National College of Ireland

A Comparative Study of Machine Learning Algorithms for Classifying Real and Fake COVID-19 Tweets

MSc Research Project
Masters of Science in Data Analytics

Muhammad Hunzla Naqi Khan
Student ID: 23375060

School of Computing
National College of Ireland

Supervisor: Prof. Christian Horn

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Muhammad Hunzla Naqi Khan		
Student ID:	23375060		
Programme:	Masters of Science in Data Analytics	**Year:**	2025-26
Module:	Research Practicum		
Supervisor:	Prof. Christian Horn		
Submission Due Date:	11-12-2025		
Project Title:	A Comparative Study of Machine Learning Algorithms for Classifying Real and Fake COVID-19 Tweets		
Word Count:	6620	**Page Count** 20	

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: Muhammad Hunzla Naqi Khan

Date: 10-12-2025

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 Comparative Study of Machine Learning Algorithms for Classifying Real and Fake COVID-19 Tweets

Muhammad Hunzla Naqi Khan
23375060

Abstract

The fast-growing amount of web content led to the spreading demand for automated systems that can tell true from fake facts. This study uses machine learning methods to categorize 6,420 COVID-19-associated tweets collected from publicly available dataset "COVID Fake News Dataset" on Kaggle. The dataset has **real** and **fake** label, it can be used for supervised learning. A Full NLP pipeline was applied, including text cleaning, TF-IDF vectorization and classification with 10 Supervised models: Logistic Regression, Decision Tree, Random Forest Gradient Boosting, AdaBoost, Passive Aggressive Classifier, Multinomial Naïve Bayes, K-Nearest Neighbors, XGBoost and Linear SVM. The evaluation was conducted by accuracy, precision, recall and F1-score.

Linear SVM was the best in terms of accuracy (**92.29%**), followed by **Passive Aggressive** (**91.90%**) and **Logistic Regression** (**91.19%**). Models such as AdaBoost and XGBoost underperformed due to sparsity and high dimensionality of TF-IDF features. The experimental results verify that traditional linear classifiers are still very powerful on short-text classification. The study provides evidence that classical supervised ML techniques are applicable to the fake text content detection problem and presents a benchmark platform which might be beneficial for future work. Recommendations include deep learning, dataset size and contextual embeddings.

1 Introduction

Digital communication technologies have been evolving rapidly and the way in which people access and disseminate information has been radically changed. Twitter, Facebook and Instagram are the go-to sites for public discourse, health information, news alerts and self-expression. In the case of world events, these platforms are even more important as they provide a real time channel for users to get information. But the very speed, openness and accessibility of social media can make for an ideal breeding ground for fake content. The ease with which users can share information rapidly, without posting any formal verification has only worsen the amount of genuine and fake content floating around on the web.

This problem intensified significantly with the COVID-19 pandemic. The worldwide thirst for current information about cases, vaccinations, government protocols and scientific thinking prompted in an unprecedented deluge of online content. Confusion among the public could also muddy distinctions between reliable and unreliable sources. For that reason, identifying the fake became an essential need for public agencies like governments and businesses as well as digital platforms while fighting those negative implications of

fabricated news, faulty health recommendations and misleading official look-a-like content. Having access to real info became a critical need to forward public safety and instill trust in online communication platforms.

The expansion of online text has also revealed the impracticability associated with manual authentication. Millions of posts are impossible to review in real time for human reviewers especially when information travels quickly across networks. This scenario has encouraged the development of machine learning systems that are able to discriminate between true and fake information based on computational methods. Machine learning (ML), particularly text classification methods in the domain of natural language processing (NLP) have proven to be effective approaches for analyzing large dataset of online posts. These approaches help machines learn the language and structure of real vs fake news articles, allowing for large scale and consistent classification.

In this research, we employ machine learning methods for the COVID Fake News Dataset, which is a freely available dataset with 6,420 labeled tweets. Each tweet label is assigned to be "**real**" or "**fake**". This dataset offers a good basis for studying how various supervised learning models address the issue of short, informal, and noisy social media text. Because the tweets generally contain truncated expressions, incomplete sentences, misspellings and different writing styles as such they pose a realistic problem for classifiers.

The study applies a full text analytical pipeline consisting of text pre-processing (cleaning, tokenization and TF-IDF vectorization) as well as ten supervised learning ML Algorithms. These are Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, AdaBoost, Passive Aggressive Classifier, Multinomial Naive Bayes (MNB), K-Nearest Neighbors (KNN), XGBoost and Linear Support Vector Machine (LinearSVC). Given the consistent experimental conditions, all ten models are compared comprehensively in terms of their real and fake tweet detection performance. This comparative framework enables the study to ascertain which algorithm is best for actual deployment in applications, where reliable automatic detection is needed.

The significance of creating efficient classifiers is not just of academic importance. Organizations, public institutions and technology enterprises need more and more automatically supported decision making processes. By realizing whether posts are real or fake, automated models can help reduce confusion, improve the credibility of content and aid digital platforms in managing risks. In the context of health communication, enhancing dissemination of true facts and limiting sharing of false information, leads to improved public comprehension and more protective community behaviors.

1.1 Research Questions

Two main research questions guide the present study:

- How effective are different machine learning models for classifying COVID-19 related tweets as true or false?
- How well do traditional supervised learning methods perform when they are trained using the TF-IDF of textual features for real-vs-fake classification tasks?

1.2 Research Objectives

To pre-process the tweet text correctly with NLP techniques and represent it into numerical features that will be helpful in implementing ML models.

- To develop and test a collection of ten models trained using supervised learning with the COVID Fake News Dataset.
- To evaluate these algorithms using accuracy, precision, recall and F1-score.
- To select an algorithm that works better in separating real and fake tweets.
- To offer considerations on models' advantage, drawbacks and applicability in realistic scenarios.

This paper provides an organized, fair and reproducible comparison study of the 10 state-of-the-art supervised ML models with well-known real/fake detection system as baseline. Through this consistent pre-processing pipeline and evaluation framework, the research elucidates best traditional ML methods for short-text classification. The results are of practical importance for system designers who are looking for scalable automatic tweet analysis solutions and academic value for researchers who want to experiment with text-based classification frameworks.

