National College of Ireland

Detection of AI-Generated Fake Reviews: A Comparative Analysis of Transformer and Recurrent Architectures Across Language Model Generations

MSc Research Project
MSC Data Analytics

SAI KALYAN BADE
Student ID: X23388111

School of Computing
National College of Ireland

Supervisor: Arghir Nicolae Moldovan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: SAI KALYAN BADE

Student ID: X23388111

Programme: MSc Data Analytics **Year:** 2026

Module: MSc Research Practicum

Supervisor: Arghir Nicolae Moldovan

Submission Due Date: 28/01/2026

Project Title: Detection of AI-Generated Fake Reviews: A Comparative Analysis of Transformer and Recurrent Architectures Across Language Model Generations

Word Count: 1568 **Page Count:** 9

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: SAI KALYAN BADE

Date: 28/01/2026

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):	

Detection of AI-Generated Fake Reviews: A Comparative Analysis of Transformer and Recurrent Architectures Across Language Model Generations

Sai Kalyan Bade
X23388111

1: HARDWARE REQUIREMENTS

1.1 Minimum Requirements (Google Colab Free Tier)

- GPU: NVIDIA Tesla T4 (15 GB VRAM)
- RAM: 12.7 GB System RAM
- Disk: 78 GB available storage
- Runtime: Python 3.10+

1.2 Recommended Requirements (Colab Pro / Local)

- GPU: NVIDIA T4 / V100 / A100 (16+ GB VRAM)
- RAM: 16+ GB System RAM
- Disk: 50+ GB SSD storage
- CUDA: 12.1 or higher

1.3 Tested Configuration

- Platform: Google Colab Pro
- GPU Type: NVIDIA Tesla T4
- CUDA Version: 12.6
- cuDNN Version: 9.10.2.21
- Python Version: 3.12

2: SOFTWARE DEPENDENCIES

2.1 Core Libraries (install via pip)

torch==2.7.0
transformers>=4.35.0
pandas>=2.0.0
numpy>=1.24.0
scikit-learn>=1.3.0
torchmetrics>=1.0.0
nltk>=3.8.0
tqdm>=4.65.0

2.2 Additional Libraries
matplotlib>=3.7.0
seaborn>=0.12.0
2.3 NLTK Data Downloads (run in Python)
python
import nltk
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
2.4 Installation Commands (Colab Cell)
bash
!pip install torch transformers pandas numpy scikit-learn torchmetrics
!pip install nltk tqdm matplotlib seaborn

3: DATASETS

3.1 Dataset Files Location
All datasets stored in: noteBooks_dataset/datasets/

File	Description
fake reviews dataset.csv	DS1 - GPT-2
dataset_qwen_72b.csv	DS2.1 - Qwen
dataset_llama3_8b.csv	DS2.2 - LLaMA
dataset_qwen_dsr1.csv	DS2.3 - DeepSeek
final_combined_dataset.csv	DS2.4 - Combined

3.2 Dataset 1: GPT-2 Generated Reviews (DS1)
Repository: https://github.com/joolsa/FakeReviews/tree/main
Citation:
Salminen, J., Kandpal, C., Kamel, A. M., Jung, S., & Jansen, B. J. (2021). Creating and detecting fake reviews of online products. Journal of Retailing and Consumer Services, 64, 102771. https://doi.org/10.1016/j.jretconser.2021.102771

- **File:** fake reviews dataset.csv
- **Total Samples:** ~40,433
- **Categories:** 10 Amazon product categories
  - Books, Clothing_Shoes_and_Jewelry, Electronics
  - Home_and_Kitchen, Kindle_Store, Movies_and_TV
  - Pet_Supplies, Sports_and_Outdoors, Toys_and_Games
  - Video_Games
- **Classes:** CG (Computer Generated), OR (Original Real)
- **Balance:** 20,238 CG, 20,216 OR

