY DATA ANALYSIS

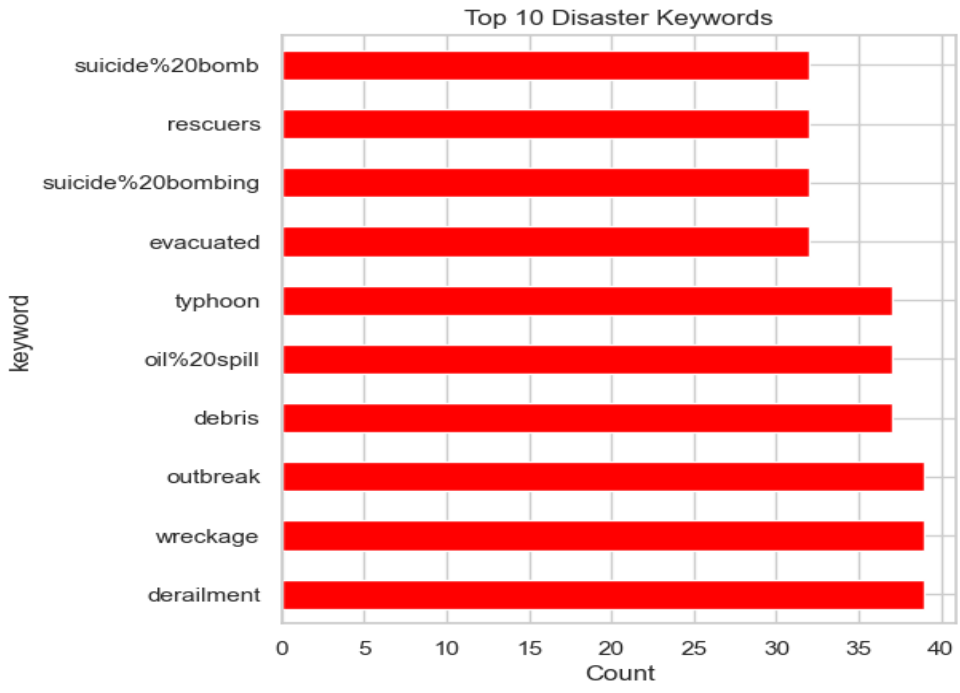

Figure 4: Top 10 Disaster Keywords

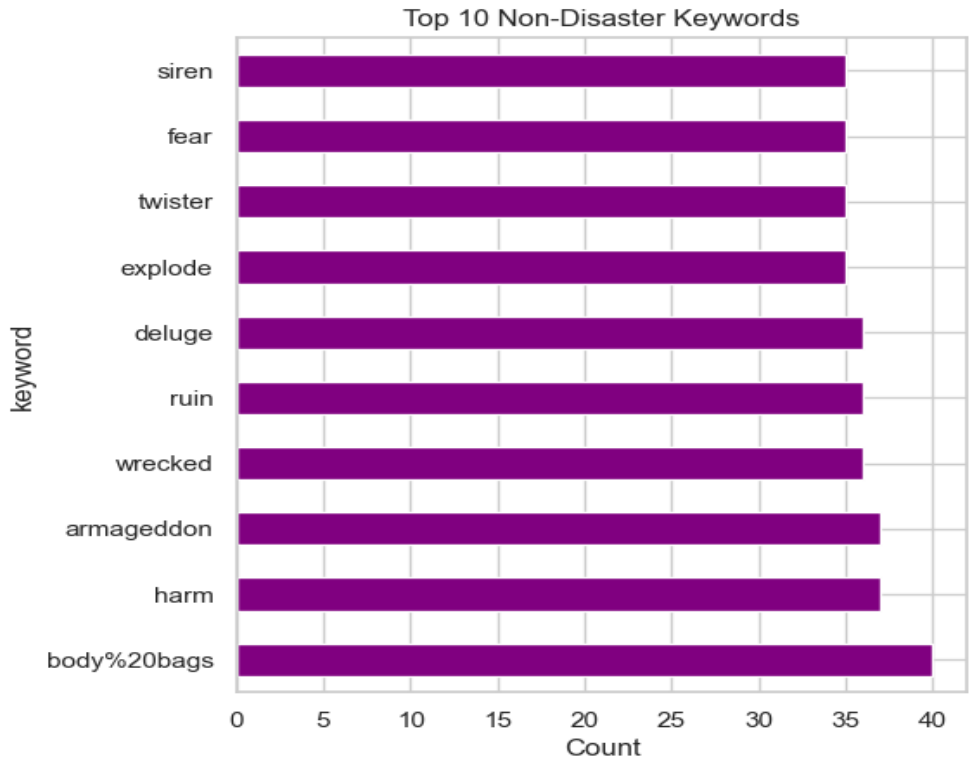

Figure 5: Top 10 Non-Disaster Keywords

Figure 4 and 5 above shows the top 10 disaster and non-disaster keywords

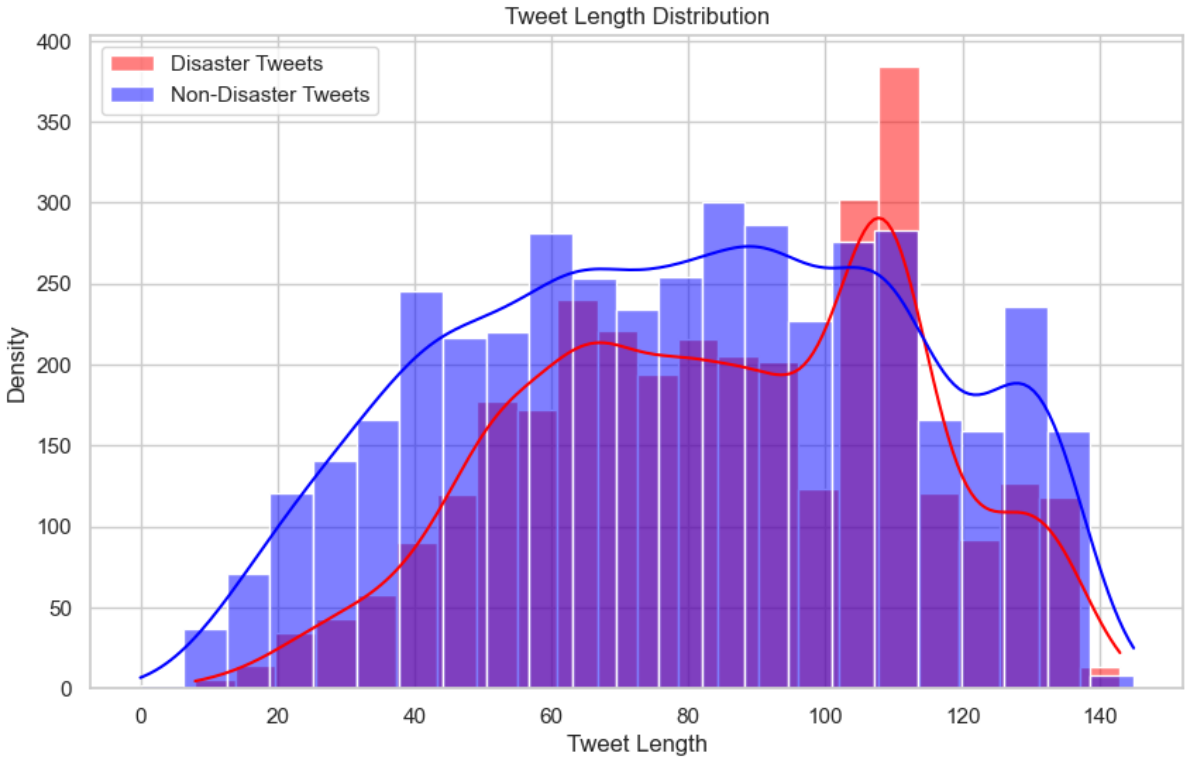

Figure 6: Tweet Length Distribution

Figure 6 above shows the length of tweets.

Word cloud was used to highlight common disaster terms such as flood, earthquake and hurricane within the tweets as shown in Figure 7

Figure 7: WordCloud of Tweets

8.0 TRADITIONAL MACHINE LEARNING METHODS

1. **Multinomial Naive Bayes (MNB)**- This model uses Bayes' Theorem and assumes feature independence.
Strength*:* Fast and works well with text data.
Weakness: It assumes all features are independent, which often isn't the case in real-world text data, and it can have trouble understanding more complex patterns.
Context: Not the best choice for this task

2. **Logistic Regression (LR)**- A linear model that estimates the probability a tweet is disaster-related based on input features.
Strength*:* Interpretable and performs well on binary classification tasks.
Weakness*:* Assumes linear relationships and struggles with complex patterns.
Context*:* Suitable because it's reliable, easy to tune, and performs well on clean, vectorized text.

3. **K-Nearest Neighbours (KNN)**- A non-parametric model that classifies tweets based on the most similar data points (neighbours).

Strength*:* Simple and makes no assumptions about data distribution.
Weakness: Slow with large datasets and high-dimensional data like TF-IDF vectors.
Context: KNN is inefficient due to large dataset of the task.

4. **Decision Tree (DT)**- A flowchart-like model that splits data based on feature values to make predictions.
Strength: Easy to visualize and handles non-linear patterns.
Weakness: Prone to overfitting, especially with noisy data like tweets.

5. **Random Forest (RF)**- An ensemble model combining multiple decision trees for better generalization.
Strength: High accuracy and robustness against overfitting.
Weakness: Slower and less interpretable than a single tree.
Context: Efficient for this task for its strong performance in capturing complex patterns across varied tweet structures.