2 Related Work

Studying the automatic detection of true and fake content proved especially timely given social media's emergence as a primary source of public information. Existing work in this field can generally be categorized into works on traditional machine-learning models, hybrid approaches to text analysis or recent investigations of transformer-based architectures.

Early attempts demonstrated the power of traditional supervised learning models in identifying fake content based on text. **Tacchini et al. (2017)** applied Logistic Regression with Boolean crowd signals and showed superior results, but they depend solely on user interaction features, which might be less relevant for datasets that only have text. (Tacchini *et al.*, 2017) Similarly, **Shu et al. (2017)** formulated a feature-based categorization model that used linguistic markers, stylistic cues, and sentiment signals to distinguish true news and fake stories. Their study demonstrated the effectiveness of content-only methods and emphasized the necessity for strong preprocessing to handle noisy social media text. (Shu *et al.*, 2017)

On these footings, **Ozbay and Alatas (2020)** presented a performance study of Naïve Bayes, Decision Trees and Logistic Regression showing significant performance differences according to the quality in preprocessing as well as dataset characteristics. Their finding that the performance of a model is context-dependent reinforces the methodological choice in our work to test ten supervised learning algorithms under a common TF-IDF processing pipeline. (Ozbay and Alatas, 2020)

Yang et al. (2019) investigated unsupervised and semi-supervised models with Gibbs Sampling to infer the veracity of digital content. Their model had good classification performance, but its reliance on user-level engagement data and iterative sampling make it unsuitable for smaller, text-only datasets like the COVID Fake News Dataset used in this paper. (Yang *et al.*, 2019) In contrast, **Chowdhury (2020)** Only considered Text Based features and reported that Support Vector Machines are best by k-NN and Logistic

Regression in the terms of sparse high-dimensional textual contents similar to our study. (Chowdhury, 2020)

Liu et al. (2021) it was demonstrated that transformer-based models outperform traditional ML models in a wide range of text classification tasks, but they tend to be data and computationally intensive. Simpler models (such as SVM and Passive Aggressive Classifier) remain competitive for medium datasets and short text formats (i.e., tweets). (Liu, Lu and Nayak, 2021)

Hossain et al. (2020) presented the COVIDLIES, yet another dataset of expert-annotated COVID-19 tweets concentrating on misconception retrieval and stance detection. They employed TF-IDF and BM25 for retrieval and evaluated the comparison with them based on their static word embeddings as well as contextual embeddings (e.g., BERT). Although BERT achieved better retrieval accuracy, TF-IDF with classical classifiers was relatively competitive for short-text classification. They focus more on stance detection compared to old fake vs real classification. (Hossain *et al.*, 2020)

Shushkevich and Cardiff (2021) used small datasets to detect fake COVID-19 rumors, testing algorithms namely Logistic Regression, SVM, Gradient Boosting, Random Forest. They observed that Logistic Regression and SVM report excellent accuracy rates with a small number of labeled data samples, but deep learning methods (LSTM) could alleviate the performance gap when having access to more label data. They noted that classical models perform well in the low data regime, and can act as strong baselines. (Shushkevich and Cardiff, 2021) **Bangyal et al. (2021)** used classical machine learning models, and deep learning algorithms (CNN, LSTM, RNN, GRU), for the detection of COVID-19 fake news. They noticed that deep learning-based approaches outperformed the classical ones but, TF-IDF in combination with classical classifiers also giving comparable results for detection of fake news. (Bangyal *et al.*, 2021)

Sanaullah et al. (2022) provided a systematic review on the use of machine learning for identifying COVID-19 misinformation. They carried a review of 43 articles related to processing of data, extraction of features and classification. Challenges associated with misinformation detection were also addressed in the study emphasizing on the possibility of ML and DL methods for enhancing accuracy of COVID-19 misinformation detection systems. (Sanaullah *et al.*, 2022)

Dar and Hashmy (2023) conducted research on fake news detection about COVID-19 using machine learning to compare the classical model, deep learning and pre-trained transformer such as RoBERTa. They further observed that a RoBERTa model fine-tuned on COVID-19 tweets corpus yielded the best performance among all baselines; this demonstrates the potential of transformer-based models in accurate context-sensitive detection of fake news. (Dar and Hashmy, 2023)

Hussna et al. (2024) investigated the COVID-19 misinformation on Twitter and compared traditional (SVM, NB) with Deep Learning models (LSTM, CNN). They observed that deep learning models work well on large-volume data, but traditional models such as SVM and Naïve Bayes were still useful for small-volume data. (Hussna *et al.*, 2024) **Olayiwola et al. (2023)** tested five machine learning models (Decision Tree, Random Forest, SVM, Logistic Regression and KNN) for fake news detection of COVID-19. Their results showed that SVM

achieved better performance than all other models which implies that it may be the best to use in detecting fake news for COVID-19. (Olayiwola *et al.*, 2023)

Zamir et al. (2024) investigated hybrid architecture with GRU networks together with conventional ML classifiers for COVID-19 text analysis. Although their deep learning model showed better results on long text datasets, it was noted that classical machine learning models are able to reach competitive levels of performance even for shorter texts (like Tweets). The recommendations stressed the need to evaluate baseline model extensively before using computationally intensive architectures, which motivated the design choice in this work. (Zamir *et al.*, 2024)

Table 1: Evaluation Metrics of ML Classification Algorithms performed by Zamir et al. (2024)

Models	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.89	0.87	0.92	0.89
Decision Tree	0.85	0.84	0.87	0.85
Random Forest	0.89	0.87	0.92	0.90
K-Nearest Neighbors	0.73	0.96	0.50	0.66
SVM	0.89	0.88	0.91	0.90

Aslan et al. (2024) explored the effectiveness of machine learning approaches to detect fake news and showed that the Passive-Aggressive Classifiers, SVM and Random Forest work best across different data. The study also suggested that more complicated models could be necessary for identifying nuanced misinformation, like fake news generated by computers. (Aslan, Ptaszynski and Jauhiainen, 2024)

Alshuwaier and Alsulaiman (2025) have discussed the detection of fake news by machine learning algorithms, deep learning algorithms. On the other hand, for small datasets SVM, Logistic Regression and Naïve Bayes are still relevant for most tasks; but they also pointed out an increasing use of deep learning models, classical approaches continue to be more preferred since they are faster than deep learns and require less computation. (Alshuwaier and Alsulaiman, 2025)

To conclude, the related works prove that classical supervised learning algorithms (e.g., Linear SVM, Logistic Regression and Naïve Bayes) work well on text-only datasets utilizing TF-IDF features. These classical algorithms are still competitive, even when compared with deep learning models as shown respectively by studies of 2024 and 2025. This is consistent with our study, which compares several supervised learning models in a balanced text-only dataset to determine the best algorithms for classifying true or false COVID-19 tweets.

