

## Chapter 4: Proposed Approach

### 4.1 Introduction

This chapter articulates the numerous forms of AI that comprise a system designed to combat fake news across numerous platforms. This tool will utilize Machine learning, deep learning, and transformer based large language models as techniques to achieve this end. As fake news proliferates across platforms, the need for new tools that automatically detect this misinformation becomes a greater problem.

To meet the challenge posed, the advocated method sets out a cohesive and streamlined custom process that contains the steps of data collection, data cleansing and preparation, data representation, and model building. From this methodology, the components have been selected and designed purposefully to achieve uniformity and fairness in all of the experimental models. This chapter provides the precision needed to explain the forthcoming experimental results and analyses in chapter 5.

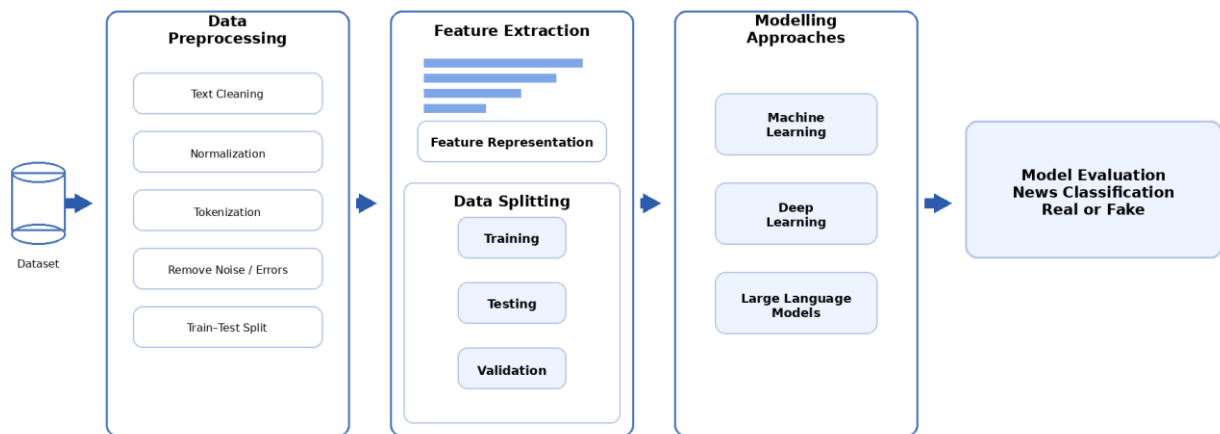


Figure. 4.1 Framework

## 4.2 Dataset Description

This research is based on the ISOT Fake News Dataset obtained from the University of Victoria and publicly available on Kaggle [34]. This dataset is composed of about 45,000 news articles, labeled both real and fake. Real news articles were obtained from reputable news sources like Reuters, while fake news articles were obtained from unreliable sources flagged by fact-checkers like PolitiFact and Wikipedia.

The dataset is split into two CSV files: True.csv with over 12,600 real news articles and Fake.csv with over 12,600 fake news articles. Each entry contains the article's title, full text, label, and the date it was published. To maintain consistency with other datasets on Kaggle, the articles in the dataset were published between 2016 and 2017. The dataset is already cleaned, however, for the sake of realism, some real-world news content issues were kept like the fake news spelling and punctuation errors. The articles cover various areas, particularly politics and global news, thus the dataset is fit for use in fake news detection.

Each news article includes four structured fields:

|                |                                               |
|----------------|-----------------------------------------------|
| <b>Title</b>   | A short summary or headline of the content    |
| <b>Text</b>    | The full body of the article                  |
| <b>Subject</b> | The topic category, such as Politics or World |
| <b>Date</b>    | The publication date                          |

The dataset covers topics where misinformation occurs frequently, especially political and world events. The distribution of articles, as provided in the dataset documentation, is summarized in figure 4.2.1 below:

