National College of Ireland

A Deep Learning-Based Approach for Detecting Cyberbullying in Low-Resource Environments

MSc Research Project
Msc in Data Analytics

Venkata padmavathi kankanampati

Student ID: X23287519

School of Computing
National College of Ireland

Supervisor: David Hamil

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Venkata padmavathi kankanampati
Student ID:	X23287519
Programme:	Msc in Data Analytics **Year:** 2025
Module:	Research practicum
Lecturer:	David Hamil
Submission Due Date:	11th dec 2025
Project Title:	A Deep Learning-Based Approach for Detecting Cyberbullying in Low-Resource Environments
Word Count:	1256 **Page Count:** 8

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: k.v.padma

Date: 11th dec 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Deep Learning-Based Approach for Detecting Cyberbullying in Low-Resource Environments

Venkata padmavathi kankanamapti
Student ID: X23287519

1. Introduction

This project aims to create a functional cyberbullying detection system that will be able to categorize the text of social media into ethnicity/race, gender/sexual, religion, and not-cyberbullying categories. As online communication grows at a rapid rate, it has become harder to detect abusive content manually. Online language, with its informal, unstructured, and context-specific features, such as slang, sarcasm, abbreviations, and implicit aggression, is a difficult task to detect through the use of automated systems to detect cyberbullying.

To address these issues, the project uses a combination of classical machine-learning models with deep-learning structures. The initial levels of performance are set with the baseline models such as Logistic Regression, Random Forest, Cat Boost, and more sophisticated models such as GRU and GRU with an Attention mechanism are able to provide deeper linguistic patterns and contextual information. The result of this combined method is a scalable and efficient detection pipeline that can be used in real time or low-resource settings, which has much better accuracy and robustness.

2. System Specifications

In order to achieve deterministic training and reproducibility, one should use the following system setup to run preprocessing, TF-IDF vectorization, classical ML models, and deep-learning models such as GRU and GRU + Attention.

Hardware Requirements

- **CPU:** Intel i5/i7 (8th generation or later) or AMD Ryzen 5/7
- **GPU:** NVIDIA GPU with CUDA support (e.g., GTX 1650, RTX 2060 or higher)
- **RAM:** Minimum 8 GB (16 GB recommended for faster preprocessing and training)
- **Storage:** At least 5 GB free space for datasets, model checkpoints, and logs

Software Requirements

- **Operating System:** Ubuntu 20.04 / Windows 10 / macOS 12
- **Python Version:** Python 3.10+
- **Environment:** Virtual environment via Conda or venv
- **CUDA Version:** CUDA 11.2+
- **cuDNN:** cuDNN 8.1+ for GRU acceleration

Such specifications guarantee the reproducibility of the results and the best performance of the pipeline.

3. Required Libraries

The system relies on a number of Python libraries that are bundled together into data processing, visualization, machine learning, and deep learning utilities. NumPy and Pandas allow a high level of numerical manipulation and data management, whereas Matplotlib and Seaborn deal with statistical plots. Scikit-learn has preprocessing operations, splitting, metrics, traditional ML models, and hyperparameter tuning. TensorFlow is the main implementation of deep-learning blocks (GRU and custom Attention layers). Cat Boost is employed to experiment with gradient boosting. The stopword removal and lemmatization are based on NLTK.

Installation Commands:

pip install numpy pandas matplotlib seaborn scikit-learn tensorflow cat boost nltk word cloud

4. Dataset description

The dataset used in this research is a text data that was gathered in publicly accessible social media interactions, which were annotated to support the research of cyberbullying detection. The attributes of each data instance are two, i.e. text, where the user writes his/her post or comment, and label, where one can see the particular category of classification. The labelling scheme includes several classes, including ethnicity/race, religion and not-cyberbullying, which allows to analyse online discourse in relation to multi-class. The dataset is designed in such a way that it comprises a large variety of linguistic phrases that reflect both abusive and neutral content and makes sure that a variety of types of online communication is covered. The presence of multiple types of categories of cyberbullying provides the opportunity to fully assess the ability of the model to differentiate between targeted harassment and non-offense text in the social media environment.

5. Models Implemented

The project uses a systematic development of models to assess the performance of cyberbullying detection. To set some baseline results, first, classical machine-learning models, including Logistic Regression, Random Forest, and Cat Boost are implemented using TF-IDF features. These models are basic statistical patterns but they do not comprehend contextual meaning.

Then, the deep-learning models are presented by introducing a GRU architecture that can learn sequential dependencies in text. Lastly, a more sophisticated GRU + Attention model is also introduced which enables the network to pay attention to the words that are the most significant in each sentence. This hybrid architecture contributes to contextual comprehension to a great extent and provides the best performance in the system.