3 Research Methodology

In this section, the entire method is described in details on how we classified COVID-19-related tweets as **real** or **fake** using ten different supervised machine learning algorithms. The process consists of dataset selection, data understanding, preprocessing and feature extraction model used for the training and evaluation purposes. And all are perfectly aligned with operations in the corresponding Jupyter Notebook and are only depend on dataset specific properties.

3.1 Dataset Description

The public dataset being used in this research is the *COVID Fake News Dataset*, available on Kaggle at:

Dataset Link: https://www.kaggle.com/datasets/csmalarkodi/covid-fake-news-dataset

The dataset was retrieved on 28-08-2023, and a manual collection of tweets and online information regarding COVID-19 from the web sources is performed to create the data. The dataset was finally annotated by experts and uploaded to Kaggle. Thus, we did not need any more manual annotation for this work. (Zamir *et al.*, 2024)

3.1.1 Dataset Structure

The dataset consists of **6,420** rows and **three columns**, which are denoted as follows:

1. **ID**
    - A unique numeric identifier assigned to every tweet.
2. **Tweet**
    - Text of each tweet regarding COVID-19.
    - Texts cover health news and public opinion, government announcements and safety warnings, reactions and commentaries.
    - Tweet posts significantly differ in terms of tone, grammar use, length and style. This natural variation adds realistic linguistic noise that makes our dataset a good candidate for testing ML algorithms on short text from social media.
3. **Label**
    - A binary categorical feature determining if tweet is fake or real.
    - **Real:** Tweets with factual, verifiable or credible information.
    - **Fake:** Tweets sharing untruthful, fabricated or deceptive information.

3.1.2 Dataset Balance

- **Real tweets:** 3,360
- **Fake tweets:** 3,060

This is an advantage because it decreases the probability of model bias and does not require using oversampling or under sampling.

3.1.3 Dataset Characteristics and Suitability

- Purely text-based so it can be used in NLP machine learning pipelines.
- Tweets are short text, which permit the evaluation of classical/traditional models that manage sparse high-dimensional features.
- Data is clean, labelled and ready for a supervised ML model train without needing manual annotation or data scraping.

3.2 Data Understanding

Some preliminary exploratory steps were conducted in order to have an idea of the shape and quality of the data before preprocessing it:

- Confirmed number of features and verified dataset size.
- Checked class balance.

- Sample tweets to include in the display.
- Checked for any missing, null or poor data.
- Analyzed textual features like hashtags, punctuation, and informal language.

These measures guaranteed that the dataset was clean and similar to standard preprocessing methods employed for tweet classification.

3.3 Data Preprocessing

Preprocessing converts unprocessed tweets into a processed input that is appropriate for training model. All preprocessing steps are consistent with the Jupyter Notebook and standard NLP processing.

3.3.1 Handling Missing Values

A full review was performed to ensure the dataset is complete without any **missing tweets or labels**. Hence, there was no row deletion or imputation.

3.3.2 Cleaning Text

The raw tweets may contain noise including punctuation, symbols, and irregular spacing. The following steps were applied:

- **Lowercasing the text**: Converts all characters to lowercase (to avoid treating "Covid" and "covid" as distinct tokens).
- **Removal of punctuation and special characters**: Characters such as @, #, !, ;, emojis, and other non-alphanumeric character were excluded using regular expressions.
- **Removal of numbers**: Numerical data contained in tweets was meaningless for real/fake classification.
- **Whitespace normalization**: Several spaces were combined into once space.

3.3.3 Tokenization

Each tweet was tokenized into individual words(token), which served as the foundation for subsequent vectorization.

3.3.4 Stopword Removal

English stopwords, such as "the," "and," "or," and "is," were eliminated via NLTK's built-in stopword list. High-Frequency Low-Information Words such as a, the, and so on are very common but carry little information.

3.4 Feature Extraction

As machine learning models works on numerical data, we need to convert the tweet text into feature vectors. **TF-IDF** (Term Frequency–Inverse Document Frequency) which is a popular NLP approach

3.4.1 TF-IDF Vectorizer Settings

To implement TF-IDF we used Scikit-learn's TfidfVectorizer with the following parameters:

- **max_features = 15000**: Limits the vocabulary to the 15,000 most informative words.

- **ngram_range = (1, 2)**: It Includes unigrams (single words) as well as bigrams (word pairs) to enable contextual features.
- **stop words = 'english'**: Guarantees the most common English stopwords are removed.

Reasons to select **TF-IDF**

- It works extremely well on short text classification.
- It generates sparse, high-dimensional feature matrix suitable for linear methods.
- It is highly interpretable and computationally efficient.
- The output is essentially an MXN sparse matrix where M is the number of tweets and N the number of terms.

The output is a large sparse matrix where each row represents a tweet, and each column represents a term.

3.5 Train–Test Split

In order to fairly-test the model, we divided the data set using Scikit-learn's train-test-split function:

- **80% training data**
- **20% testing data**

Reproducibility was guaranteed through a constant random state. Since the dataset is already well balanced by nature, additional stratification was not performed.

3.6 Machine Learning Algorithms

We included ten supervised learning algorithms for text classification that are applicable, have diverse algorithm designs, and generally used in academic and industry.