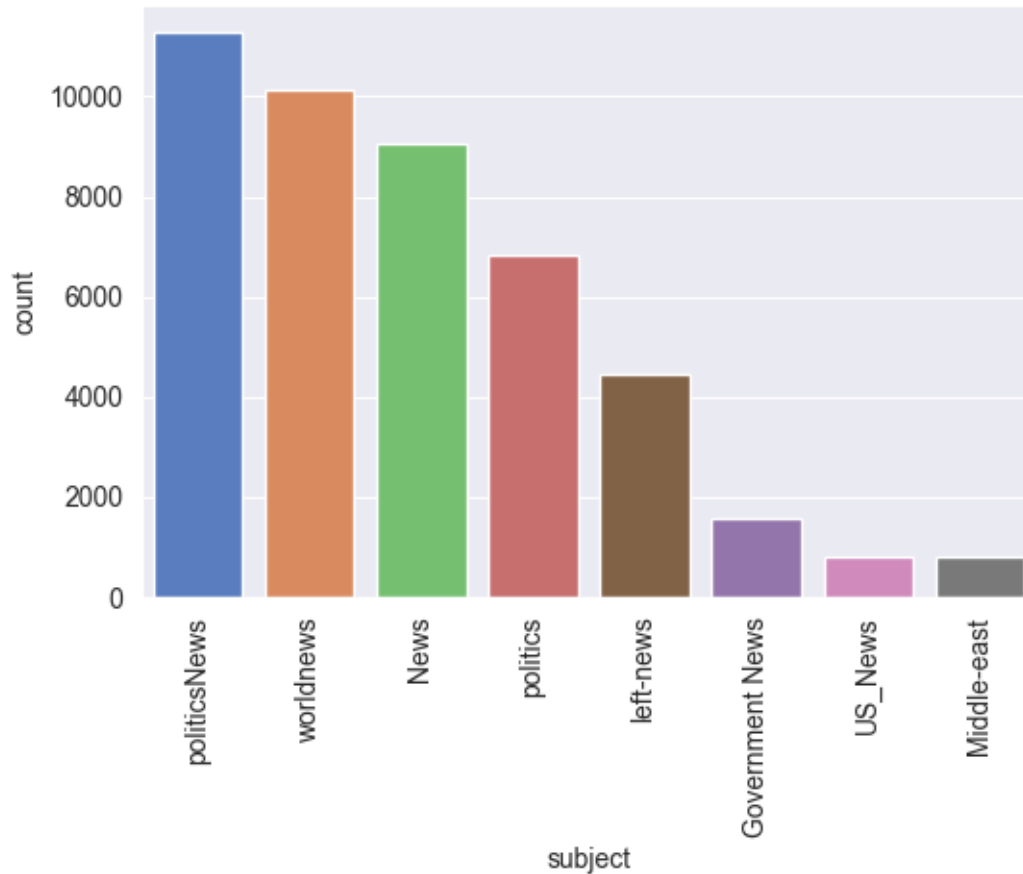
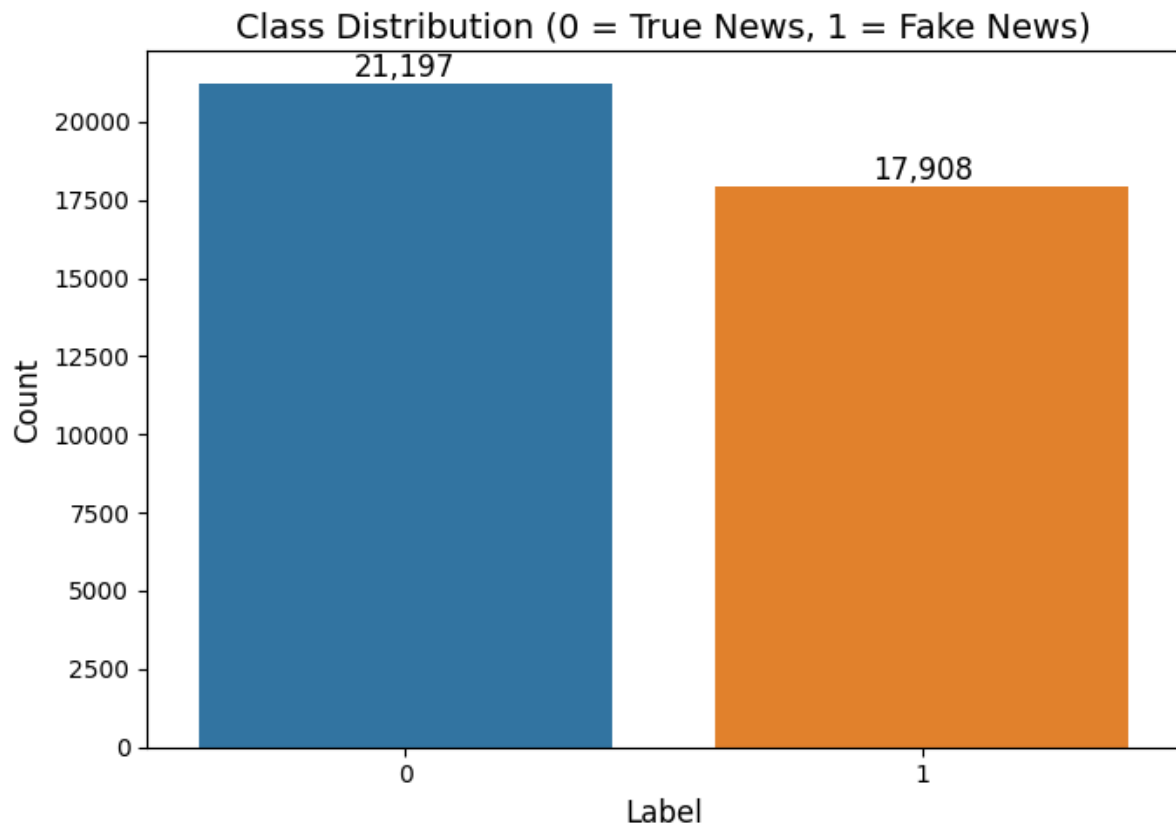