3.3 Dataset 2: FraudSquad - Modern LLMs (DS2)
Repository: https://anonymous.4open.science/r/FraudSquad-5389/README.md
Citation:
Liu, X., Xu, R., Jia, X., Liao, J., Sun, J., Huang, L., & Xu, W. (2025). Detecting LLM-Generated spam reviews by integrating language model embeddings and graph neural network. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2510.01801
Original Reviews Source: Amazon Reviews 2023 (https://amazon-reviews-2023.github.io)
DS2.1 Qwen-72B Subset

- File: dataset_qwen_72b.csv
- Total: 4,990 samples (2,495 CG + 2,495 OR)

DS2.2 LLaMA-3-8B Subset

- File: dataset_llama3_8b.csv
- Total: 4,978 samples (2,489 CG + 2,489 OR)

DS2.3 DeepSeek-R1-Qwen Subset

- File: dataset_qwen_dsr1.csv
- Total: 5,000 samples (2,500 CG + 2,500 OR)

DS2.4 Combined Dataset

- File: final_combined_dataset.csv
- Total: 14,968 samples (7,484 CG + 7,484 OR)

Categories (7 total):

- All Beauty, Appliances, Baby Products
- Health and Personal Care, Musical Instruments
- Software, Video Games

3.4 Data Split Configuration

Parameter	Value
Train/Test Ratio	80/20
Split Method	Stratified sampling
Random Seed	2021

3.5 Dataset Column Names and Structure

3.5.1 Dataset 1 (GPT-2) - fake reviews dataset.csv

● Text Column: review_body
● Label Column: label (or generated_label)
● Label Mapping: CG=1 (Computer Generated), OR=0 (Original Real)
● Notes: Remove rows with missing values in review_body or label columns. Apply label mapping before train/test split.

3.5.2 Dataset 2 (FraudSquad) - All variants (Qwen, LLaMA, DeepSeek, Combined)

● Text Column: text (or review_text)
● Label Column: label
● Label Mapping: CG=1 (Computer Generated), OR=0 (Original Real)
● Notes: Ensure consistent column names across all DS2 variants. Handle any class imbalance before splitting.

4: EXPERIMENT NOTEBOOKS

4.1 Notebook Location

All notebooks stored in Code Artifacts /

4.2 Dataset 1 (GPT-2) Notebooks

Location: Code Artifacts / Dataset-1- GPT-2 /

Model	Notebook File
Attention-GRU	Attention_GRU-Model ipynb
RoBERTa	Roberta Model.ipynb
DeBERTa	Deberta Model.ipynb
XLNet	XLNet Model ipynb

4.3 Dataset 2 Combined Notebooks
Location: Code Artifacts / Dataset-2-FraudSquad/Combined/

Model	Notebook File
Attention-GRU	Attention_GRU Model.ipynb
RoBERTa	Roberta Model.ipynb
DeBERTa	Deberta Model.ipynb
XLNet	XLNet Model.ipynb

4.4 Dataset 2 Individual - Qwen-72B Notebooks
Location: Code Artifacts / Dataset-2-FraudSquad /Individual/ Qwen-72B /

Model	Notebook File
Attention-GRU	Attention_GRU Model.ipynb
RoBERTa	Roberta Model.ipynb
DeBERTa	Deberta Model .ipynb
XLNet	XLNet Model.ipynb

4.5 Dataset 2 Individual - LLaMA-3-8B Notebooks
Location: Code Artifacts / Dataset-2-FraudSquad /Individual / LLaMA-3-8B /

Model	Notebook File
Attention-GRU	Attention_GRU Model.ipynb
RoBERTa	Roberta Model.ipynb
DeBERTa	Deberta Model .ipynb
XLNet	XLNet Model.ipynb

4.6 Dataset 2 Individual - DeepSeek-R1-Qwen Notebooks
Location: : Code Artifacts / Dataset-2-FraudSquad /Individual / DeepSeek-R1-Qwen/

Model	Notebook File
Attention-GRU	Attention_GRU Model.ipynb
RoBERTa	Roberta Model.ipynb
DeBERTa	Deberta Model. ipynb
XLNet	XLNet Model.ipynb