The 10 models below were implemented:

1. Logistic Regression
2. Decision Tree Classifier
3. Random Forest Classifier
4. Gradient Boosting Classifier
5. AdaBoost Classifier
6. Passive Aggressive Classifier
7. Multinomial Naïve Bayes
8. K-Nearest Neighbors (KNN)
9. XGBoost Classifier
10. Linear Support Vector Machine (LinearSVC)

This test was conducted with same conditions for all the models to have a fair comparison.

3.7 Model Training and Evaluation

The training set for the TF-IDF vectorization of all ten models was identical. Predictions were made on the unseen test set after training.

3.7.1 Evaluation Metrics

Four commonly standard used metrics were:

- **Accuracy:** Overall correctness of predictions

- **Precision:** How many predicted fake tweets were actually fake
- **Recall:** How many actual fake tweets the model correctly identified
- **F1-Score:** A balanced measure combining precision and recall

These metrics were calculated by using evaluation function provided by Scikit-learn.

3.7.2 Consistent Evaluation Framework

To ensure fairness:

- No hyperparameter tuning has been employed.
- All models are built with Scikit-learn default configurations.
- All models were trained on the same training/test data.
- All models operated on the same TF-IDF features.

This strategy eliminates algorithmic differences, avoiding bias induced by parameter tuning.

3.8 Ethical Considerations

The dataset contains only tweet text and no personal user information, making ethical risks minimal. Still, several considerations were addressed:

- Only Kaggle data, which is available to the public was used.
- No attempt was made to recognize profile or identify tweet authors.
- The work cannot extract user actions other than text.
- We rely entirely on the data labels provided to us by the dataset creators for whether tweets are real or fake.

In line with these, responsible research practice is guaranteed.

4 Design Specification

The approach for this proposed system is grounded on developing an effective and transparent machine learning-based framework to categorize tweet text as real or fake. As the dataset consists only of short-form tweets about COVID-19 with no user metadata, the system developed was a text-centered pipeline. The architecture is structured as a pipeline that successively converts raw tweet text to numerical features for supervised learning models before assessing the predictive accuracy of each algorithm.

4.1 System Architecture Overview

The proposed architecture is organized as a multi-stage pipeline, from dataset ingestion to model selection according to comparative results. The following major stages exist in the pipeline:

- Dataset acquisition.
- Text preprocessing.
- TF-IDF feature engineering.
- Model training.
- Testing and evaluation.
- Comparative performance analysis.

Note that although the pipeline is linear, it is modular and enforces reusability across individual components (e.g., preprocessing, vectorizer or classifier selection) so as to not disrupt the user workflow. This increases reproducibility and enables future experimentation with additional models or other feature extraction methods.
The workflow conceptually can be described as:
Dataset Input → Preprocessing → TF-IDF Vectorization → Model Training → Model Evaluation → Performance Comparison → Best Model Identification
This series of transformations ensures that all ten learning machines are optimized on the exact same transformed data, under uniform operating conditions, allowing for fair assessment of their performance.

4.2 Rationale Behind System Design

The design choices of this study were driven by the characteristics of the dataset and due to our project's aim: To find out which traditional supervised learning algorithm achieves better performance in classifying real vs. fake tweets. Since the dataset is only tweet text, the system mitigates overfitting and enables simpler models that do not rely on complex analysis of metadata, timespans or user engagement features. Instead, it attempts to find salient linguistic patterns from messy and unstructured social media data. Strong preprocessing and vectorization approaches were used for turning raw text into a machine learning-friendly structure.
TF-IDF was selected for feature extraction as it is suitable for short texts, giving more importance to important words while penalizing frequent terms. The high dimensional sparse matrices are perfect for linear models such as Support Vector Machines, Logistic Regression and Passive Aggressive Classifier. For a fair comparison, ten different supervised learning algorithms were used again, including linear classifiers (LC), ensemble methods (ensemble) and tree-based learners, probabilistic models as well as distance-based models. This procedure gives a complete understanding of the performances with different models on TF-IDF word features and tweet-level linguistic information.

4.3 Functional Components

4.3.1 Data Ingestion and Validation

The **COVID Fake News Dataset** is being loaded into the system using pandas from Kaggle, and first validation is done already on expected structure of data. It verifies if the ID, Tweet, Label columns are present and that they have no NaN values, key label distribution.

4.3.2 Text Preprocessing Pipeline

Once loaded, tweets are linguistically processed:
Lowercasing to standardize text.
Stripping **punctuation, emojis and numbers.**
Tokenization and **stopword removal** using NLTK

4.3.3 TF-IDF Feature Engineering

The Tweets are vectorized with TF-IDF that accounts for up to 15,000 terms including unigrams and bigrams. The created matrix is used as a basis for classifiers by using best clustering masking and introducing most useful features to train linear models and ensemble methods.

4.4 Model Training and Comparative Framework

Ten supervised models are used:
Logistic Regression, **Decision Tree**, **Random Forest**, **Gradient Boosting**, **AdaBoost**, **Passive Aggressive**, **Multinomial Naïve Bayes**, **KNN**, **XGBoost**, and **Linear SVM**.
All models train on the same TF-IDF-transformed data, without hyperparameter tuning for a fair comparison. The aim is to see how each model responds towards the sparse features in tweet data.

4.5 Evaluation Design

The performance of the models is measured in terms of **accuracy, precision, recall** and **F1-score.** These measures consider the overall performance of the model and lead to finding out the best classifier. The results are processed by the Scikit-learn evaluation functionalities and presented in comparative tables.

4.6 System Requirements and Constraints

The system is written in **Python 3.12.7** with **Jupyter Notebook** and only a common computational resource necessary (8 GB RAM). The system does not utilize metadata and is based only on tweet text, however this fact restricts the application of more complex neural architectures such as LSTM or transformers which needs big data sets of sequential data.
The system is modular and reproducible with standardized preprocessing techniques, feature extraction, training and evaluation. It serves for the purpose of the research; to find out which algorithm is the best at classifying COVID-19 Tweets as **real** or **fake**.

