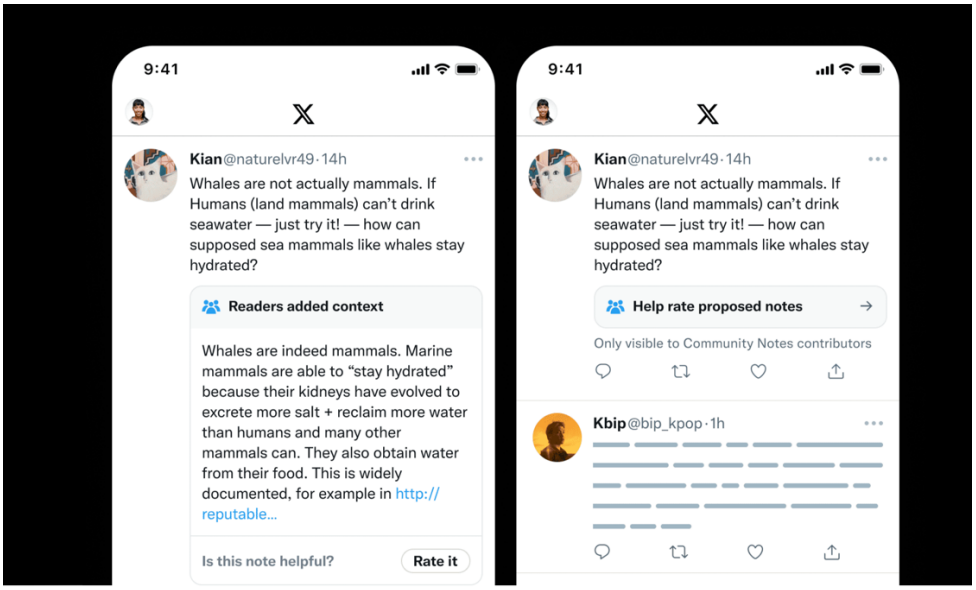
Chapter 1: Introduction

The internet offers a highly integrated global platform that anybody may use to share knowledge quickly and inexpensively to millions of people (Berners-Lee et al., 2010). The rise of social media platforms throughout the past 10 years has raised the importance of interacting and connecting digitally in everyday life. By creating and sharing material about anything, these social media platforms enable people to voice their opinions rapidly and with no constraints, irrespective of their personality traits or behaviours. Twitter is currently ranked among the world's most popular social media platforms.

On the popular social networking site Twitter, users may exchange short communication known as "tweets." With hundreds of thousands of users, Twitter is a significant forum for social interaction, debate, and the sharing of up-to-date information. Twitter (now X) had 436 million monthly active users and 237.8 million daily active users as of 2023, when it managed to post 500 million tweets a day. (HubSpot, 2023, Business of Apps, 2024, Tech Report, 2024). Twitter messages are intended to promote rapid and effective communication, with a character restriction of 280. With this capability, information can be conveyed rapidly on anything from current events to everyday happenings, making Twitter an essential tool for interacting on a worldwide scale. But the website's openness and rapid information flow have also made it a haven for the transmission of misleading information, which has the potential to harm society by manipulating public opinion and encouraging false narratives (Kumar and Shah, 2018).



In 2021, Twitter introduced Community Notes (formerly known as Birdwatch) to improve the accuracy of information published on the site. Users can take advantage of this project to add evidence or clarifications to tweets that they think to be false. After a collective assessment, other users could evaluate the value of these notes, increasing the verification procedure and aiming to enhance the clarity and precision of the material (Roth, 2021). This feature attempts to improve the accuracy and reliability of information on the

platform through promoting openness and communication, and teamwork, allowing community members to actively participate in its advancement (Zannettou et al., 2019).

Even with these developments, it is still hard to recognise and classify inaccurate data. While human moderation and community-driven projects have their role, technology-driven approaches like machine learning may improve the ability to scale and effectiveness of fraudulent data identification. Several machine learning methods have been investigated in earlier research to identify and reduce false information on Twitter. For example, Vosoughi, Roy, and Aral (2018) highlighted the need for automated methods to identify misleading information more efficiently. Similarly, Bian et al. (2020) demonstrated how machine learning could be employed to classify tweets based on their legitimacy.

