# Introduction

Automated classification of toxicity in tweets or text in general is an important aspect in the real of social media and text recognition. It refers to identifying patterns in data that deviate significantly from the norm. The goal is to identify hate speech or offensive language that could indicate immoral behavior and mitigate consequences prior to significant damage and viral movement. Identifying toxicity in tweets could be challenging because it requires the ability to detect toxic behavior even in subjective circumstances. Machine learning techniques such as Naive Bayes, Support Vector Machine, Logistic Regression and K-nearest Neighbor are commonly used for classification of toxicity in tweets.

In this project the author has elaborated about the problem domain and research. A thorough literature review was conducted to evaluate and identify different techniques and algorithms that can be used in this domain, especially using AI.

Moving on, the author implemented a prototype using a LSTM (Long Short Term Memory) model to classify the toxicity in tweets accurately. LSTM is a type of a neural network, RNN (recurrent neural network) to be exact. LSTMS were primarily architectured to handle the challenges of learning and remembering long-term dependencies in sequential data. LSTMs are particularly effective in tasks involving natural language processing, time series analysis, and speech recognition.

Finally, the model was tested and evaluated against multiple metrics and graphs such as accuracy, precision, recall, auc score, f1 score along with the ROC curve. The author ends with his remarks on the overview and effectiveness of using LSTMs for automated classification of toxicity in tweets.

# Part A - Application Area Review

## Problem Domain

The automated classification of hate speech and toxicity in tweets is a vital task in enabling a safer and friendly online environment. Detecting and mitigating harmful, offensive and hateful language is crucial in preventing harassment, cyber bullying and the spread of toxic and harmful content. Therefore, this type of analysis and classification is important in the realm of social media and text recognition as it can help identify potential problems or issues before they cause significant damage.

The main goal of automated classification of toxicity in tweets is to identify which patterns of text deviate significantly from normal text. This can be done using various machine learning techniques such as Naïve Bayes, clustering, and deep learning.

## Literature Review

Anomaly detection(Elmrabit et al., 2020) using machine learning has become an active area of research in recent years, with many studies exploring the use of various machine learning techniques for this task.

One popular approach is the use of unsupervised learning techniques(Deng and Brown, 2021), such as clustering or density-based methods, to identify unusual patterns in the data. These methods have been shown to be effective in detecting anomalies in various types of data, including images, text, and time series data.

Supervised learning techniques(Elmrabit et al., 2020), such as decision trees, random forests, and support vector machines, have also been used for anomaly detection. These methods involve training a model on labeled data and then using it to classify new instances as normal or abnormal. These techniques have been shown to be effective in detecting anomalies in structured data, such as credit card transactions or network traffic.

Deep learning techniques(Burgetova, Matousek and Rysavy, 2021), such as autoencoders and recurrent neural networks, have also been applied to anomaly detection. Autoencoders are neural networks that are trained to reconstruct the input data, and they have been shown to be effective in detecting anomalies in images and time series data. Recurrent neural networks have been used to model time series data, and they have been shown to be effective in detecting anomalies in sequential data such as audio, video, and sensor data.

A recent study proposed the use of ensemble methods for anomaly detection(Burgetova, Matousek and Rysavy, 2021), which combine multiple models to improve the overall performance. This technique has been applied to time series data and it has proven to be effective in detecting anomalies with high accuracy.

Anomaly detection on multivariate time series(Trardi et al., 2022) data has been a subject of active research in recent years, with many studies exploring the use of various machine learning techniques for this task.

One popular approach is the use of \*\*unsupervised learning techniques\*\*(Limthong et al., 2014), such as clustering or density-based methods. These methods have been used to identify patterns in the data and identify instances that do not conform to these patterns as anomalies. For example, a study proposed the use of a density-based clustering algorithm called DBSCAN to detect anomalies in multivariate time series data from an industrial process.

Another approach that has been used is the use of \*\*dimensionality reduction techniques\*\*, such as Principal Component Analysis (PCA)(Dang et al., no date), to reduce the complexity of the data and make it easier to detect anomalies. For example, a study proposed the use of PCA to reduce the dimensionality of multivariate time series data from a power system, and then used a density-based clustering algorithm to detect anomalies.

\*\*Supervised learning techniques\*\*(Jalal and Ezzedine, 2020), such as decision trees, random forests, and support vector machines, have also been used for anomaly detection. These methods involve training a model on labeled data, and then using it to classify new instances as normal or abnormal. For example, a study proposed the use of a decision tree algorithm to detect anomalies in multivariate time series data from a manufacturing process.