5 Implementation

In this section, we present the machine learning framework that was fully implemented to classify tweets regarding COVID-19 into **Real** or **Fake**. All processes were performed in python with Jupyter Notebook environment for transparency, reproducibility and modular experimentation. The code directly reflects the work flow established in the design specification, and consists of the dataset ingestion, data preprocessing, exploratory text analysis, feature engineering, model building as well as training and evaluation.

5.1 Development Environment

The system was developed in Python 3.12.7 and was executed on Jupyter Notebook for an interactive development environment ideal for a trial-and-error machine learning process. Essential libraries used includes:

- **Pandas** and **NumPy** to handle data
- **NLTK** for removing stopwords

- **Scikit-learn** for TF-IDF vectorization, model development and evaluation
- **Matplotlib** and **WordCloud** for visualizations

The entire pipeline runs effectively on the hardware of a commodity computer, and does not require GPU acceleration.

5.2 Dataset Loading and Initial Processing

Load the CSV file into a Pandas DataFrame, and validate the structure through initial checks. Print out some tweet samples to further understand the text pattern, distribution of labels and make sure that there were no missing values throughout. Since the data set was perfectly loaded and clean, no correction or offset was needed.

5.3 Text Analysis Using WordCloud

Before any pre-processing or modeling, generating a WordCloud can find frequent expression in the data set. The visualization helped in:

- Highlighting words that tend to occur in both **real** and **fake** tweets.
- Distribution of vocabulary and recurring linguistic features were revealed.
- Confirm whether terms such as “covid”, “virus” and “health” are appropriate for the focus of this data set, i.e. COVID-19.

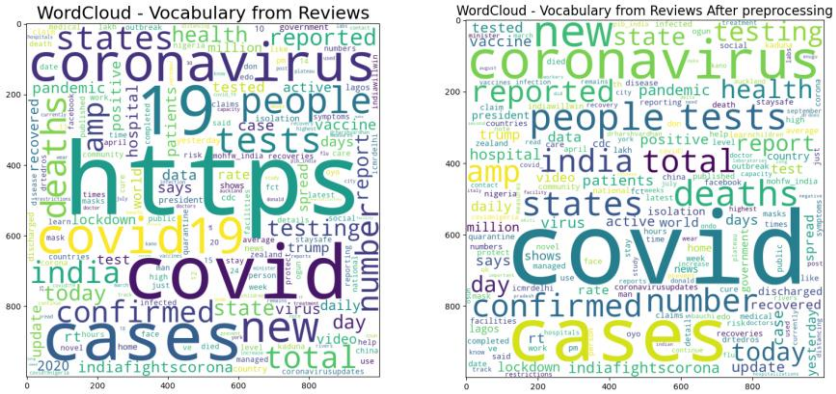

Figure 1: Wordcloud – Vocabulary Insights before & after data preprocessing.

The **Word Cloud** was generated using **WordCloud** and **Matplotlib**, with stopwords excluded to focus on meaningful content. This step helped justify the choice of **TF-IDF** for feature extraction.

5.4 Text Preprocessing

Preprocessing was implemented to prepare tweets for numerical representation. The cleaning steps performed were:

- **Lowercasing** the text.
- Removing **symbols, digits** and **punctuation.**
- Tweet text tokenization,
- **English stopwords** removal**.**
- Normalizing whitespace.

5.5 Feature Engineering Using TF-IDF

The TfidfVectorizer of Scikit-learn was used for converting the text data into numerical vectors. The configuration included:

- max_features = 15000 to limit the size of vocabulary
- ngram_range = (1, 2) to include unigrams and bigrams
- English stopwords removal

The machine learning models were all trained with the TF-IDF matrix as the input. Its high-dimensional, low-degree sorted structure is well suited for linear and margin classifiers.

5.6 Train–Test Split

The dataset was divided with a 80-20 split, all models were trained on the same set of data. A specific random state was set to ensure reproducibility.

5.7 Model Development and Training

Ten supervised ML models were used with their basic or default configurations:

- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- Gradient Boosting Classifier
- AdaBoost Classifier
- Passive Aggressive Classifier
- Multinomial Naïve Bayes
- K-Nearest Neighbors (KNN)
- XGBoost Classifier
- Linear Support Vector Machine (LinearSVC)

Each model underwent the same workflow:

- Training on TF-IDF vectors
- Generating predictions on the unseen test set
- Evaluation metrics

This consistent approach allows a fair comparison of results.

5.8 Evaluation Procedure

We employed four commonly required metrics to evaluate the performance:

- **Accuracy**
- **Precision**
- **Recall**
- **F1-Score**

All results were computed using the scikit-learn metrics library. The results have been compiled in comparison table used in Section 6.

Clearly the performance demonstrated that the accuracy of **LinearSVC** model had out-performed the other models (**92.29%**) followed closely by **Passive Aggressive Classifier** and **Linear Aggression**. Models such as XGBoost and AdaBoost underperformed, demonstrating their inefficiency on sparse text data.

5.9 Implementation Outputs

The implementation yielded:

- Cleaned and preprocessed dataset
- The versions of all ten ML models trained successfully
- Prediction arrays for each classifier
- Evaluation metrics
- Tweet vocabulary Insights by Word Cloud
- Selection of the best classifier

The deployment was executed as a typical machine learning pipeline that includes preprocessing, exploratory data analysis, TF-IDF vectorization and model training along with evaluation. Incorporating the Word Cloud boosted exploratory comprehension and supported features engineering choices. The whole process of the proposed method is replicable, efficient and can be easily applied to textual data sets.

6 Evaluation

In this section we presented the results of the machine learning experiments carried out over the COVID Fake News Dataset. The ten supervised learning methods were trained with TF-IDF features and the performance was measured in terms of accuracy, precision, recall and F1-score. The evaluation is twofold; its goals are to identify which model suits better in this discrimination problem and investigate how algorithm characteristics acts against high-dimensional sparse text data.