Furthermore, Shu et al. (2017) carried out a detailed evaluation of methods for discovering fraudulent material on social media, highlighting the need of employing machine learning models to handle the enormous amounts of data produced daily. Adding to this collection of work, Ahmed et al. (2018) Ahmed et al. (2018) conducted a thorough examination of the use of Natural Language Processing (NLP) approaches to detect misleading material, underlining the crucial role NLP plays in recognising the linguistic features of incorrect information. Their review classifies different NLP techniques into discourse-level, lexical, syntactic, and semantic approaches, illustrating how these methods may be applied to analyse information successfully at the context and content levels.

Objectives of the Study

The primary objective of this study is to evaluate the effectiveness of Twitter's Community Notes in enhancing the detection and categorization of false information on the platform.

Specific Objectives:

1. **Assess the impact of Community Notes on misinformation accuracy:** Evaluate the extent to which Community Notes contribute to improving the accuracy and reliability of information on Twitter, particularly in distinguishing between deceptive and non-deceptive content.
2. **Examine the influence of Community Notes on misinformation detection:** Investigate how user-generated notes impact the identification of false information, with a focus on their overall contribution, relevance, and detail in the misinformation detection process.
3. **Investigate opportunities to enhance classification accuracy:** Explore potential improvements in distinguishing between misleading and non-misleading content, considering the role of additional contextual information provided by Community Notes.

By concentrating on these goals, this study seeks to further knowledge about how Twitter Community Notes may be used to enhance the veracity of content on the network and promote more trustworthy public debate.

Chapter 2 : Literature Survey

Introduction to the Landscape of Misinformation Detection

In the digital era, misinformation has become one of the most significant issues, especially on social networking sites where news tends to circulate more quickly than it is validated. Due of this, inaccurate data may now spread unchecked and have significant social repercussions in a complex and dynamic setting. Researchers and politicians are concerned about the spread of false information because it poses serious risks to democratic processes, public health, and public confidence (Lazer et al., 2018). Because of their open nature and the simplicity with which information can be easily spread and increased, social media platforms such as Twitter have been especially vulnerable to the dissemination of false information. Several high-profile events, such as voting, health emergencies, and global wars, have drawn attention to the worldwide impact of disinformation. False narratives have caused widespread misunderstanding, fear, and dangerous conduct (Vosoughi, Roy, & Aral, 2018).

The importance of eliminating disinformation cannot be emphasised. In an increasingly technologically advanced world, the integrity of information is critical. Misinformation can destroy public faith in institutions, influence election results, and escalate public health crises. As a result, developing effective ways to identify and reduce disinformation is not only a technological difficulty, but also a societal responsibility (Pennycook & Rand, 2018). Without legitimate means for distinguishing truth from untruth, the foundation of responsible decision-making in democratic society is compromised. This emphasises the need of expanding research in this sector, as well as the vital role that technical solutions and community participation play in ensuring public discourse integrity.

Key Concerns in Misinformation Detection

Technical Challenges: One of the key issues in disinformation detection is the massive volume of data created by social media sites. The huge amount of information generated everyday needs the development of robust and effective detection systems that are capable of processing and analysing data in real time. Traditional solutions, like manual moderation and phrase sorting, are no longer enough for handling both the scope and complexity of the problem (Gupta et al., 2014). These methods typically fail to capture the complex nature of disinformation, which can be subtle and dependent on context, necessitating more sophisticated approaches. Machine learning and natural language processing (NLP) are growing robust methods for handling these difficulties by crowdsourcing the investigation of huge amounts of data. These technologies enable the identification of patterns and abnormalities in text, which may indicate the existence of disinformation (Shu et al., 2017). However, developing models that can efficiently determine between real and misleading information stays an important challenge due to the complicated structure of language and evolving techniques utilised by those distributing disinformation (Conneau et al., 2017).

One of the most major technological issues is dealing with the constantly shifting nature of disinformation. As countermeasures come into effect, individuals who distribute disinformation change their strategies, making detection a constant fight. Furthermore, the global and diverse traits of social media add an element of complexity, necessitating models that are adaptive and successful across several languages and cultural situations. Continuous learning and modification are essential for these models' long-term performance (Liu & Wu, 2018). Furthermore, the issue of simplicity in machine learning models is crucial; stakeholders must understand how and why a piece of material is identified as misleading, which may be tough with complex algorithms like models based on deep learning (Ribeiro, Singh, & Guestrin, 2016).