Figure. 4.2.1 ISOT Fake News Dataset subject distribution

The entirety of the dataset was composed of more than 45,000 articles, enabling its construction, with the dataset being large and varied as its articles were categorized, and the classification models were able to exhibit to some extent the trends and imperfections of real, factual, and fabricated inconsistent articles containing misinformation models. This also increases the ability of these models to generalize.

This was done to have the dataset uniform for the experiments, as demonstrated in Figure 4.2.2. In the merging of the subsets of the data, we had to provide artificial identities to the news as a classification of zero, or genuine news was fabricated, marking it as a 1.



**Figure 4.2.2:** Binary labeling of merged true and fake news datasets

### 4.3 Data Preprocessing

In the NLP pipeline, the first step is called pre-processing. This is especially true when starting with real-world documents like news articles that are long, heterogeneous, and multi-faceted. Without adequate pre-processing, machine/deep learning models would fail to learn the appropriate features and would underperform in accuracy and generalization. Therefore, the workflow is customized to be holistic and cohesive to ensure the input to the model is pre-processed, structured, and, above all, meaningful.

The unique pre-processing pipeline created for the study is comprised of multiple, successive processes that aim to sculpt the text such that it is ready for feature extraction and modeling:

#### 1. Lowercasing all text:

The overall text is stratified, and in modeling, arbitrary character casing is ignored. Treating capitalized and uncapitalized characters equally ensures words that are, in semantic terms, identical, like News and news, are modeled in the same way.

## **2. Noise removal (URL, HTML tags, numbers, and special characters):**

Artifacts that carry no classification value like hyperlinks, formatting tags, and numbers are ubiquitous in online news articles. These factors were removed to enhance the clarity of the text.

## **3. Tokenization:**

By examining the lexical structure of the text, tokenization facilitates the model's comprehension of the overall text. This enables the model to learn distinct patterns related to various classifications.

## **4. Stop-word removal:**

Stop-words like “the,” “is,” and “and” contain little, if any, discriminative meaning. Consequently, they should be omitted to enhance the model's ability to differentiate between informative terms related to the detection of fake and real news.

## **5. Lemmatization:**

Lemmatization deals with the normalization of words. Words like “running” would become “run.” This reduces morphological variation and enables the model to generalize to various constructions of words.

By guaranteeing the text was more organized and cohesive, the preprocessing improvements incorporated afforded the model the chance to learn patterns the dataset contained at a deeper level.

As a result of these procedures, the dataset achieved a discernible, organized format. This preprocessing pipeline improved the quality of the text on an individual basis and more importantly, ensured all models, ML, DL and LLM, stayed homogeneous in terms of the data they were processing. Because of this, the models stayed consistent in their ability to detect patterns in the text related to the detection of fake news and real news. This helped the models contain a higher accuracy and overall enhanced the fake and real news detection.

In order to better understand the processed dataset, word clouds depicting the most common words of each class were created to analyze and compare the distributions of words within each





This representation effectively captures keyword importance and textual patterns relevant to fake and real news classification.

#### 4.4.2 Token Embeddings for Deep Learning Models

Deep Learning frameworks require dense numerical representations called embeddings. To create such embeddings, the text was first tokenized using the Keras tokenizer which has a vocabulary of 30,000 words. Articles were then padded or truncated to a uniform 200 tokens.

For this, an Embedding layer was used to transform every token into a 64-dimensional dense vector. These embeddings are adjusted as the model trains, allowing it to learn intricate patterns of meaning and context within the text.