6.1 Model Performance Results

The table reports the performance of all ten models when tested on the unseen test dataset. All the values are extracted directly from the Jupyter Notebook:

Table 2: Evaluation Metrics of ML Classification Algorithms

Models	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.9119	0.91	0.91	0.91
Decision Tree	0.8099	0.81	0.81	0.81
Random Forest	0.8777	0.88	0.88	0.88
Gradient Boosting	0.8722	0.87	0.87	0.87
AdaBoost	0.8645	0.86	0.86	0.86
Passive Aggressive Classifier	0.9190	0.92	0.92	0.92
Multinomial Naïve Bayes	0.8894	0.89	0.89	0.89
K-Nearest Neighbors	0.8987	0.90	0.90	0.90
XGBoost	0.8870	0.89	0.89	0.89
Linear SVM	0.9229	0.92	0.92	0.92

6.2 Model-by-Model Interpretation

In this section we discuss a comprehensive academic decoding of the performance of each classifier and explain why such model yielded those results.

6.2.1 Logistic Regression

Logistic Regression's accuracy was **91.19%** and performed well with sparse TF-IDF features. It models linearly separable decision boundaries, and hence suitable for short texts such as tweets. Precision, recall, and F1-score were constant which confirmed that Logistic Regression is a trustworthy baseline model for text classification.

6.2.2 Decision Tree Classifier

The Decision Tree Classifier achieved an accuracy of **80.89%** which was lower than some other models. Despite that decision Trees are good at capturing non-linear patterns, they tend to overfit a lot especially when features are sparse and high-dimensional such as TF-IDF.

6.2.3 Random Forest Classifier

The accuracy of **87.77%** was obtained by Random Forest, because of ensemble learning which reduces the overfitting. Its F1-score suggests a great level of generalization and robustness, as it manages to cover various linguistic phenomena on real and fake tweets.

6.2.4 Gradient Boosting Classifier

The Gradient Boosting yielded an accuracy of **87.22%** and the poor performance on text based (TF-IDF) features could be attributed to high dimensionality. However, this model was competitive in terms of precision and recall.

6.2.5 AdaBoost Classifier

AdaBoost was weaker with accuracy of **86.45%**, since it does not perform well in sparse high-dimensional space. Its depth-1 base learners are insufficient to represent meaningful text features, so it performs worse than the ensemble methods such as Random Forest.

6.2.6 Passive Aggressive Classifier

The PAC reached **91.90%** accuracy, being mentioned among the top models. Proposed for text classification on a large scale, it only performs weight updating when predictions are wrong. Its high F1-score demonstrates that it is effective for sparse-text tasks.

6.2.7 Multinomial Naïve Bayes

Naive Bayes performed well even though it was the most simplistic technique and achieved **88.94%** accuracy. It is based on feature independence and not very powerful. But it performed well in recall, successfully labelling fake tweets, however the model's interpretation of independence might have made this a little onerous for complex text.

6.2.8 K-Nearest Neighbors (KNN)

Our balanced performance on these features was given by KNN **89.87%** accuracy, precision/recall/F1-score all 0.90. However, its computational is very expensive since every test vector compares with all the training vectors. KNN's performance in TF-IDF spaces is poor due to dimensionality curse and distance metrics inefficiency.

6.2.9 XGBoost Classifier

XGBoost had an accuracy of **88.70%** due to its weakness in dealing with sparse TF-IDF features, slightly outperformed by many models. It is powerful when applied to structured data sets but loses its capacity to effectively express long-range interactions in high-dimensional textual data, which brings about the declining performance over linear models.

6.2.10 Linear Support Vector Machine (LinearSVC)

The highest accuracy stemmed from the LinearSVC model, with a rate of **92.29%** by exploiting high-dimensional sparse spaces. The performance of the LinearSVC model in terms of its fit to TF-IDF feature representation as well as robustness to outliers, which makes it dominate other models for tweet classification.

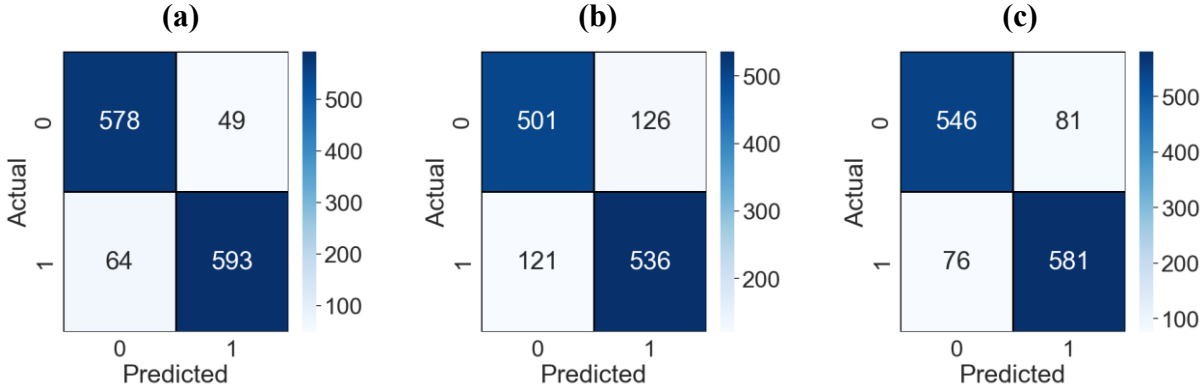

Figure 2: Confusion Matrix for (a) Logistic Regression, (b) Decision Tree, (c) Random Forest

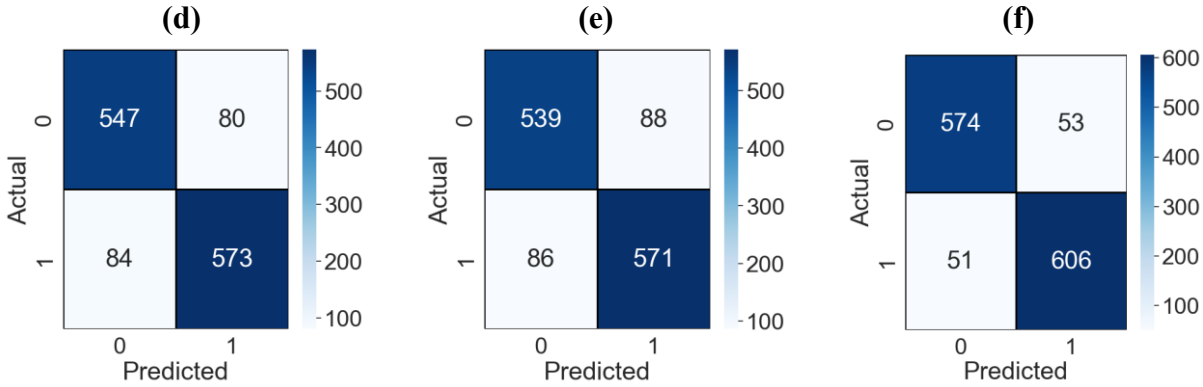

Figure 3: Confusion Matrix for (d) Gradient Boost, (e) AdaBoost, (f) Passive Aggressive