Social and Ethical Concerns: The use of automated disinformation detection systems presents several ethical considerations. These include the possibility for restriction, where legal information might be incorrectly classified as fraudulent, and the risk of propagating biases in the training data (Binns, Veale, & Van Kleek, 2018). These biases might result from the information utilised in building models, that may reflect underlying biases in society, resulting in unfair results for some groups of individuals (Noble, 2018). Furthermore, privacy problems arise since these systems frequently demand access to enormous volumes of information about users, which might be exploited if not regulated properly (Floridi, 2016). The relationship among security and confidentiality is complex, and it essential that such systems do not go too far in their data collecting and processing techniques (Floridi & Taddeo, 2016).

Given these obstacles, it is evident that misleading information detection is a key area of research with far-reaching impacts on society. Failure to successfully handle disinformation has the potential to reduce trust in the media and institutions, leading to a more fractured and polarised society (Lazer et al., 2018). likewise, the legal issues concerning disinformation identifying highlight the need for platforms that value transparency, responsibility, and fairness in the deployment of these tools. As disinformation continues to develop, so do the methods and technology adopted to deal with it, to guarantee they are effective open to all, and legal (Pennycook & Rand, 2018).

Efforts and Methodologies in Misinformation Detection

Early Approaches: The early efforts to prevent disinformation on social networking sites were mostly dependent on human filtering and easy keyword evaluation. These methods, however, partially efficient in smaller-scale options, were instantly overloaded by the sheer number and variety of posts on platforms that included Twitter (Gupta et al., 2014). Manual moderation is costly and not scalable, whereas search term screening might overlook quiet or well-hidden disinformation. The limits of these first attempts underlined the necessity for expanding and powerful techniques, particularly when misleading information become more complicated and difficult to identify (Shu et al., 2017).

As the number of social media platforms rose, so did the requirement for automated false information detection systems. Early automated approaches included systems based on rules that recognised text based on defined standards, but they were often too rigid to deal with the wide variety and complexities of human

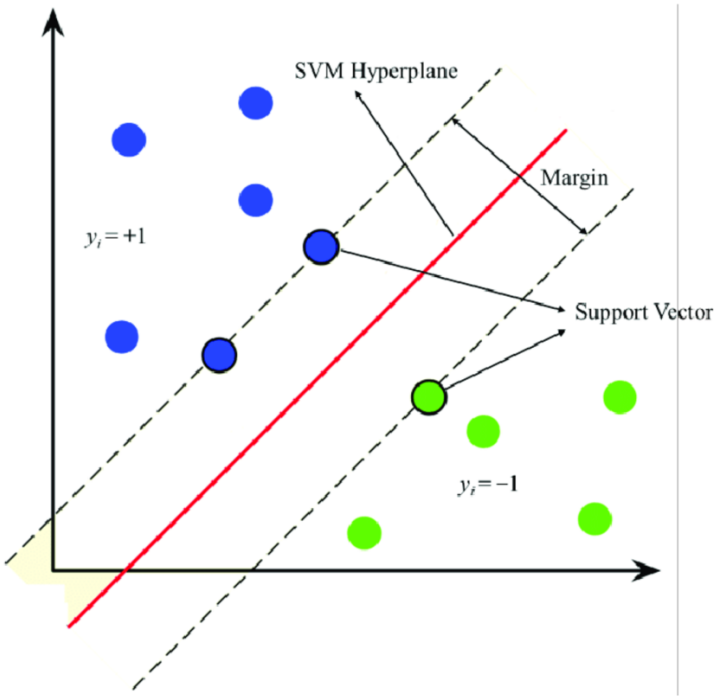
language (Conneau et al., 2017). Eventually, these techniques have been enhanced with machine learning algorithms capable of learning from big datasets and improving accuracy over time. Still, earliest predictive models struggle to deal with the enormous variety of information and continuously transforming approaches used by those promoting disinformation (Liu & Wu, 2018).

Machine Learning and NLP: In response to these issues, academics started looking into the incorporation of machine learning and natural language processing to streamline disinformation recognition. These developments allowed the developing of classifiers that identify possible inaccurate data via analysis of patterns of language and user actions (Shu et al., 2017). NLP methods have been beneficial for finding smaller linguistic information that may illustrate lies, for example the use of dramatic or politically charged language (Pennebaker, 2011). For example, network behaviour analysis may follow the flow of knowledge through interactions among users, detecting dissemination patterns typical of disinformation campaigns (Ruchansky, Seo, & Liu, 2017). These developments are a big step forward in the battle against disinformation, enabling better and efficient detection methods that will operate in real time (Conneau et al., 2017).