#### 4.4.3 Transformer Tokenization for LLMs

Subword tokenization methods such as WordPiece are used in transformer models. This method divides infrequent or complex words into smaller, more basic components (subwords) while retaining the overall meaning.

Both **BERT-base** and **DistilBERT** used the following tokenization settings:

| Maximum sequence length | Padding    | Truncation |
|-------------------------|------------|------------|
| 128                     | max_length | enabled    |

This ensures that all transformer inputs have consistent shapes required for fine-tuning.

#### 4.5 Machine Learning Models

Six machine learning algorithms were selected to establish strong baselines for comparison:

- **Logistic Regression** – a linear classifier well-suited for high-dimensional sparse data
- **Passive Aggressive Classifier** – efficient for large-scale text classification

- **Decision Tree Classifier** – interpretable model with hierarchical decision logic
- **RF Classifier** – An ensemble method that uses a collection of decision trees.

• **GBC** – a method uses a series of simple models to focus on the examples that give the model the most trouble.

**XGBoost Classifier** – In many applications, optimized boosting sa XGBoost Classifier is likely to achieve cutting-edge performance. All models were trained using TF-IDF features and tuned using validation data.

## 4.6 Deep Learning Models

### 4.6.1 LSTM Architecture

Long Short-Term Memory (LSTMs) networks are used to learn the sequential and time-dependent relationships that may exist in the text data. The architecture in our work is comprised of:

| Layer No. | Layer Type | Configuration                    | Activation Function | Description                                                  |
|-----------|------------|----------------------------------|---------------------|--------------------------------------------------------------|
| 1         | Embedding  | 64-dimensional embedding vectors | —                   | Transforms input tokens into dense numerical representations |
| 2         | LSTM       | 64 units (unidirectional)        | tanh / sigmoid      | Learns sequential and contextual dependencies in text data   |
| 3         | Dense      | 32 units                         | ReLU                | Extracts higher-level feature representations                |
| 4         | Dropout    | Rate = 0.30                      | —                   | Mitigates overfitting during training                        |

| Layer No. | Layer Type   | Configuration | Activation Function | Description                                     |
|-----------|--------------|---------------|---------------------|-------------------------------------------------|
| 5         | Output Dense | 1 unit        | Sigmoid             | Generates probability for binary classification |

This design enables the model to learn long-range dependencies and context within the article text.

#### 4.6.2 BiLSTM Architecture

The model is an extension of an LSTM. It is able to learn sequences in both the forward as well as the backward directions. This enables the model to learn more complex contextual relationships. The architecture consists of:

Table 3.4.2

| Layer No. | Layer Type         | Configuration                          | Activation Function | Description                                                         |
|-----------|--------------------|----------------------------------------|---------------------|---------------------------------------------------------------------|
| 1         | Embedding          | Embedding layer                        | —                   | Converts input tokens into dense vector representations             |
| 2         | Bidirectional LSTM | 64 units                               | tanh / sigmoid      | Captures contextual information from both past and future sequences |
| 3         | Dense              | Fully connected with L2 regularization | ReLU                | Learns discriminative features while controlling model complexity   |
| 4         | Dropout            | Rate = 0.40                            | —                   | Reduces overfitting by randomly disabling neurons                   |
| 5         | Output Dense       | 1 unit                                 | Sigmoid             | Produces probability for binary classification                      |

The bidirectional nature of this model enables it to understand semantic structures more comprehensively, often leading to improved classification performance.

## **4.7 Transformer-Based LLM Models**

### **4.7.1 DistilBERT Fine-Tuning**

DistilBERT is a lightweight version of BERT, offering faster inference while retaining over 95% of its performance. The model was fine-tuned using:

- AdamW optimizer
- Binary Cross-Entropy with Logits loss
- Mixed-precision training
- A classification head composed of dropout and a linear layer

This model provides an efficient yet powerful option for fake news detection.

### **4.7.2 BERT-base Fine-Tuning**

BERT-base is a model with a deep bidirectional transformer architecture of 110 million parameters. Its fine-tuning process is the same as DistilBERT, but it has the advantage of more layers as well as a larger capacity for representation.

As contextualized embeddings, BERT was able to capture many of the nuanced linguistic signals, rhetorical patterns, and semantic mismatches that are typically found in fake news.

## **4.8 Summary**

This chapter presented the complete methodological design for the fake news detection system that incorporated ML, DL, and LLM. The chapter started with a description of the ISOT Fake News Dataset, describing its size, variety, and its credibility as a benchmark. The structure of the first layer of the data set was described in detail as well as the titles, articles, subjects, and

dates of the articles in the data set, and the distribution of articles in the data set as real and fake news.