In comparison with other traditional learning models Logistic Regression was chosen because it's a reliable supervised learning method that works well for binary classification problems like identifying disaster-related tweets and is easy to implement (Nusinovici et al., 2020). Random Forest was selected for its ability to handle complex and varied tweet content. It is an advanced version of a decision tree that instead of just one uses multiple trees , allowing it to make more accurate and reliable predictions (Shaik and Srinivasan, 2019)

9.0 DEEP LEARNING METHODS

1. **Recurrent Neural Network (RNN)** - RNNs process texts sequentially
   Strength: They're good at handling text that depends on word order.
   Weakness: They tend to forget earlier words in longer tweets and struggle with context.
   Context: Not ideal here since tweets vary in length and can lose meaning without full context.

2. **Long Short-Term Memory (LSTM**)- LSTMs improve on RNNs by remembering important words for longer.
   Strength: Great at picking up important patterns in text.
   Weakness: Slower to train and needs more data.
   Context: A solid choice for this task because disaster-related tweets often depend on understanding the full sentence.

3. **Bidirectional LSTM(BiLSTM)** - BiLSTM reads texts both forward and backward, giving it a fuller view of the sentence.
   Strength: Excellent at understanding word meaning from both directions.
   Weakness: Takes even more time and resources to train.
   Context: Very useful here, since tweets can be short and tricky — this model picks up on hidden meanings better.

4. **Gated Recurrent Unit (GRU)** - GRUs are like LSTMs but a bit simpler and faster.
   Strength: Less complex and quicker to train, while still capturing key patterns.
   Weakness: Might miss subtle patterns in more complex tweets.
   Context: A good middle ground — strong performance without heavy training time.

5. **Bidirectional Encoder Representations from Transformers (BERT)**- BERT is a powerful language model that understands the full context of words by looking at everything at once.
   Strength: Delivers top results on many text tasks and really understands nuance.
   Weakness: Very large and needs serious computing power.
   Context: Ideal for this kind of task if you've got the resources it understands even the subtlest tweet.