\*\*Deep learning techniques\*\*, such as autoencoders(Choi, Kim and Kang, 2022) and recurrent neural networks(Di Mattia et al., 2021), have also been applied to anomaly detection on multivariate time series data. These techniques have been used to model complex patterns in the data and detect instances that deviate from these patterns as anomalies. For example, a study proposed the use of a deep autoencoder to detect anomalies in multivariate time series data from a transportation system.

A recent study proposed the use of \*\*ensemble methods\*\* for anomaly detection(Burgetova, Matousek and Rysavy, 2021), which combine multiple models to improve the overall performance. This technique has been applied to multivariate time series data and it has proven to be effective in detecting anomalies with high accuracy.

Overall, there are many machine learning techniques that can be used for anomaly detection on multivariate time series data, and each has its own strengths and weaknesses. The choice of technique will depend on the complexity and specific characteristics of the data, the resources available, and the specific problem that needs to be addressed. Multivariate time series data refers to a dataset that contains multiple variables or features that change over time. Anomaly detection in multivariate time series data can be challenging because it requires the ability to simultaneously detect unusual patterns in multiple variables. Machine learning techniques such as Random Forest, Autoencoder, Principal Component Analysis, Support Vector Machine, DBSCAN and Decision Trees are commonly used for anomaly detection in multivariate time series data.

AI has many applications in various domains, such as education, health, entertainment, business, and social media. One of the challenges and opportunities of applying AI to social media platforms, such as Twitter is to detect and prevent hate speech. Hate speech can be considered as any form of expression that attacks or insults a person or a group based on their identity, such as race, religion, gender, sexual orientation, disability, or nationality (Davidson et al., 2017).

Hate speech on Twitter is a serious problem that can have negative impacts on individuals and society, such as psychological harm, discrimination, violence and radicalization. According to a report by the Anti-Defamation League, there were 4.2 million anti-Semitic tweets posted by 3 million users in 2017, which represents an increase of 38% from 2016 (ADL, 2018). Moreover, a study by Amnesty International found that 7.1% of tweets sent to women politicians and journalists in 2017 were abusive or problematic, which amounts to one abusive tweet every 30 seconds (Amnesty International, 2018). Therefore, to ensure the safety of victims as well as ethical and maintain ethical and standards on platforms there is a need to for effective and efficient methods and tools to detect and counter hate speech on Twitter.

However, hate speech detection on Twitter is a very challenging task, due to the complexity and diversity of the natural language used on the platform, such as slang, abbreviations, emojis, hashtags, sarcasm, irony, and humor.

In recent years, AI has been increasingly applied to the domain of hate speech detection on Twitter, using various techniques and methods, such as natural language processing, machine learning, deep learning and computer vision.

Some of the main techniques and methods that have been used for hate speech detection on Twitter are:

* Support vector machine. SVM can handle high-dimensional and non-linear data, and avoid overfitting by finding the best hyperplane that maximizes the margin/distance between the closes data points of different classes. For example, Nobata et al. (2016) used SVM to detect hate speech on Twitter, based on various features, such as word n-grams, character n-grams, part-of-speech tags, sentiment scores, and topic models. They found that SVM achieved an F1 score of 0.84, which was higher than other methods, such as Naive Bayes, logistic regression, and random forest.
* Pre-trained language models: Pre-trained language models are deep learning models that have been trained on large amounts of text, such as BERT, GPT, or XLNet, to learn the general patterns and structures of natural language. Pre-trained language models can be fine-tuned or adapted to specific tasks or domains, such as hate speech detection, by adding additional layers or parameters, and using a smaller amount of labeled data. For example, Aluru et al. (2020) used BERT, a pre-trained language model based on the transformer architecture, to classify tweets as hateful or not. They found that BERT achieved an F1 score of 0.91, which was higher than other baselines, such as SVM, CNN, and LSTM.
* Deep learning techniques: Deep learning models attain the most relevant or important parts of the input or output, such as words, sentences, or images. Attention mechanisms in specific can enhance the performance of deep learning models by providing a way to weigh or highlight importance of each result to the output. For example, Zhang et al. (2018) used an attention-based LSTM model to detect hate speech on Twitter, by assigning different weights to different words in a tweet, based on their relevance to the task. They found that their attention-based LSTM model achieved an F1 score of 0.77, which was higher than other models, such as SVM, CNN, and LSTM without attention.

