AI Course

Capstone Project   
Final Report

For students (instructor review required)

ⓒ2023 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright of this document.

This document is a literary property protected by copyright law so reprint and reproduction without permission are prohibited.

To use this document other than the curriculum of Samsung Innovation Campus, you must receive written consent from copyright holder.

|  |
| --- |
| Fake News & Misinformation Detector |

9/12/2025

Vision Group

< Ahmed Albeladi >

< Raid Almousa >

< Fahad Almousa >

< Abdulaziz Mgarry>

< Thamer Alotaibi >

< Moayad Alharbi >

Content

1. Introduction

1.1. Background Information

1.2. Motivation and Objective

1.3. Members and Role Assignments

1.4. Schedule and Milestones

2. Project Execution

2.1. Data Acquisition

2.2. Training Methodology

2.3. Workflow

2.4. System Diagram

3. Results

3.1. Data Preprocessing

3.2. Exploratory Data Analysis (EDA)

3.3. Modeling

3.4. Testing and Improvements

4. Projected Impact

4.1. Accomplishments and Benefits

4.2. Future Improvements

5. Team Member Review and Comment

6. Instructor Review and Comment

1. Introduction

1.1. Background Information

Fake News is being disseminated at rates never seen before. While there are many good things regarding this, there is also, in my humble opinion, a very current bad thing: there is simply so much misinformation and disinformation that is being spread. Social media, blogs, and even online newspapers are becoming just a smokescreen for lies to spread. This misinformation can distort people's viewpoints; shape public opinion; heighten political divisiveness; and in cases of health and finance, harm people. Therefore, we have established an important area in both academia and industry by building strong systems to identify and eliminate fake news

1.2. Motivation and Objective

The motivation behind this project is the growing need for tools that help readers and platforms distinguish between trustworthy information and deceptive content. We aim to develop and deploy an AI-based system that is able to detect fake news articles with a high level of accuracy. The system would analyze text input, learn patterns between real and fake content, and then classify new articles as real or fake. Ultimately, we are hoping to make a detector that could assist journalists, readers, and fact checking organizations in keeping the online information reliable.

1.3. Members and Role Assignments

* **Raid Almousa & Fahad Almousa**: Responsible for data preprocessing and exploration, which include cleaning the text, stemming, stopwords and exploring the dataset.
* **Abdulaziz Mgarry & Thamer Alotaibi** : Responsible for models’ selection and training, which includes choosing the right models, splitting the dataset, training and fine-tuning.
* **Ahmed Albeladi & Moayad Alharbi :**Responsible for evaluating and validating the models using accuracy, precision, recall, confusion matrix and ROC curve.
* **Member 4 (Project Manager)**: Managed timelines, tracked milestones, and coordinated collaboration among team members.

1.4. Schedule and Milestones

* **Week 1**: Data acquisition and preprocessing, initial model development.
* **Week 2**: Finish the model development and start the model evaluation and validation
* **Week 3:** Model revision and finishing the final report.
* **Milestones**:
  + Data preprocessing completed by mid-Week 1.
  + Model trained and evaluated by the end of Week 2.
  + Final report completed by the end of Week 3.

2. Project Execution

2.1. Data Acquisition

For our project, we used the WELFake dataset, which consists of 72,134 news articles: 35,028 real and 37,106 fake. The dataset was created by merging four widely used datasets (Kaggle, McIntire, Reuters, and BuzzFeed Political) to reduce overfitting and ensure broader coverage of fake and real news sources.

2.2. Training Methodology

A Logistic Regression classifier was trained as a baseline model for fake news detection. Text features were represented using TF-IDF vectors, with stopwords removed and the maximum document frequency capped at 70% to reduce noise from common words. The dataset was split into 80% training and 20% testing. The model was trained with a maximum of 1,000 iterations and regularization tuning to ensure convergence. Evaluation on the test set yielded an accuracy of 94.9%, with a precision of 0.94, recall of 0.94, and F1-score of 0.94. While this model was fast to train and provided strong performance for a baseline, its main limitation was the inability to capture deeper contextual dependencies within the text.

To improve upon the baseline, a Support Vector Machine was implemented using TF-IDF features derived from preprocessed and stemmed text. A LinearSVC model was selected with a regularization parameter 𝐶=1 and a maximum of 1,000 iterations. Hyperparameter tuning involved experimenting with the penalty parameter 𝐶 and performance was validated with 5-fold cross-validation, which produced a stable mean F1-score of 0.91. The model achieved an accuracy of 95.2%, with precision of 0.95, recall of 0.95, and F1-score of 0.95 on the test set. The SVM produced a more stable and consistent decision boundary compared to Logistic Regression, though it required longer training time.