4.7 Total Experiments Summary

Dataset	GRU	RoBERTa	DeBERTa	XLNet	Total
DS1 (GPT-2)	1	1	1	1	4
DS2.4 (Combined)	1	1	1	1	4
DS2.1 (Qwen)	1	1	1	1	4
DS2.2 (LLaMA)	1	1	1	1	4
DS2.3 (DeepSeek)	1	1	1	1	4
TOTAL	**5**	**5**	**5**	**5**	**20**

5: MODEL CONFIGURATIONS

5.1 ATTENTION-GRU MODEL

Architecture:

Parameter	Value
Embedding Dimension	64
GRU Hidden Dimension	256
GRU Layers	2
Bidirectional	True
Dropout	0.2
Attention	Additive attention mechanism
Output	Fully connected + Sigmoid

Training Parameters:

Parameter	Value
Batch Size	128 (train and validation)
Learning Rate	3e-4 (0.0003)
Optimizer	Adam
Loss Function	Binary Cross-Entropy
Epochs	10
Gradient Clipping	Max norm 5
Max Sequence Length	1,000 tokens

Text Preprocessing:

- Lowercase conversion
- Contraction expansion
- URL, username, hashtag removal
- HTML tag removal
- Stopword removal (NLTK English)
- Lemmatization (WordNet)
- Vocabulary size: 1,000 tokens
- Padding: Left-side

Estimated Training Time: ~15-20 minutes per dataset (T4 GPU) **Model Size:** ~2 million parameters

5.2 RoBERTa MODEL

Base Model: roberta-base (Hugging Face) **Task:** Sequence Classification (2 classes)

Training Parameters:

Parameter	Value
Max Sequence Length	256 tokens
Batch Size	8 (train and validation)
Learning Rate	1e-5 (0.00001)
Optimizer	Adam Training Implementation Details: ● Training Loop: Hugging Face Transformers library Trainer (recommended) with TrainingArguments ● Evaluation: Compute metrics at end of each epoch using standard accuracy, precision, recall, and F1-score ● Gradient Accumulation: Not used (steps per epoch is small)

	● Mixed Precision: Not used (fp32 training) ● Learning Rate Scheduler: Linear with warmup (default in Transformers Trainer); warmup_steps=100 ● Early Stopping: Not applied (single epoch) ● Validation Split: Internal validation uses 10% of training data (automated by train_test_split with random_state=2021)
Epochs	1
Padding	max_length with truncation

Tokenizer Settings:

- Tokenizer: RobertaTokenizer
- Add Special Tokens: True
- Padding: max_length
- Truncation: True

Estimated Training Time: ~20-30 minutes per dataset (T4 GPU) **Model Size:** ~125 million parameters

5.3 DeBERTa MODEL

Base Model: microsoft/deberta-v3-base (Hugging Face) **Task:** Binary sequence classification
Training Parameters:

Parameter	Value
Max Sequence Length	256 tokens
Batch Size	8 (train and validation)
Learning Rate	1e-5 (0.00001)
Optimizer	Adam
Epochs	1

Tokenizer Settings:

- Tokenizer: DebertaV2TokenizerFast
- Add Special Tokens: True
- Padding: max_length
- Truncation: True

Estimated Training Time: ~25-35 minutes per dataset (T4 GPU) **Model Size:** ~184 million parameters

5.4 XLNet MODEL

Base Model: xlnet-base-cased (Hugging Face) **Task:** Sequence classification

Training Parameters:

Parameter	Value
Max Sequence Length	256 tokens
Batch Size	8 (train and validation)
Learning Rate	1e-5 (0.00001)
Optimizer	Adam
Epochs	1
Padding Side	LEFT (XLNet requirement)

Tokenizer Settings:

- Tokenizer: XLNetTokenizerFast

- Pad Token: eos_token
- Padding Side: left
- Add Special Tokens: True

Estimated Training Time: ~20-30 minutes per dataset (T4 GPU) **Model Size:** ~117 million parameters

6: REPRODUCIBILITY SETTINGS

6.1 Random Seeds

Component	Seed Value
Python Random	2021
NumPy	2021
PyTorch	2021
CUDA	2021 (if available)