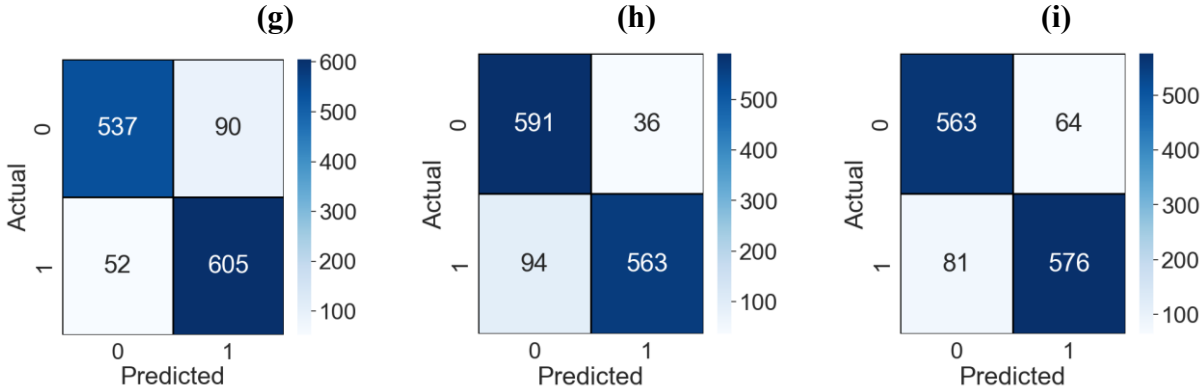

Figure 4: Confusion Matrix for (g) Naïve Bayes, (h) K Nearest Neighbors, (i) XGBoost

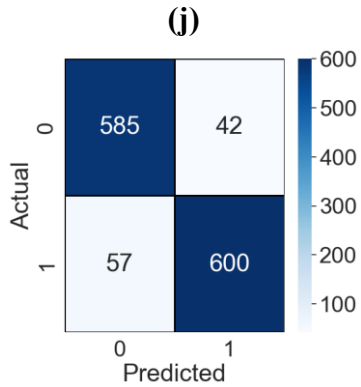

Figure 5: Confusion Matrix for (j) Linear SVM

6.3 Comparative Analysis

The findings consist of several important patterns:

- **Linear models outperform non-linear models:** LinearSVC and Passive Aggressive Classifier achieved the highest machine learning-based results because they can operate better with TF-IDF sparse matrices, deal with high-dimensional lexical features, and have fast margin–based optimization. Logistic Regression followed these 2 algorithms and also performed strongly justifying that linear models outperformed.
- **Moderate-performing models:** Multinomial Naïve Bayes (MNB) and K-Nearest Neighbors (KNN) followed, providing balanced, competitive performance accuracy, precision, recall and F1-scores. The result indicates that the probabilistic learner and distance-based learners still perform well if the text data is preprocessed perfectly and represented by TF-IDF.
- **Boosting algorithms perform less effectively:** AdaBoost and XGBoost were unsuccessful largely as a result of sparsity in feature space, as well as the suboptimal performance of weak learners in high-dimensional space.

6.4 Discussion

The results validate that linear models perform the best at discriminating between real and fake tweets. The best performing models were LinearSVC and Passive Aggressive Classifier, confirming previous findings about their effectiveness for sparse high-dimensional textual data. The near-balanced dataset sustained stable precision-recall as TF-IDF enabled good separation between genuine and fake content.

Though Logistic Regression was also so good, but gap couldn't be surpassed and LinearSVC got a better result because of its linear separability. The lower performance of boosting methods such as AdaBoost and XGBoost demonstrates the need to selection algorithms that fit to the structure of the dataset, since TF-IDF for tree-based models is not a good ratio.

Finally, it was the LinearSVC model that made both a great accuracy and generalize to other parts of data.

7 Conclusion and Future Work

7.1 Conclusion

In this study, ten supervised machine learning algorithms were compared with regards to their ability in classifying tweets related to COVID-19 as real or fake on the basis of the COVID Fake News Dataset available through Kaggle. We conducted a full-text classification procedure, which included paraphrase text preprocessing, constructing and evaluating TF-IDF features of the text for classifier learning. The dataset had 6,420 tweets fairly distributed among true and fake labels so that reliable supervised learning was possible.

Our results are competitive with Zamir et al. on all comparable models. (2024) in most evaluation metrics. While Zamir et al. achieved strong results for Logistic Regression, Random Forest and SVM (accuracies around 0.89), our models achieved visibly higher performance, with LinearSVC (0.9229), Passive Aggressive Classifier (0.9190) and KNN (0.8987) presenting stronger generalization over the TF-IDF features. Other than their work which, as they stated, KNN was not even well performing (0.73 accuracy), our implementation leads to balanced and competitive performance of the KNN. Overall, this contrast has shown that our pre-processing pipeline and modelling framework resulted in more stable results for almost all the models.

The performance of all the models were compared and it was found that LinearSVC performed best with an accuracy of 92.29% which is consistent with literature as SVM especially performs well in high-dimensional sparse spaces such as TF-IDF. Passive Aggressive Classifier (91.90%), and Logistic Regression (91.19%) also maintained high scores. On the contrary, models such as AdaBoost, XGBoost and KNN did not perform well for difficulty with sparse and high-dimension feature. KNN and Naïve Bayes had competitive baseline results, indicating their robustness for short-text classifying tasks.