To better capture sequential dependencies and long-range word relationships, a Long Short-Term Memory (LSTM) network was developed. Text was tokenized and converted into padded sequences with a maximum length of 500 tokens, and a vocabulary size of 10,000 words. An embedding layer of 100 dimensions was used, followed by an LSTM layer with 128 hidden units, along with dropout (0.2) and recurrent dropout (0.2) to reduce overfitting. The model was compiled with the Adam optimizer and binary cross-entropy loss. Training was conducted for 5 epochs with a batch size of 64, using a 10% validation split. Hyperparameter tuning included experiments with learning rate, batch size, and sequence length, which improved stability and reduced overfitting. The LSTM achieved the highest performance among the tested models, with 96.1% accuracy on the test set, precision of 0.96, recall of 0.95, and F1-score of 0.96. Although the LSTM required more computational resources than traditional models, it effectively captured context and sequential patterns in the text, making it the best-performing approach.

2.3. Workflow

The workflow we followed can be summarized step-by-step:

1. Load raw dataset (WELFake).

2. Clean data:

2.1. Dropped index column (Unnamed: 0).

2.2.Filled missing title and text with empty strings.

2.3.Removed duplicates (title+text).

2.4.Filtered out non-English articles (~2,100 rows dropped).

3. Text preprocessing:

3.1. Lowercased text, removed punctuation, normalized unicode.

3.2.Removed stop words and applied stemming.

3.3.uilt two versions: clean\_text (for BERT/LSTM) and stemmed\_text (for TF-IDF models).

4. Feature extraction:

4.1.TF-IDF (5000 max features) → used with Logistic Regression, SVM.

4.2.Word embeddings (Word2Vec, Keras Tokenizer) → used with LSTM.

5. Model training:

5.1. Compared Logistic Regression, SVM, LSTM.

6. Model evaluation:

6.1. Classification metrics, confusion matrices, ROC curves.

2.4. System Design

The system was designed to be modular, with distinct components for data preprocessing, model training, evaluation, and deployment. The backend consisted of a Python-based model training script, while the frontend was built using a simple web interface that allowed users to upload their own handwritten digit images for classification. The system used Keras with a TensorFlow backend for model training and Flask for the user interface.

3. Results

3.1. Data Preprocessing

Our preprocessing pipeline was critical in transforming the raw WELFake dataset into a clean, machine-learning–ready form. Key steps included:

• Dropping irrelevant columns: The serial number column (Unnamed: 0) was removed, as it carried no useful information.

• Handling missing values: 558 missing titles and 39 missing texts were replaced with empty strings.

• Duplicate removal: Approximately 8,456 duplicate articles (about 11.7%) with identical title + text were dropped.

• Language filtering: About 2,100 non-English rows were identified and excluded, ensuring the dataset consisted only of English-language articles.

• Text cleaning: Headlines and texts were lowercased, punctuation and special symbols were removed, and unicode was normalized.

• Stopword removal & stemming: A cleaned version of text (stemmed\_text) was created for TF-IDF models, while the unstemmed version was kept for LSTM.

• Combined fields: Title and text were merged into full\_text, ensuring that both headline cues and body content were considered.

Example transformation:

•Before cleaning

Title: “BREAKING: TRUMP WINS!!!”

Text: “Donald Trump has declared victory in a shocking turn of events…”

•After cleaning

Cleaned text: “breaking trump win donald trump declar victori shock turn event”

3.2. Exploratory Data Analysis (EDA)

Exploring the dataset provided valuable insights:

* **Class balance:** After cleaning, the dataset had about **34,791 fake news** and **28,844 real news** articles — a difference of 5,947, meaning the dataset was slightly skewed toward fake articles.
* **Word usage patterns:** Fake news tended to use sensational or emotional words like *“shocking”*, *“exclusive”*, and *“breaking”*, while real news favored neutral, factual wording.
* **Word count analysis:** Real news articles had slightly higher word counts on average, while fake news often relied on shorter, punchy language.

3.3. Modeling

The Logistic Regression model was trained using TF-IDF vectors with stopwords removed and a maximum document frequency of 0.7. It converged after 1,000 iterations and achieved a test accuracy of 94.9%, with balanced precision and recall (both 0.94). Misclassifications mainly occurred when short headlines lacked sufficient context. In addition to the confusion matrix shown in figure 1 .

A blue squares with white text

AI-generated content may be incorrect.

*Figure 1:* Logistic Regression Confusion Matrix

The Support Vector Machine (Linear SVM) was also trained on TF-IDF features with preprocessing and stemming applied. With a regularization parameter 𝐶=1 and 5-fold cross-validation, the model achieved a slightly higher accuracy of 95.2%, with precision and recall values of 0.95 each. In addition to the confusion matrix shown in figure 2 . Errors were fewer than in Logistic Regression but still appeared in ambiguous articles where wording overlapped between fake and real news.

A blue squares with white text

AI-generated content may be incorrect.