These are some of the examples of how AI is applied to the domain of hate speech detection on Twitter, using various techniques and methods. However, there are still many challenges and limitations that need to be addressed and overcome, such as the lack of consistent and reliable datasets, the difficulty of capturing the context and nuances of hate speech, the ethical and legal issues of censorship and moderation, and the potential biases and harms of AI systems. Therefore, there is a need for further research and innovation in this domain, to develop more effective, efficient, robust, fair, and responsible AI solutions for hate speech detection on Twitter.

# Part B - Compare and Evaluate AI Techniques

The author’s research is based on automated classification of toxicity in tweets. In this section, the author analyzes the possible machine learning techniques that can be used for automated classification of toxicity in tweets and implements a technique which detects the toxicity in tweets.

The goal of the project is to identify the possible machine learning techniques that can be used for the classification of toxicity in tweets. Analyze each of the techniques and identify their strengths, weakness, advantages and disadvantages and finally implement a machine learning technique that gives out good accuracy of the prediction.

The 3 AI techniques in this project that will be explored and discussed further for classification of toxicity in tweets are:

1. Support Vector Machine
2. Random Forest
3. LSTM (Long Short-Term Memory)

Support Vector Machines (SVM):

SVM can be applied by representing tweets as feature vectors using techniques like TF-IDF or word embeddings. SVM attempts to create a hyperplane that separates hate speech from non-hate speech in the feature space. And generally, the input data for SVM typically involves preprocessed text data in vectorized form (e.g., TF-IDF or word embeddings). The expected output is a binary classification indicating whether a tweet contains hate speech or not.

Strengths:

Effective in High-Dimensional Spaces: SVM works well in high-dimensional spaces, making it suitable for text classification tasks with a lot of features.

Robust to Overfitting: SVM is less prone to overfitting, especially in high-dimensional spaces, due to its margin maximization objective.

Versatility: It can handle various types of data including non-linear data by using different kernel functions.

Weaknesses:

Computational Intensity: SVM can be computationally intensive, especially with large datasets.

Difficulty with Noisy Data: It may perform poorly with noisy data or when classes are heavily overlapping.

Parameter Sensitivity: The performance of SVM is sensitive to the choice of hyperparameters, and finding the optimal parameters can be challenging.

Overall, SVM is an efficient and versatile algorithm which can handle high-dimensional data and overfitting. However, it may not do well in environments which have a lot of noise and less computation capabilities.

Random Forest:

Random Forest can be applied to hate speech detection by using features extracted from tweets and training an ensemble of decision trees. Each tree contributes to the final classification decision.

Random Forest, like SVM, requires vectorized input data. The expected output is a binary classification indicating whether a tweet contains hate speech or not.

Strengths:

Ensemble Learning: Random Forest is an ensemble method, combining multiple decision trees, which helps improve accuracy and generalization.

Robust to Overfitting: Similar to SVM, Random Forest is less prone to overfitting.

Handles Non-Linearity: It can capture complex relationships and non-linear patterns in the data.

Weaknesses:

Interpretability: Random Forest models are often considered as "black boxes," making it challenging to interpret the reasoning behind predictions.

Computationally Intensive Training: Training a large number of trees can be computationally expensive.

Potential for Bias: If the dataset is imbalanced, Random Forest may show bias towards the majority class.

Overall, random forest is a powerful and versatile tool which can handle non-linearity well and is robust to overfitting. It also can be generalized due to its ensemble structure. However, it requires a lot of computational power, and it has the potential for biases. Also, it is quite difficult to interpret the meaning behind its predictions.

Long Short-Term Memory (LSTM):

LSTMs can be applied to hate speech detection by treating tweets as sequential data and learning the temporal patterns of hate speech. The sequential nature of LSTMs allows them to capture context and dependencies between words in a tweet.

LSTMs require tokenized and sequenced input data. The expected output is, similar to other techniques, a binary classification indicating whether a tweet contains hate speech or not.

Strengths:

Sequential Learning: LSTM is designed to capture dependencies and patterns in sequential data, making it suitable for analyzing the temporal nature of tweets.

Handles Variable-Length Sequences: LSTMs can process variable-length sequences, which is beneficial for tweets of varying lengths.

Effective for Text Data: LSTMs are particularly effective in handling natural language processing tasks due to their ability to capture long-range dependencies.

Weaknesses:

Computational Complexity: Training LSTMs can be computationally expensive, especially with large datasets.

Prone to Overfitting: LSTMs can be prone to overfitting, and regularization techniques may be necessary.

Interpretability: Like other deep learning models, LSTMs can be challenging to interpret.

Overall, LSTM can handle text with multiple variations and sizes and is efficient for sequential data. It is however prone to overfitting and requires high computational power.