6.2 Code to Set Seeds

```
python
import random
import numpy as np
import torch

SEED = 2021
random.seed(SEED)
np.random.seed(SEED)
torch.manual_seed(SEED)
if torch.cuda.is_available():
    torch.cuda.manual_seed_all(SEED)
```

6.3 Data Split Consistency

Use stratified train_test_split with random_state=2021

7: EVALUATION METRICS

7.1 Primary Metrics

- **Accuracy:** Overall correct predictions / total predictions
- **Precision:** TP / (TP + FP) - per class
- **Recall:** TP / (TP + FN) - per class
- **F1-Score:** 2 * (Precision * Recall) / (Precision + Recall)

7.2 Class-Wise Metrics

Compute separately for:

- CG Class (Computer Generated/Fake)
- OR Class (Original/Real)
- Macro Average (unweighted mean)

7.3 Category-Wise Analysis

Compute accuracy for each product category separately. Report mean and standard deviation across categories.

8: EXECUTION STEPS

8.1 Environment Setup (Google Colab)

1. Open new Colab notebook
2. Change runtime: Runtime > Change runtime type > T4 GPU
3. Install dependencies (see Section 2.4)
4. Upload dataset files to Colab
5. Verify GPU availability:

```
python
import torch
print(torch.cuda.is_available())
print(torch.cuda.get_device_name(0))
```

8.2 Running Experiments - Order

For each dataset (DS1, DS2.1, DS2.2, DS2.3, DS2.4):

1. Load and preprocess data
2. Split into train/test (80/20, stratified)
3. Train Attention-GRU model
4. Train RoBERTa model
5. Train DeBERTa model
6. Train XLNet model
7. Evaluate all models on test set
8. Generate category-wise metrics
9. Save results

8.3 Experiment Matrix

Total Experiments: 4 models x 5 datasets = 20 runs **Estimated Total Time:** 8-12 hours (T4 GPU)

Model	**DS1**	**DS2.4**	**DS2.1**	**DS2.2**	**DS2.3**
Attention-GRU	Run 1	Run 5	Run 9	Run 13	Run 17
RoBERTa	Run 2	Run 6	Run 10	Run 14	Run 18
DeBERTa	Run 3	Run 7	Run 11	Run 15	Run 19
XLNet	Run 4	Run 8	Run 12	Run 16	Run 20

9: TROUBLESHOOTING

9.1 Out of Memory (OOM) Errors

- Reduce batch size from 8 to 4 for transformer models
- Clear GPU cache: torch.cuda.empty_cache()
- Restart runtime and reload data

9.2 Slow Training

- Verify GPU is being used: torch.cuda.is_available()
- Check GPU type: Runtime > Manage sessions
- Consider Colab Pro for better GPU allocation

9.3 Tokenizer Warnings

- XLNet: Set pad_token = eos_token before tokenization
- DeBERTa: Use fast tokenizer version

9.4 NaN in Loss

- Check for empty text samples
- Remove rows with NaN in text column
- Verify label encoding is correct (0 and 1 only)

10: REFERENCE LINKS

10.1 Datasets

- DS1 - GPT-2 FakeReviews: https://github.com/joolsa/FakeReviews/tree/main
- DS2 - FraudSquad: https://anonymous.4open.science/r/FraudSquad-5389/README.md
- Amazon Reviews 2023 (base data): https://amazon-reviews-2023.github.io

10.2 Dataset Papers

- DS1 Paper: https://doi.org/10.1016/j.jretconser.2021.102771
- DS2 Paper: https://doi.org/10.48550/arxiv.2510.01801

10.3 Pre-trained Models (Hugging Face)

- RoBERTa: https://huggingface.co/roberta-base
- DeBERTa: https://huggingface.co/microsoft/deberta-v3-base
- XLNet: https://huggingface.co/xlnet-base-cased

10.4 Documentation

- PyTorch: https://pytorch.org/docs/stable/index.html
- Transformers: https://huggingface.co/docs/transformers
- Scikit-learn: https://scikit-learn.org/stable/