*Figure 2:* SVM Confusion Matrix

The final LSTM model consisted of an embedding layer with 100 dimensions, followed by an LSTM layer with 128 units, dropout and recurrent dropout of 0.2 to reduce overfitting, and a fully connected output layer with sigmoid activation. Text sequences were tokenized and padded to a maximum length of 500, and the model was trained with the Adam optimizer and binary cross-entropy loss for 5 epochs using a batch size of 64. The LSTM achieved the best performance, with a test accuracy of 96.1%. A confusion matrix in figure 3 showed that most errors occurred when sensational fake headlines closely resembled legitimate news, or when factual but ambiguous real articles were classified as fake.

A blue squares with white text

AI-generated content may be incorrect.

*Figure 3:* LSTM Confusion Matrix

Figure 4 shows the ROC curve for the Logistic Regression model. With an AUC of 0.99 our model discriminated extremely well between fake and real news articles as shown by the steep rise to the top-left corner of plot, which is where the true positive rate is high and the false positive rate is low.

A graph of a logistic curve

AI-generated content may be incorrect.

*Figure 4:* ROC curve for the Logistic Regression

From Figure 5, we can see the ROC curve for the Support Vector Machine (SVM). with an AUC value of 0.9898, it performed similarly to the Logistic Regression model. The SVM model consistently offered good sensitivity and specificity regardless of the thresholds applied, indicating it was a very robust model.

A graph of a curve

AI-generated content may be incorrect.

*Figure 5:* ROC curve for the SVM

Figure 6 contains the ROC curve for the LSTM model, which performed better than the classical models with an AUC value of 0.9906. It could identify some sequential dependance in the text, which led to slightly more reliable classification as the text neared the boundaries of the class.

A green line graph with black lines

AI-generated content may be incorrect.

*Figure 6:* ROC curve for LSTM

In summary, all three models were near perfect in performance (based on the AUC values) but the LSTM provided the best combination of accuracy (96.1%) and ROC-AUC performance making it the most appropriate choice for deployment.

3.4. Testing and Improvements

Models were tested on a 20% holdout set (12,611 samples) using accuracy, precision, recall, F1-score, confusion matrices, and ROC-AUC. Logistic Regression sometimes misclassified short headlines, which was improved by combining titles and text. The slight class imbalance was reduced using balanced class weights. Early LSTM runs showed overfitting, addressed with dropout (0.2) and batch size tuning.

Through these adjustments, Logistic Regression achieved 94.9% accuracy, SVM improved to 95.2%, and the LSTM reached 96.1%, showing that preprocessing refinements and hyperparameter tuning were key to improving performance.

4. Projected Impact

4.1. Accomplishments and Benefits

The major accomplishments of this project include developing a functional Fake News & Misinformation Detector capable of achieving a high accuracy of 96.1% on the test set, outperforming baseline models such as Logistic Regression and SVM. The project successfully implemented a clean and reproducible data pipeline for preprocessing and feature extraction, along with multiple classification approaches, culminating in the LSTM model as the best-performing solution.

The project exceeded initial objectives by not only building an accurate model but also delivering a system with practical applications for journalists, fact-checkers, readers, and even social media platforms. Beyond technical achievements, the work contributes to addressing the critical social challenge of misinformation, highlighting how machine learning can be applied to support media integrity and informed public discourse

4.2. Future Improvements

While effective, our system has limitations that future work could address:

* **Multi-class labeling:** Distinguish between fake news, satire, opinion, and verified news rather than just binary classification.
* **Multimodal detection:** Incorporate analysis of images, videos, and metadata (source credibility, posting patterns) in addition to text.
* **Scalability:** Optimize the LSTM model for deployment on large-scale platforms with real-time detection needs.
* **Explainability:** Add interpretable AI techniques (e.g., SHAP, LIME) to show *why* an article is flagged as fake.
* **Continuous learning:** Allow the system to update with new labeled data as misinformation evolves.

5. Team Member Review and Comment

|  |
| --- |
| <ATTACH A TEAM PICTURE HERE> |

|  |  |
| --- | --- |
| NAME | REVIEW and COMMENT |
| Ahmed Albeladi |  |
| Raid Almousa |  |
| Fahad Almousa |  |
| Abdulaziz Mgarry |  |
| Thamer Alotaibi |  |
| Moayad Alharbi |  |

6. Instructor Review and Comment

|  |  |  |
| --- | --- | --- |
| CATEGORY | SCORE | REVIEW and COMMENT |
| IDEA | \_\_/10 |  |
| APPLICATION | \_\_/30 |  |
| RESULT | \_\_/30 |  |
| PROJECT MANAGEMENT | \_\_/10 |  |
| PRESENTATION & REPORT | \_\_/20 |  |
| TOTAL | \_\_/100 |  |