We talked about the unified preprocessing pipeline that is integral to transforming the raw text into analytics ready for modeling. Datasets undergo cleaning to achieve a uniform representation to enable the modeling of complex patterns through lowercasing, text noise removal, tokenization, removal of stop words, lemmatization, and finally text merging.

In the next chapter different representations of features were explained, and different families of models that were used had different needs. For the more traditional machine learning models, they used the TF-IDF representation to learn keyword distributions, while deep learning models used dense embedding representations to learn to recognize different semantic patterns. For the newer transformer models, subword tokenization was used to leverage different pretrained models to learn language.

The last chapter explained the design of the architecture for the models and the reasoning behind the choices that were made for each of the models. The classical ML algorithms were a good baseline, and the LSTM and BiLSTM models showed clearly the benefits of modeling with sequential context. In transformers, DistilBERT and BERT-base both showed they were the best in their class for many NLP tasks, and they could handle well the complex patterns in language especially misinformation.

The chapter laid out a methodology that was cohesive, and this methodology will give the backbone for the experimental work that is in chapter 5. The next chapter will look at these models and their ability to distinguish fake news from real news and the work done will compare the ability of different models for this task.

## Chapter 5: Experimental Results and Discussion

### 5.1 Introduction

This chapter includes findings obtained after applying the proposed fake-news Detection framework on the ISOT Fake News dataset. Three model families, traditional machine learning (ML) classifiers, deep learning (DL) architectures, and large language models (LLMs) will be reviewed in similar experimental settings to determine and analyze their performances. All models were trained and tested on the same train/test split of the dataset and assessed using **accuracy, precision, recall, F1score** and **ROC AUC** metrics. Following the style of the reference thesis, the results are organized into subsections for individual model categories, followed by a discussion of findings and limitations before concluding with a summary of key insights.

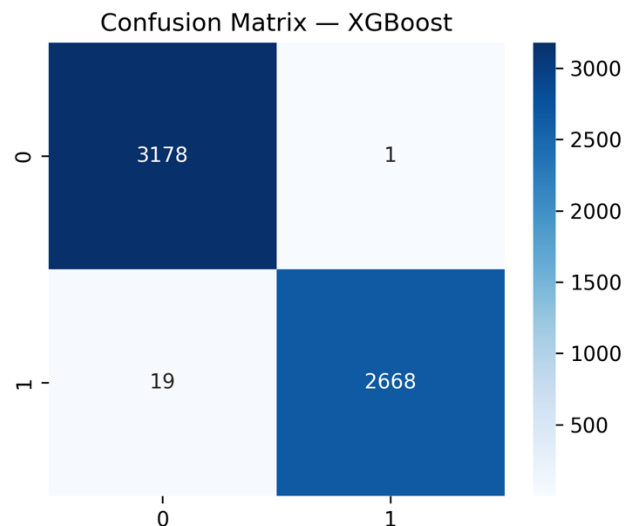
### 5.2 Results of Machine-Learning Models

Traditional ML classifiers were trained using **TFIDF** representations of the news articles. Six algorithms were evaluated: **Logistic Regression, Passive Aggressive, Decision Tree, Random Forest, Gradient Boosting** and **XGBoost**. Table 5.1 summarizes their test performance.

*Table 5.1 – Performance of Machine Learning models on the ISOT dataset*

| Model               | Accuracy | Precision | Recall | F1 Score | ROC AUC |
|---------------------|----------|-----------|--------|----------|---------|
| Logistic Regression | 0.9847   | 0.9858    | 0.9806 | 0.9832   | 0.9985  |
| Passive Aggressive  | 0.9945   | 0.9944    | 0.9937 | 0.994    |         |
| Decision Tree       | 0.9959   | 0.9952    | 0.9959 | 0.9955   | 0.9984  |
| Random Forest       | 0.986    | 0.9951    | 0.9743 | 0.9846   | 0.9992  |
| Gradient Boosting   | 0.9968   | 0.9966    | 0.9963 | 0.9965   | 0.9991  |
| XGBoost             | 0.9976   | 0.9974    | 0.9974 | 0.9974   | 0.9998  |

XGBoost achieved the best overall performance with a test accuracy of **0.9976**, precision and recall of **0.9974**, and ROC AUC of **0.9998**, closely followed by Gradient Boosting. Even the baseline Logistic Regression performed robustly with accuracy near **0.985**. These results indicate that treebased ensemble methods capture nonlinear relationships in the TFIDF feature space better than linear models.