In comparison with other deep learning models BiLSTM was chosen because it achieves higher accuracy than LSTMs because they can learn from the input data in both directions, effectively using the information twice during training (Siami-Namini, Tavakoli and Namin, 2019). GRU was selected because it uses a gated structure to control how information is updated, which helps it achieve better performance in many tasks (Xu et al., 2023)

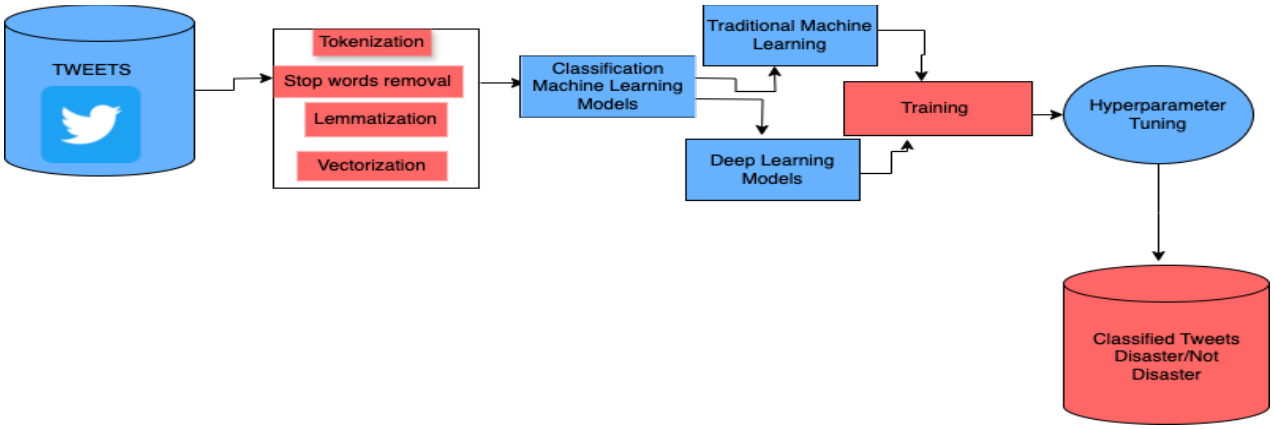

Figure 8: Pipeline Architecture

10.0 IMPLEMENTATION AND REFINEMENT

To build and evaluate the NLP pipeline, several key Python libraries were used:

- **NLTK** was used for text preprocessing tasks like tokenization (word_tokenize), stop word removal, and lemmatization with WordNetLemmatizer.
- **WordCloud** helped visualize the most common terms in the dataset.
- **Scikit-learn** provided core models including Logistic Regression, Random Forest Classifier, and Multinomial NB. It also supported feature extraction with Tfidf Vectorizer, scaling with StandardScaler
- **TensorFlow/Keras** was used to build deep learning models using layers like Embedding
- **Keras Tuner** was used for automated hyperparameter tuning of deep learning models.

Data augmentation: SMOTE technique was used to balance the dataset (Oversampling) as shown in Figure 3.

Hyperparameter Tuning

- **GridSearchCV** is used generally for optimizing hyperparameters in AI models (Kartini, Nugrahadi and Farmadi, 2021).

- **Attention-** This makes the model focuses on the most important and clinically relevant features, improving classification accuracy (Mohanty, Subudhi, Dash and Mohanty, 2024).

Other tuning techniques used includes Early stopping, Regularization and tweaking dropout rate, learning rate.

The architecture of all the models are presented in Table 2-6 below:

LOGISTIC REGRESSION MODEL 1 – GRIDSEARCH

Table 2: Logistic Regression Model 1 Architecture

Parameters (GridSearchCV)	Values Tested
Regularization(C)	0.1,0.5,0.1
Fit Intercept	True, False
Max Iteration	1000

LOGISTIC REGRESSION MODEL 2(GRIDSEARCH)-HYPERPARAMETER TUNING

Table 3: Logistic Regression Model 2 Architecture

Parameters (GridSearchCV)	Values Tested
Regularization(C)	0.1,0.5,1,10
Fit Intercept	True, False
Max Iteration	100,200,300
Penalty	l2,l1
Solver	Liblinear,Saga
Learning rate	1e-4,1e-3,1e-2

RANDOM FOREST MODEL HYPERPARAMETER TUNING

Table 4: Random Forest Architecture

Parameters (GridSearchCV)	Values Tested
Estimators	100,200
Maximum Depth	None,10,20
Minimum Samples Split	2,10
Minimum Samples Leaf	1,4
Maximum Features	Sqrt, log2
Class weight	None, balanced

BiLSTM ARCHITECTURE

Table 5: BiLSTM Architecture

Step	Parameters	Details
Model Building	**Dropout** **Recurrent dropout** **Activation Function** **BiLSTM Layer**	**0.3** **0.3** **Sigmoid** **128**
Model Compilation	**Optimizer** **Learning rate**	**Adam** **0.001**
Model Training	**Epochs** **Batch Size**:	**10** **64**
Hyperparameter Tuning	**Earlystopping** **Class weights** **Attention** **Adding more**	 **Balanced**

GRU ARCHITECTURE – INCORPORATING RANDOMSEARCH

Table 6: GRU Architecture

Step	Parameters	Values/Values Tested
Model Building	**Dropout** **Recurrent dropout** **Activation Function** **GRU Layer**	**0.1,0.5** **0.3** **Sigmoid** **32,128**
Model Compilation	**Optimizer** **Learning rate**	**Adam** **0.01,0.001,0.0001**

Model Training	**Epochs** **Batch Size**:	**10** **32**

11.0 EVALUATION

To measure how well the various models performed evaluation metrics like precision, accuracy, and F1-score were used. The results are presented in table 7 -13 below**;**

LOGISTIC REGRESSION MODEL 1

Class (Training)	Precision	Recall	F1-Score
0 (Not Disaster)	0.92	0.94	0.93
1(Disaster)	0.94	0.92	0.93
Accuracy			**0.93**
Macro Avg	0.93	0.93	0.93
Weighted Avg	0.93	0.93	0.93

Class (Validation)	Precision	Recall	F1-Score
0 (Not Disaster)	0.79	0.77	0.78
1(Disaster)	0.77	0.79	0.78
Accuracy			0.78
Macro Avg	0.78	0.78	0.78
Weighted Avg	0.78	0.78	0.78