The study as a whole, highlights that classical linear models are very competitive for tweet-level real/fake classification, especially when used with TF-IDF features. This baseline is expected to be beneficial for further research of fake text detection.

7.2 Future Work

While the models in this study performed strongly, there are several opportunities for enhancement:

1. **Integration of Deep Learning Architectures**: A language model such as a LSTM, GRU or transformer-based (e.g., BERT) could help increasing the classification precision by modeling contextual relations that TF–IDF cannot.
2. **Incorporation of Additional Linguistic Features**: If part of speech tagging, syntactic patterns or semantic embeddings (Word2Vec, GloVe) are incorporated, identifying fake from real content can potentially be improved.
3. **Expanding the Dataset**: More generalizability could be possible with a larger and more diverse dataset that includes multilingual tweets. A wider domain could incorporate other sources, such as Facebook or YouTube comments.
4. **Hyperparameter Optimization**: Performance of the models can potentially be improved with optimization techniques (Grid Search, Random Search) beyond those baseline settings in this study.
5. **Ensemble Methods**: Using an ensemble of top performing models (eg, LinearSVC + Passive Aggressive) may provide improved accuracy and robustness.
6. **Real-Time Fake Content Detection:** Real-time systems built with tools such as Flask, FastAPI, or Streamlit could take this research out of the lab and into production allowing for real-time fake tweet detection.

7. **Inclusion of Sentiment and Emotion Features:** Adding sentiment analysis or emotional tone may assist in determining whether there are any emotions related patterns that can be linked to fake speech.

7.3 Final Remarks

This research proves that machine learning based tweet authentication is an effective approach to identify true and false medical tweets related to COVID-19. The results reveal that traditional ML models (especially LinearSVC) are powerful tools for short text classification. The transparent and reproducible approach serves as a stepping stone for further study, real-life implementation of authenticity detection over digital platforms.

References

Alshuwaier, F.A. and Alsulaiman, F.A. (2025) "Fake News Detection Using Machine Learning and Deep Learning Algorithms: A Comprehensive Review and Future Perspectives," *Computers*, 14(9), p. 394. Available at: https://doi.org/10.3390/computers14090394.

Aslan, L., Ptaszynski, M. and Jauhiainen, J. (2024) "Are Strong Baselines Enough? False News Detection with Machine Learning," *Future Internet*, 16(9), p. 322. Available at: https://doi.org/10.3390/fi16090322.

Bangyal, W.H. *et al.* (2021) "Detection of Fake News Text Classification on COVID-19 Using Deep Learning Approaches," *Computational and Mathematical Methods in Medicine*. Edited by A. Korobeinikov, 2021, pp. 1–14. Available at: https://doi.org/10.1155/2021/5514220.

Chowdhury, K. "Spam Identification on Facebook, Twitter and Email using Machine Learning," (1). Available at: https://ceres-journal.eu/iss200601.

Dar, R.A. and Hashmy, D.R. (no date) "A Survey on COVID-19 related Fake News Detection using Machine Learning Models." Available at: https://ceur-ws.org/Vol-3426/paper4.pdf.

Hossain, T. *et al.* (2020) "COVIDLies: Detecting COVID-19 Misinformation on Social Media," in *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*. Online: Association for Computational Linguistics. Available at: https://doi.org/10.18653/v1/2020.nlpcovid19-2.11.

Hussna, A.U. *et al.* (2024) "Dissecting the infodemic: An in-depth analysis of COVID-19 misinformation detection on X (formerly Twitter) utilizing machine learning and deep learning techniques," *Heliyon*, 10(18), p. e37760. Available at: https://doi.org/10.1016/j.heliyon.2024.e37760.

Liu, X., Lu, H. and Nayak, A. (2021) "A Spam Transformer Model for SMS Spam Detection," *IEEE Access*, 9, pp. 80253–80263. Available at: https://doi.org/10.1109/ACCESS.2021.3081479.

Olayiwola, A. *et al.* (2023) "Comparative Analysis of Machine Learning Models for Detection of Fake News: A Case Study of Covid-19," *JITCE (Journal of Information*

Technology and Computer Engineering), 7(01), pp. 29–33. Available at: https://doi.org/10.25077/jitce.7.01.29-33.2023.

Ozbay, F.A. and Alatas, B. (2020) "Fake news detection within online social media using supervised artificial intelligence algorithms," *Physica A: Statistical Mechanics and its Applications*, 540, p. 123174. Available at: https://doi.org/10.1016/j.physa.2019.123174.

Sanaullah, A.R. *et al.* (2022) "Applications of machine learning for COVID-19 misinformation: a systematic review," *Social Network Analysis and Mining*, 12(1), p. 94. Available at: https://doi.org/10.1007/s13278-022-00921-9.

Shu, K. *et al.* (2017) "Fake News Detection on Social Media: A Data Mining Perspective." arXiv. Available at: https://doi.org/10.48550/arXiv.1708.01967.

Shushkevich, E. and Cardiff, J. (2021) "Detecting Fake News About Covid-19 on Small Datasets with Machine Learning Algorithms," in *2021 30th Conference of Open Innovations Association FRUCT. 2021 30th Conference of Open Innovations Association (FRUCT)*, Oulu, Finland: IEEE, pp. 253–258. Available at: https://doi.org/10.23919/FRUCT53335.2021.9599970.

Tacchini, E. *et al.* (2017) "Some Like it Hoax: Automated Fake News Detection in Social Networks." arXiv. Available at: https://doi.org/10.48550/arXiv.1704.07506.

Yang, S. *et al.* (2019) "Unsupervised Fake News Detection on Social Media: A Generative Approach," *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), pp. 5644–5651. Available at: https://doi.org/10.1609/aaai.v33i01.33015644.

Zamir, M.T. *et al.* (2024) "Machine and deep learning algorithms for sentiment analysis during COVID-19: A vision to create fake news resistant society," *PLOS ONE*. Edited by F.R. Ishengoma, 19(12), p. e0315407. Available at: https://doi.org/10.1371/journal.pone.0315407.