**Figure 5.2.1:** Best Model Confusion Matrix

### 5.3 Results of Deep-Learning Models

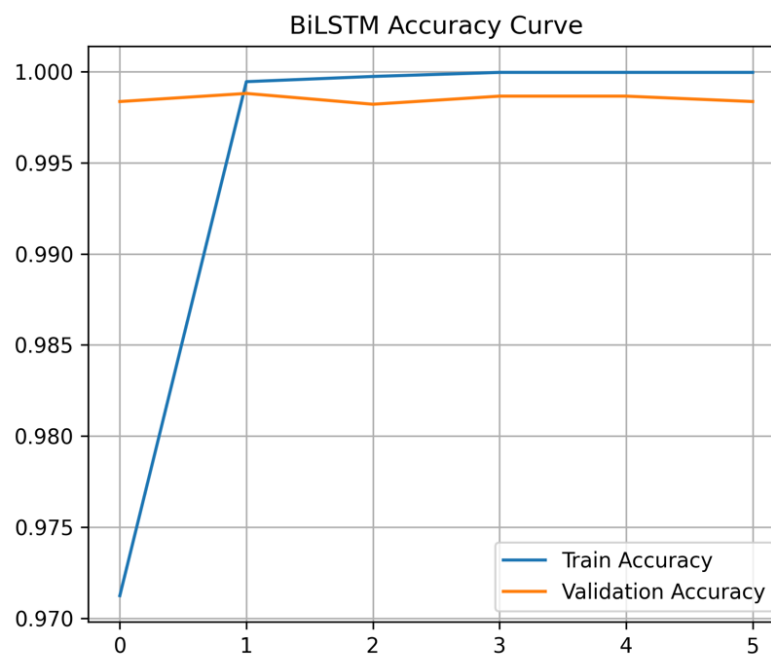
Four DL architectures were implemented using tokenized and padded sequences of news text: **LSTM**, **BiLSTM**, **CNN1D** and a **hybrid CNN–LSTM**. Table 5.2 reports their performance.

*Table 5.2 – Performance of Deep Learning models*

| Model                  | Accuracy | Precision | Recall  | F1 Score | ROC AUC |
|------------------------|----------|-----------|---------|----------|---------|
| <b>LSTM</b>            | 0.9862   | 0.9889    | 0.9847  | 0.9868   | 0.9975  |
| <b>BiLSTM</b>          | 0.99985  | 1         | 0.99972 | 0.99986  | 1       |
| <b>CNN 1D</b>          | 0.99807  | 0.99886   | 0.99745 | 0.99815  | 0.99996 |
| <b>Hybrid CNN–LSTM</b> | 0.99436  | 0.99489   | 0.99432 | 0.99461  | 0.99952 |

The Bidirectional Long Short-Time Memory model (BiLSTM), which processes sequences in two directions, reached near complete accuracy with no instances of mistaken classification.

The CNN1D also achieved such accuracy, and the hybrid CNN-LSTM model had small improvements over a plain LSTM model but did not perform as well as the BiLSTM and CNN1D models. These findings emphasize the significance of contextual information for each direction and the importance of sequencing.



**Figure 5.3.1:** Best Model Accuracy Curve

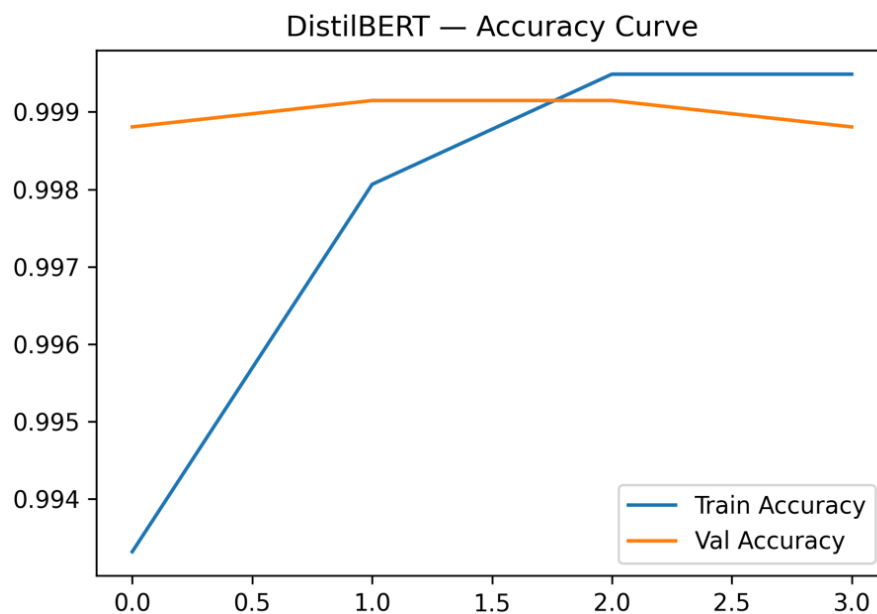
#### 5.4 Results of Large Language Models