Table 7: Training and Validation Classification report LR1 Model

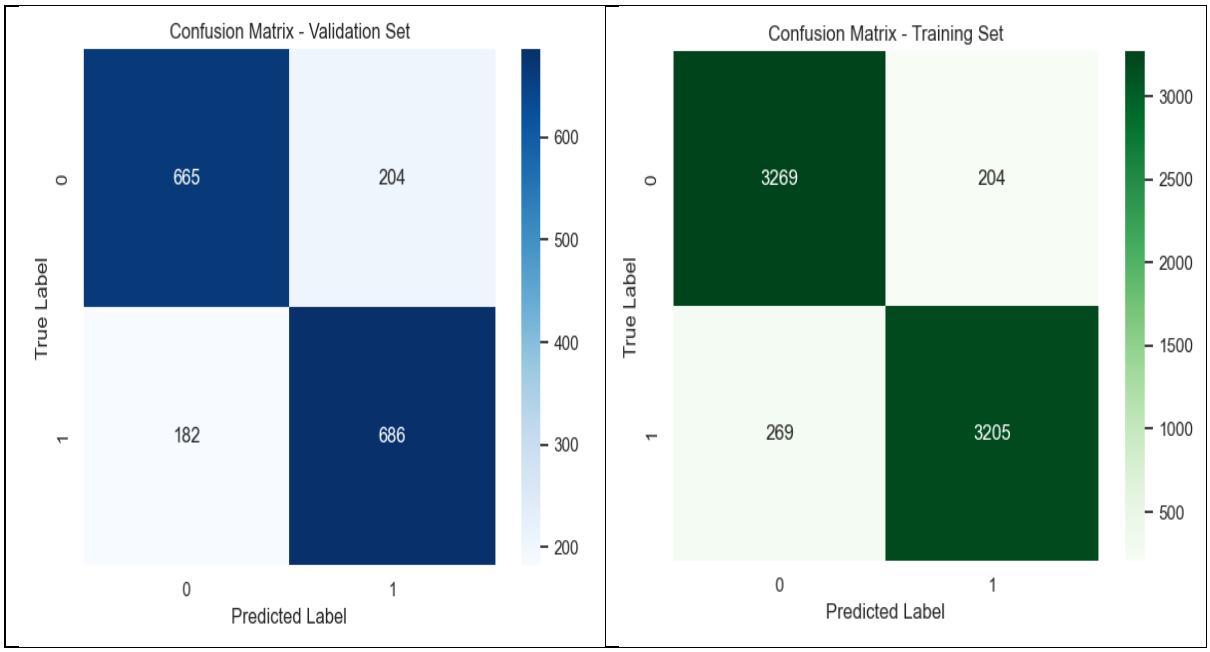

Figure 9: Confusion Matrix Training vs Validation Set LR1 Model

From the result in Table 7 above it can be observed that the model performs very well on the training set with **93% accuracy** and strong F1-scores for both classes (0.94 and 0.92) but shows a drop in precision and recall on the validation set, indicating overfitting.

LOGISTIC REGRESSION MODEL 2- ADDING MORE GRIDSEARCH PARAMETERS

Class (Training)	Precision	Recall	F1-Score
0 (Not Disaster)	0.89	0.93	0.91
1(Disaster)	0.92	0.89	0.90
Accuracy			0.91
Macro Avg	0.91	0.91	0.91
Weighted Avg	0.91	0.78	0.91

Class (Validation)	Precision	Recall	F1-Score
0 (Not Disaster)	0.79	0.77	0.80
1(Disaster)	0.81	0.79	0.80
Accuracy			0.80
Macro Avg	0.80	0.80	0.80
Weighted Avg	0.80	0.80	0.80

Table 8: Training and Validation Classification report LR2 Model

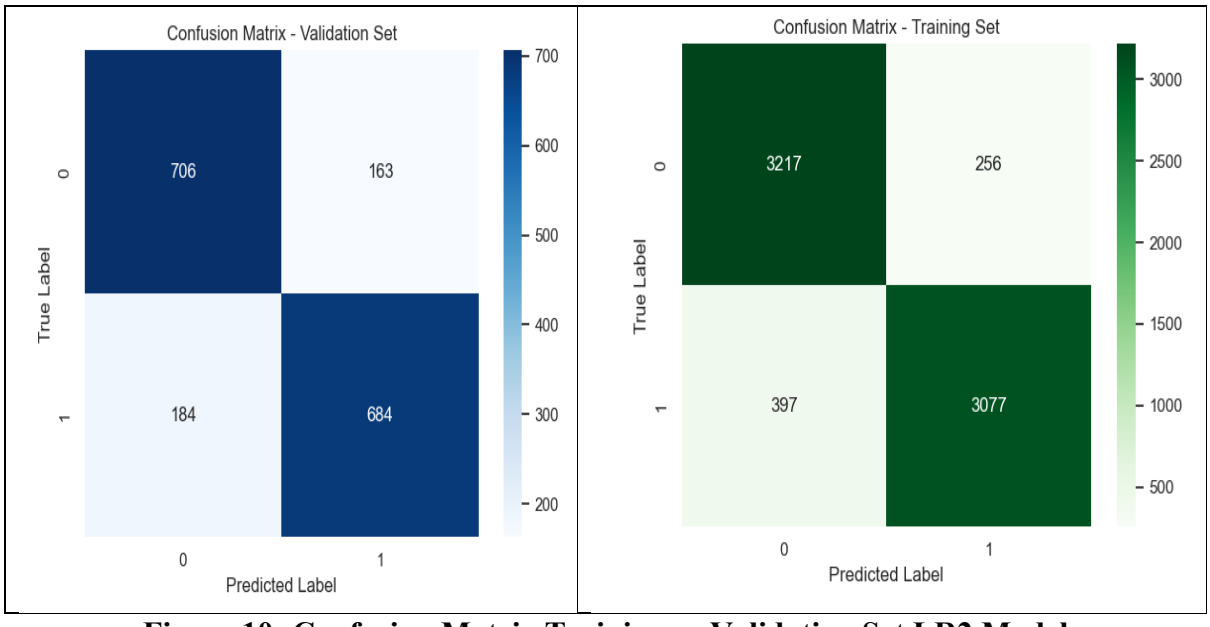

Figure 10: Confusion Matrix Training vs Validation Set LR2 Model

After tuning the Training accuracy is 91% with strong F1-scores (0.91 and 0.90), while validation accuracy is 80% with balanced F1-scores (both 0.80). The model generalizes better, but some overfitting remains as seen in Table 8.