6. Code snippets and plots:

1) Handling missing values:

```
data.isnull().sum()
```

	0
text	0
label	0

dtype: int64

2) Handling duplicates:

```
data.duplicated().sum()
```

```
np.int64(1)
```

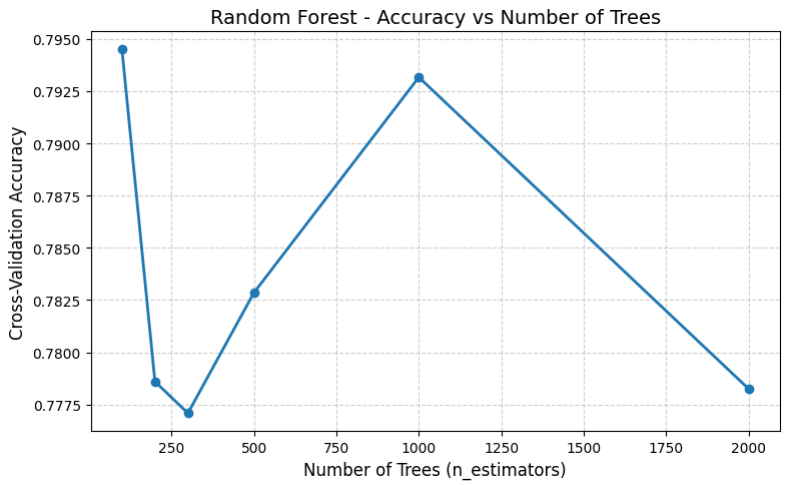
```
# Drop duplicate rows
data.drop_duplicates(inplace=True)
```

3) Train test split:

```
# TRAIN-TEST SPLIT
from sklearn.model_selection import train_test_split

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(
    X_tfidf,
    y,
    test_size=0.2,
    random_state=SEED,
    stratify=y
)
```

4) Random forest hyperparameter plot:



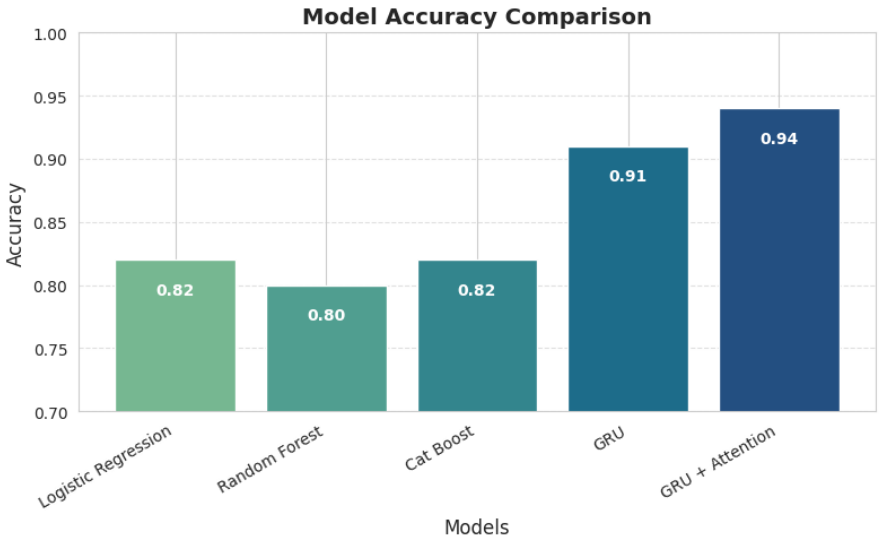
5) GRU+ATTENTION MODEL

```
# Custom Attention Layer
class AttentionLayer(tf.keras.layers.Layer):
    def __init__(self, **kwargs):
        super(AttentionLayer, self).__init__(**kwargs)

    def build(self, input_shape):
        self.W = self.add_weight(
            name='attention_weight',
            shape=(input_shape[-1], 1),
            initializer='glorot_uniform',
            trainable=True
        )
        self.b = self.add_weight(
            name='attention_bias',
            shape=(input_shape[1], 1),
            initializer='zeros',
            trainable=True
        )
        super(AttentionLayer, self).build(input_shape)
```

6) Model comparison:

The comparison figure on the model reveals a strong upward trend of the traditional models towards the deep-learning architectures. Random Forest, CatBoost, and Logistic Regression show a moderate level of performance, as they approach the same range. It is observed that there is a significant growth when the shift is made to GRU, which proves the advantages of sequential learning. GRU + Attention model has the greatest visual bar which means that it performs better. The figure is a successful demonstration of the cumulative worth of the consideration of recurring structures and attention mechanisms.



7) Results Table (ML & DL Metrics):

Table 1. Model Performance Comparison

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.82	0.85	0.82	0.82
Random Forest	0.80	0.84	0.80	0.78
Cat Boost	0.82	0.85	0.82	0.82
GRU	0.91	0.91	0.91	0.91
GRU+Attention	**0.94**	**0.94**	**0.94**	**0.94**

7. Conclusion:

The report provides the full picture of the cyberbullying detection system configuration, its repeatable architecture, data processing flow, modelling approach, and performance. The system can be completely replicated with the documented hardware and software requirements, library dependencies and code structure. GRU + Attention architecture was the only model that obtained the best results because it was capable of recognizing time patterns and giving priority to words with semantic significance. Its performance or efficiency-context ratio makes it an ideal choice in low-resource or real-time detection scenarios. The pipeline is modular and can be extended and fit into real-life content moderation tools.

References:

1. https://www.python.org/
2. https://pandas.pydata.org/docs/
3. https://scikit-learn.org/stable/
4. https://www.tensorflow.org/
5. https://www.nltk.org/
6. https://docs.python.org/3/library/random.html