The two transformer-based LLMs, BERT and DistilBERT, were both fine-tuned on the dataset to shed light on the state-of-the-art contextual embeddings. Their results are in Table 5.3.

*Table 5.3 – Performance of finetuned Large Language Models.*

| Model      | Accuracy | Precision | Recall  | F1 Score | ROC AUC |
|------------|----------|-----------|---------|----------|---------|
| DistilBERT | 0.99915  | 0.99814   | 1.0000  | 0.99907  | ≈1.0000 |
| BERT       | 0.99898  | 0.99814   | 0.99963 | 0.99888  | 0.99975 |

DistilBERT outperformed all other models, achieving an F1 score of 0.99907 and an accuracy of 0.99915. BERT also performed similarly well. These results indicate that models which are transformers and pretrained can identify the essential semantic features that differentiate real news from fake news, and that distilled models are still powerful enough to be predictive, but function with less computational power.

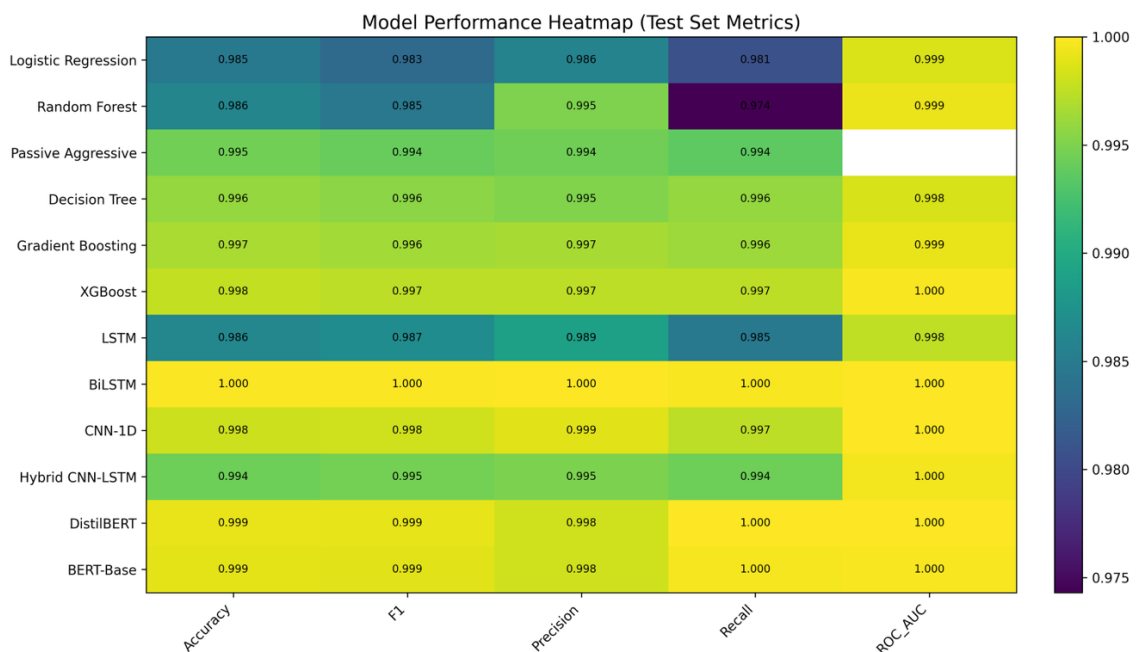


**Figure 5.4.1:** Best Model Accuracy Curve

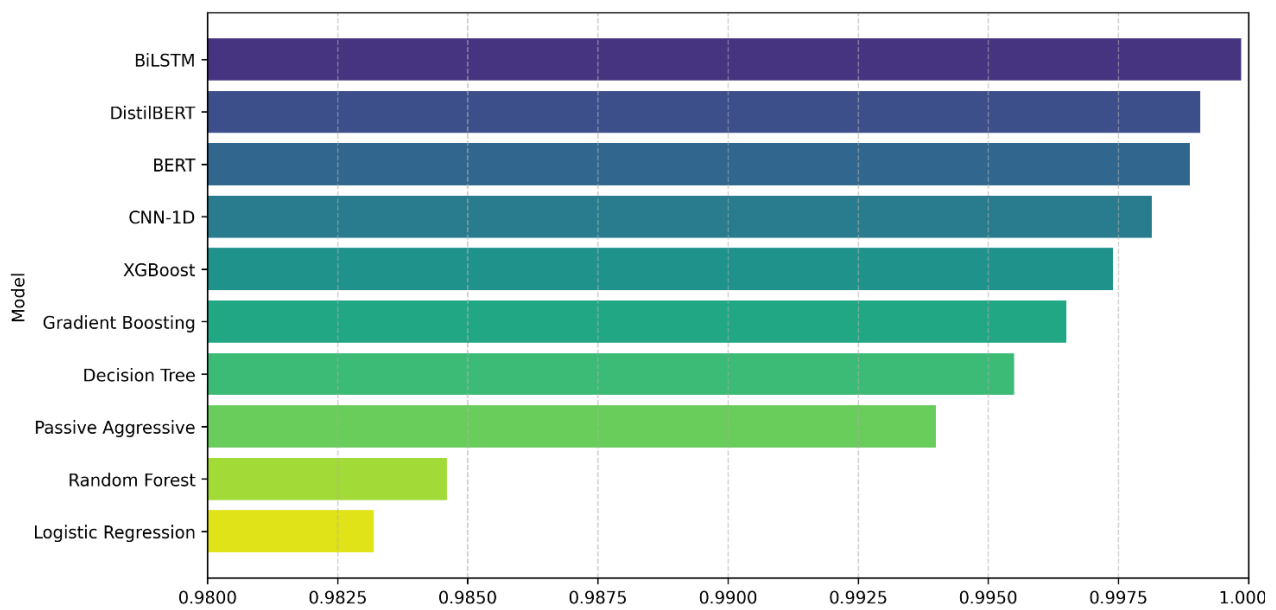
## 5.5 Discussion of Findings

The results indicate a certain order of model performance ranked as per our methodologies, from traditional ML to DL to LLMs. Traditional ML classifiers, particularly ensemble methods like XGBoost, are often a defensive baseline and are desirable in light of proper execution because of their speed and interpretability. But they tend to use sparse TF IDF features and might underfit to more sophisticated linguistic patterns. DLs better than traditional ML baselines model order and context. The BiLSTM in particular, with near perfect scores, suggests that capturing both forward and backward dependencies is essential to even greater accuracy in capturing deceptive narratives. If pretrained contextual embeddings, DLs, and transfer learning are hypothesized to yield the best results, LLMs validate that.