RANDOM FOREST MODEL 1

Class	Precision	Recall	F1-Score
0 (Not Disaster)	0.78	0.86	0.82
1(Disaster)	0.85	0.75	0.80
Accuracy			0.81
Macro Avg	0.81	0.81	0.81
Weighted Avg	0.81	0.81	0.81

Table 9: Classification report RF1 Model

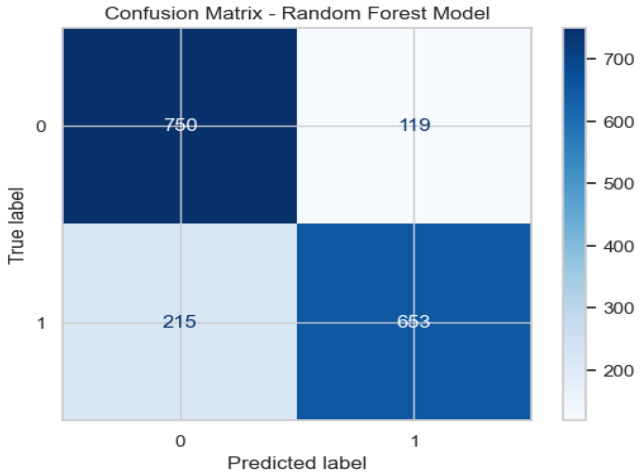

Figure 11: Confusion Matrix LR2 Model

The model achieves 81% accuracy, showing solid overall performance. It detects non-disaster tweets slightly better (F1-score: 0.82) than disaster tweets (F1-score: 0.80) as observed from the results in Table 9.

RANDOM FOREST MODEL 2 AFTER HYPERPARAMETER TUNING-GRIDSEARCH

Class	Precision	Recall	F1-Score
0 (Not Disaster)	0.78	0.89	0.83
1(Disaster)	0.87	0.75	0.81
Accuracy			0.82
Macro Avg	0.83	0.82	0.82
Weighted Avg	0.83	0.82	0.82

Table 10: Classification report RF2 Model

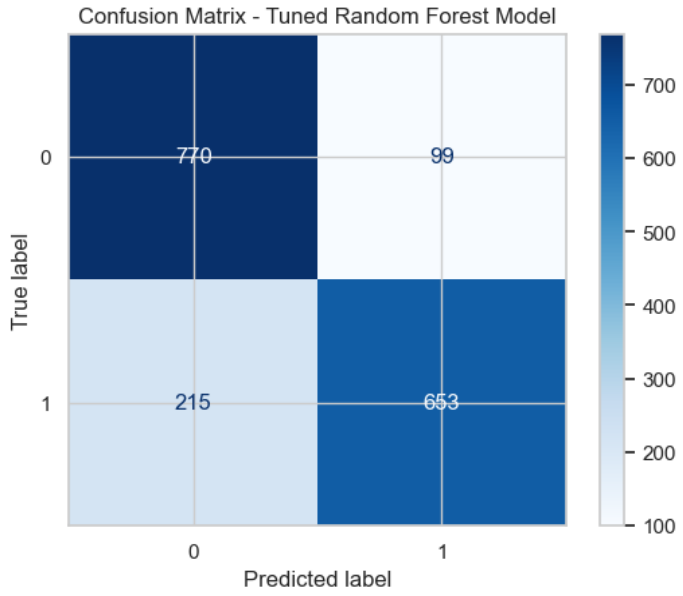

Figure 12: Confusion Matrix RF2 Model

```
Example 1:
Text: just happened a terrible car crash
Predicted Label: 1
--------------------------------------------------
Example 2:
Text: heard about earthquake is different cities stay safe everyone
Predicted Label: 1
--------------------------------------------------
Example 3:
Text: there is a forest fire at spot pond geese are fleeing across the street i cannot save them all
Predicted Label: 1
--------------------------------------------------
Example 4:
Text: apocalypse lighting spokane wildfires
Predicted Label: 0
--------------------------------------------------
Example 5:
Text: typhoon soudelor kills  in china and taiwan
Predicted Label: 1
--------------------------------------------------
Example 6:
Text: were shakingits an earthquake
Predicted Label: 1
--------------------------------------------------
Example 7:
Text: theyd probably still show more life than arsenal did yesterday eh eh
Predicted Label: 0
--------------------------------------------------
Example 8:
Text: hey how are you
Predicted Label: 0
--------------------------------------------------
Example 9:
Text: what a nice hat
Predicted Label: 0
--------------------------------------------------
Example 10:
Text: fuck off
Predicted Label: 0
--------------------------------------------------
```

Figure 13: Text sample from Test Set and Predicted Label by the Model

Result in Table 10 above the model has 82% accuracy and performs well. It's better at correctly identifying non-disaster tweets (higher recall), while it's more precise with disaster

tweets. The balanced F1-scores (0.83 for non-disaster, 0.81 for disaster) show solid performance. In Figure 13 the model classifies 8/10 samples accurately.

BiLSTM MODEL 1

Class	Precision	Recall	F1-Score
0 (Not Disaster)	0.73	0.77	0.75
1(Disaster)	0.67	0.63	0.65
Accuracy			0.71
Macro Avg	0.70	0.70	0.70
Weighted Avg	0.71	0.71	0.71

Table 11: Classification report BiLSTM Model 1

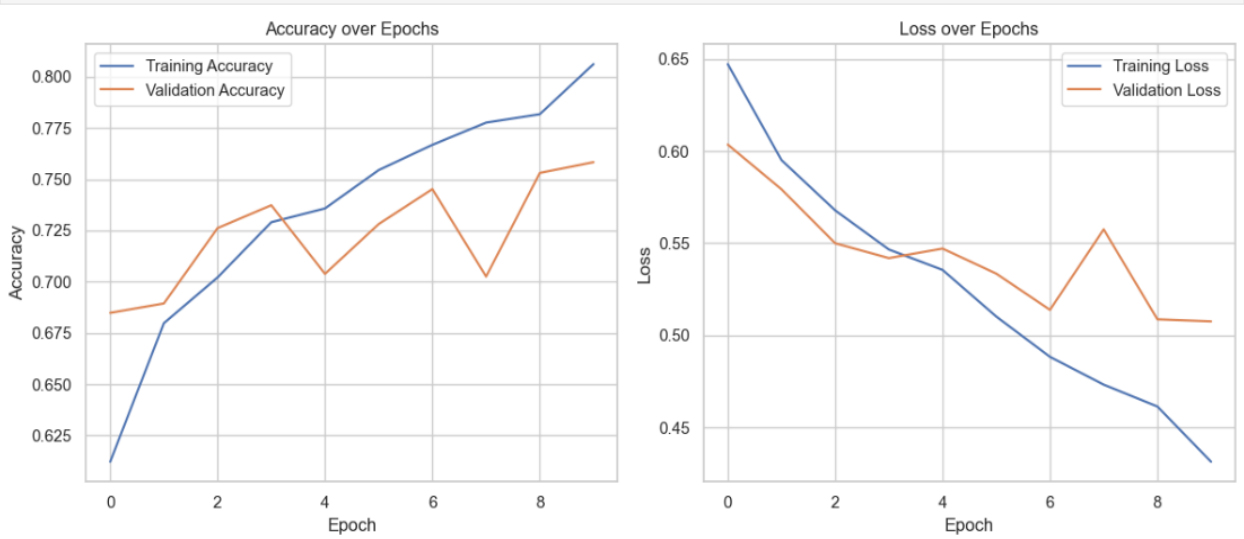

Figure 13: Training and Validation Accuracy vs Loss

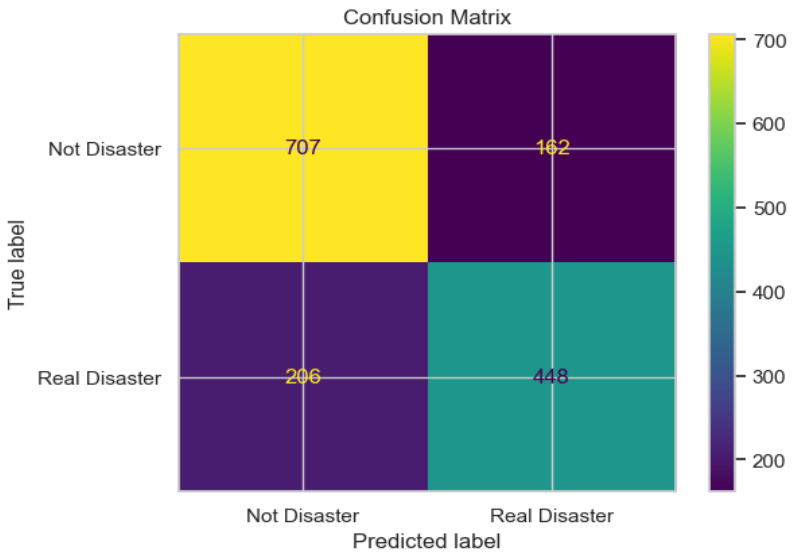

Figure 14: Confusion Matrix BiLSTM Model 1

The model has 71% accuracy, with better performance on non-disaster tweets (F1-score: 0.75) than disaster tweets (F1-score: 0.65). However, it struggles more with correctly identifying disaster tweets evident in Table 11.

BiLSTM MODEL 2 AFTER HYPERPARAMETER TUNING

Class	Precision	Recall	F1-Score
0 (Not Disaster)	0.77	0.81	0.79
1(Disaster)	0.73	0.69	0.71
Accuracy			0.76
Macro Avg	0.75	0.75	0.75
Weighted Avg	0.76	0.76	0.76