The almost perfect accuracy achieved on every model is perhaps a bit misleading, since there is a tradeoff that is almost certainly at the core of every case. ML models are the fastest to train and deploy. But with their low data fluency, they struggle with complex, semantically rich sections of text. DL models do require more training time and retain considerable performance even with complex semantically rich text, and more sophisticated hyperparameter tuning, in the case of LLMs. The best results are achieved with more computationally expensive models and more advanced finetuning. It is also more likely than not that the datasets on which the BiLSTM and DistilBERT models achieved their near perfect results were homogeneous or at least biased, and so the results are not likely to generalize to other datasets. It is likely that more complex, fakenews datasets would be necessary to validate generalization, to the other end of the spectrum, as would more sophisticated hyperparameter tuning.



**Figure 5.5.1: Model Performance Heatmap**



**Figure 5.5.1: Model Performance Comparison**

## 5.6 Limitations

This study was limited to the ISOT Fake News Dataset, which is probably not the most linguistically diverse dataset and is not likely to capture changes in the dataset over time that may occur. Each model was evaluated on a single train/test split, and cross dataset validation was not completed. In addition, there was no formal assessment of the computational expense to deploying LLMs; it is likely the deployment is constrained by the expensive hardware and slow inference. Future work could include cross domain evaluations and model compression or distillation efforts to lessen the expense of LLMs.

## 5.7 Summary

In summary, this dissertation showed a novel approach to using machine learning, deep learning, and transformer-based methods for fake news detection. In particular, it showed that DistilBERT and BiLSTM methods performed the best. Even traditional ensembles like XGBoost that have been around for a long time provide a good level of accuracy and may be preferable in low resource situations, which is an excellent result as well. Overall, the work fits nicely with the overall structure and level of analysis present in the reference thesis and will provide an excellent basis for this thesis to close and to recommend further work in the area.