Table 12: Classification report BiLSTM Model 2

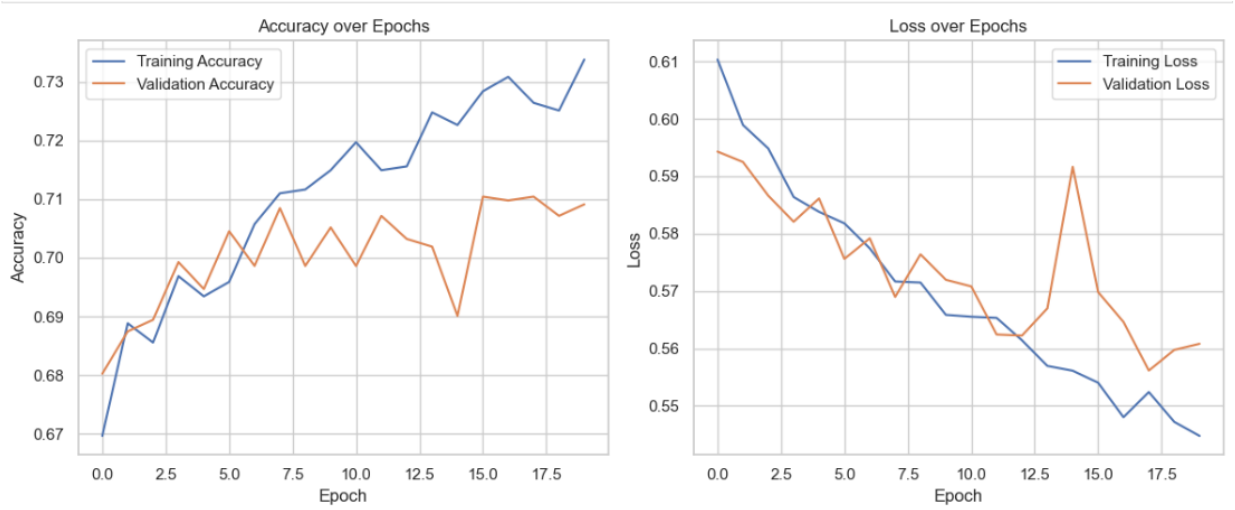

Figure 15: Training and Validation Accuracy vs Loss

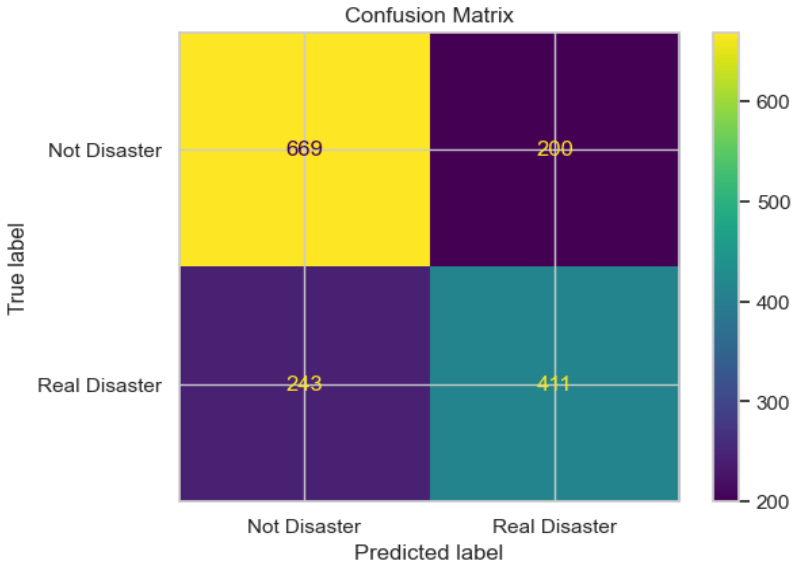

Figure 16: Confusion Matrix BiLSTM Model 2

In Table 12 The BiLSTM model achieved an 76% accuracy after tuning. It's a bit better at picking up non-disaster tweets but still struggles with the disaster tweets.

GRU MODEL USING RANDOM SEARCH

Class	Precision	Recall	F1-Score
0 (Not Disaster)	0.75	0.81	0.78
1(Disaster)	0.71	0.65	0.68
Accuracy			0.74
Macro Avg	0.73	0.73	0.73
Weighted Avg	0.74	0.74	0.74

Table 13: Classification report GRU

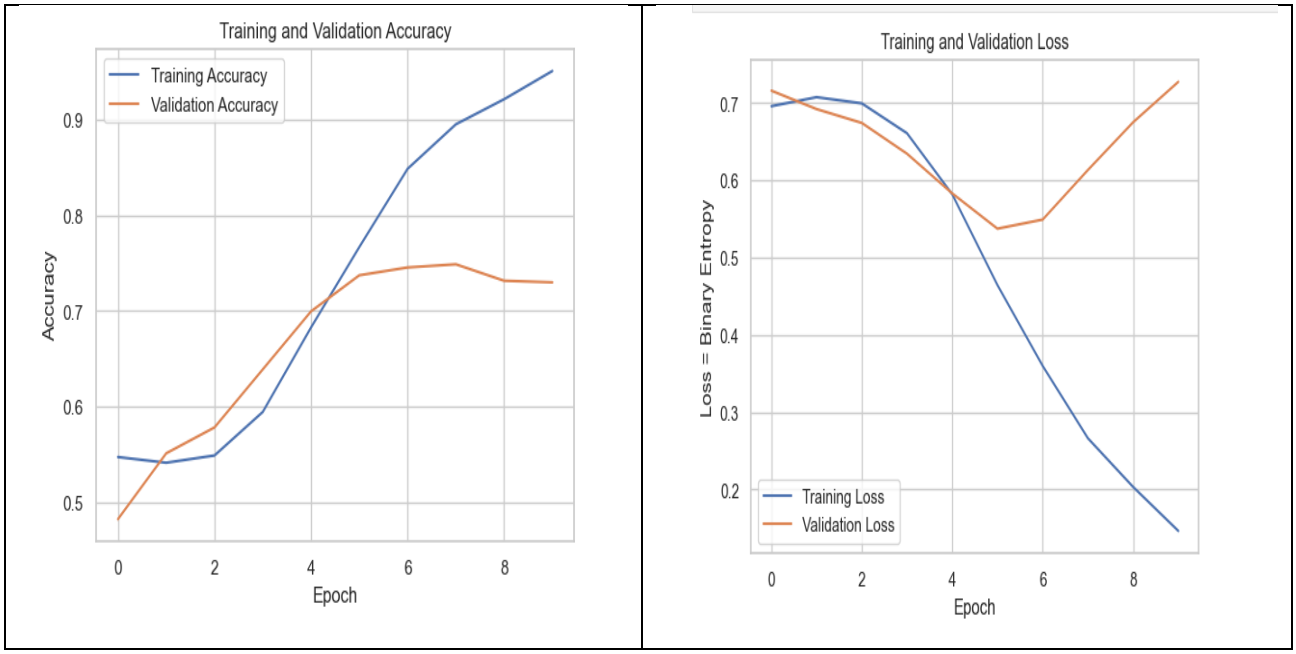

Figure 17: Training and Validation Accuracy vs Loss

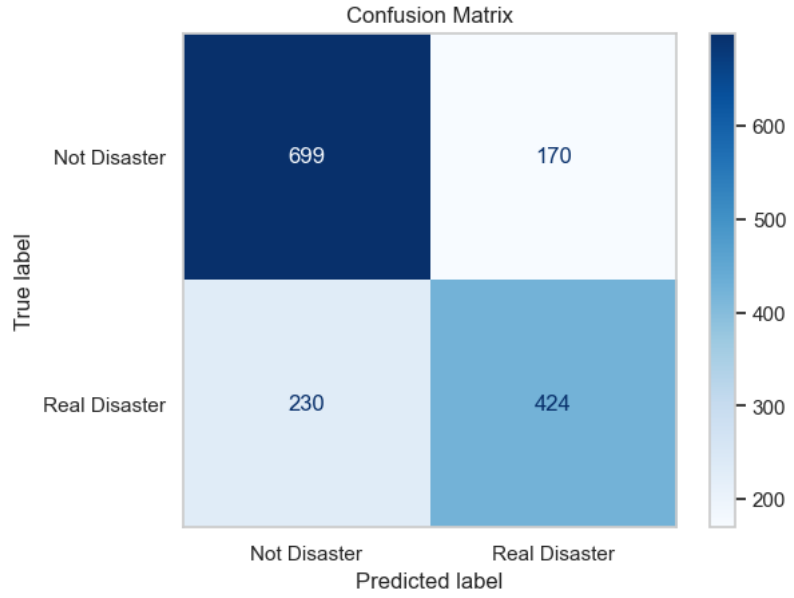

Figure 18: Confusion Matrix BiLSTM Model 2

The model reaches 74% accuracy and does slightly better with non-disaster tweets (F1: 0.78) than disaster ones (F1: 0.68) as evident in Table 13.

CONCLUSION

Random Forest outperformed all the other models. It achieved an accuracy of 82% and strong F1-scores for both non-disaster (0.83) and disaster (0.81) tweets, making it the best at

handling both classes well. Logistic Regression performed well on the training data, but the drop in validation performance this suggests it overfitted the training set. Tuning helped improve generalization, but some overfitting remained.

The BiLSTM and GRU models performed, especially after tuning, but they still struggled with disaster tweets. This was likely due to class imbalance, which made it harder for the models to learn enough from the less frequent disaster examples. While the deep learning models showed promise in picking up patterns in sequences, they didn't outperform the simpler, tree-based Random Forest in this case.

COMPARING MLA, APA, CHICAGO, HARVARD, AND VANCOUVER STYLES

Style	In-text Citation	Reference List Format	Field of Use	Justification
MLA	(Author Last Name Page Number)	Last Name, First Name. *Title*. Publisher, Year.	Humanities (e.g., literature)	Not considered
APA	(Author Last Name, Year)	Last Name, Initials. (Year). *Title*. Publisher.	Social sciences, psychology	Not considered
Chicago	(Author Last Name Year) or footnotes	Last Name, First Name. *Title*. City: Publisher, Year.	History, some humanities	Not considered
Harvard	(Author Last Name, Year)	Last Name, Initials. Year. *Title*. Publisher.	General academic use	Used this style because it is most suitable for this report
Vancouver	[Number]	Number. Author Initials Last Name. *Title*. Publisher; Year.	Medical and scientific writing	Not considered

REFERENCES

Alam, F., Ofli, F., Imran, M. and Aupetit, M., 2018. A twitter tale of three hurricanes: Harvey, irma, and maria. *arXiv preprint arXiv:1805.05144*.

Alfawareh, H.M. & Jusoh, S. (2011). Resolving ambiguous entity through context knowledge and fuzzy approach. International Journal on Computer Science and Engineering (IJCSE), 3 (1), 410 – 422.

Basit, M., Alam, B., Fatima, Z. and Shaikh, S., 2023, December. Natural disaster tweets classification using multimodal data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 7584-7594).

Imran, M., Castillo, C., Lucas, J., Meier, P. and Vieweg, S., 2014, April. AIDR: Artificial intelligence for disaster response. In *Proceedings of the 23rd international conference on world wide web* (pp. 159-162).

Internet Live Stats. (2024). *Twitter Usage Statistics*. Available at: https://www.internetlivestats.com/twitter-statistics/ [Accessed 26 Apr. 2025].

Iparraguirre-Villanueva, O., Melgarejo-Graciano, M., Castro-Leon, G., Olaya-Cotera, S., John, R.A., Epifanía-Huerta, A., Cabanillas-Carbonell, M. and Zapata-Paulini, J., 2023. Classification of tweets related to natural disasters using machine learning algorithms.

Kartini, D., Nugrahadi, D.T. and Farmadi, A., 2021, September. Hyperparameter tuning using GridsearchCV on the comparison of the activation function of the ELM method to the classification of pneumonia in toddlers. In *2021 4th international conference of computer and informatics engineering (IC2IE)* (pp. 390-395). IEEE.

Kumar, A., Singh, J.P. and Saumya, S., 2019, November. A comparative analysis of machine learning techniques for disaster-related tweet classification. In *2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129)* (pp. 222-227). IEEE.

Madichetty, S., Muthukumarasamy, S. and Jayadev, P., 2021. Multi-modal classification of Twitter data during disasters for humanitarian response. *Journal of ambient intelligence and humanized computing*, *12*, pp.10223-10237.
Mohanty, B.C., Subudhi, P.K., Dash, R. and Mohanty, B., 2024. Feature-enhanced deep learning technique with soft attention for MRI-based brain tumor classification. *International Journal of Information Technology*, *16*(3), pp.1617-1626.

Nair, M.R., Ramya, G.R. and Sivakumar, P.B., 2017. Usage and analysis of Twitter during 2015 Chennai flood towards disaster management. *Procedia computer science*, *115*, pp.350-358.

Ningsih, A.K. and Hadiana, A.I., 2021, March. Disaster tweets classification in disaster response using bidirectional encoder representations from transformer (BERT). In *IOP Conference Series: Materials Science and Engineering* (Vol. 1115, No. 1, p. 012032). IOP Publishing.

Nusinovici, S., Tham, Y.C., Yan, M.Y.C., Ting, D.S.W., Li, J., Sabanayagam, C., Wong, T.Y. and Cheng, C.Y., 2020. Logistic regression was as good as machine learning for predicting major chronic diseases. *Journal of clinical epidemiology*, *122*, pp.56-69.

Shaik, A.B. and Srinivasan, S., 2019. A brief survey on random forest ensembles in classification model. In *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2018, Volume 2* (pp. 253-260). Springer Singapore.

Siami-Namini, S., Tavakoli, N. and Namin, A.S., 2019, December. The performance of LSTM and BiLSTM in forecasting time series. In *2019 IEEE International conference on big data (Big Data)* (pp. 3285-3292). IEEE.

Udanor, C., Aneke, S. and Ogbuokiri, B.O., 2016. Determining social media impact on the politics of developing countries using social network analytics. *Program*, *50*(4), pp.481-507.

UNDRR (2020) *The human cost of disasters: An overview of the last 20 years (2000–2019)*. United Nations Office for Disaster Risk Reduction. Available at: https://www.undrr.org/publication/human-cost-disasters-overview-last-20-years-2000-2019 (Accessed: 25 April 2025).

van den Bulk, L.M., Bouzembrak, Y., Gavai, A., Liu, N., van den Heuvel, L.J. and Marvin, H.J., 2022. Automatic classification of literature in systematic reviews on food safety using machine learning. *Current Research in Food Science*, *5*, pp.84-95

Xu, Z., Lv, Z., Chu, B. and Li, J., 2023. Fast autoregressive tensor decomposition for online real-time traffic flow prediction. *Knowledge-Based Systems*, *282*, p.111125.

Zhao, X. and Sun, Y., 2022. Amazon fine food reviews with BERT model. *Procedia Computer Science*, *208*, pp.401-406.