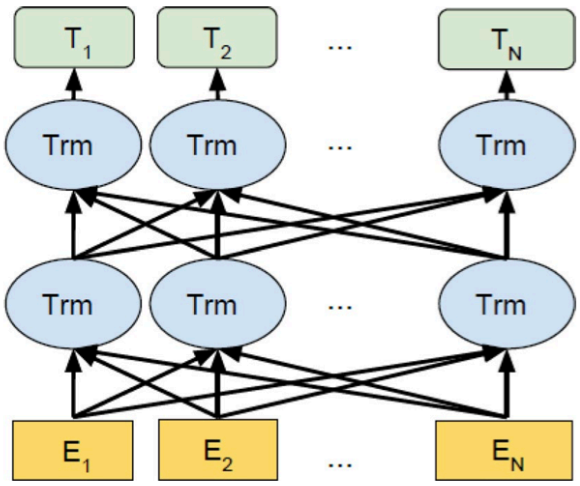
Recent advances in deep learning, specifically through the utilisation of transformers and attention methods, have enhanced NLP's capabilities for recognising misleading information. Models like BERT have proven a remarkable capacity to comprehend contextual and logic, which makes them particularly valuable for problems like analysis of sentiment, named item verification, and, most significantly, misleading information prevention (Devlin et al. 2018). These models use large-scale pre-training on different text databases to detect a broad range of language patterns and transfer this information for tasks via fine-tuning (Rogers et al., 2020).

Specific Techniques and Models: Support Vector Machines (SVM) and Bidirectional Encoder Representations from Transformers (BERT) are two of the most utilised machine learning techniques to identify misinformation.

SVMs are commonly used for problems with classification for their accuracy in feature spaces that are highly dimensional. They have been appropriately utilised for text identify assignments such as spam detection and sentiment analysis, which makes them a powerful instrument in the conflict against false information (Joachims, 1998; Castillo et al., 2011). SVMs' diversity, particularly using kernel-based techniques like as the Radial Basis Function (RBF), helps them to take on non-linear classification problems, which is crucial while dealing with the complicated nature of textual data. Their ability to manage limited and complex datasets, which include ones produced by social media sites, makes them especially ideal for this sector (Agarwal et al., 2011).



Devlin et al. (2018) developed BERT, which introduces bidirectional transformer training, illustrating an important leap in NLP. This method enables BERT to better capture word context than previous models, which makes it appropriate for challenging analysis of text tasks like disinformation detection (Devlin et al., 2018; Rogers et al., 2020). BERT's bidirectional characteristic enables it to look into an entire scheme of a word by examining the words that come prior to and following it in an expression, which gives greater understanding of semantics and subtlety (Sun et al., 2019). BERT's architecture has established new standards in a variety of NLP tasks, particularly question answering, emotion analysis, and text categorising, resulting in an essential element of present NLP research.



Related Work on Community Notes

Introduction to Community Notes: Twitter's Community Notes (previously Birdwatch) is an innovative strategy to addressing disinformation that incorporates the contributions of the community. This function allows users to contribute context or modifications to tweets, which are subsequently determined by other users for correctness and relevance (Roth, 2021). The aim is to make use of Twitter's user base's collective knowledge to improve the quality of data available on the network, which is in tune with a wider trend of crowd-sourced filtering of content (Pennycook et al., 2020). Community-driven methodologies like this offer the ability to enhance machines via providing context and concepts that algorithms might miss (Zannettou et al., 2020).

Community Notes seems especially fascinating from a research aspect as it blends the potential of human evaluation alongside the scalability of online communities. By permitting users to be involved in the method of moderation, Twitter aims to develop a more readily available and transparent system for managing false information. This approach yields an original database comprising not just those initial messages but additionally annotations and conversations that follow, providing essential information for analysing the dynamics of disinformation and responses from the community (Pennycook et al., 2020). The combination of these human-generated ideas with machine learning methods offers a possibility to improve the precision as well as dependability in misleading detection tools (Roth, 2021).

Existing Research on Community Notes: Although the potential of Community Notes, research on their utility is still in its early phases. Some researchers have investigated the success rates of community-driven methods to misleading recognition, indicating that such methods can be helping when appropriately applied (Pennycook & Rand, 2021). However, there is a shortage of thorough research on how Community Notes could potentially be combined with machine learning algorithms to improve their usefulness. Exploring this combination is an exciting field for research since it has the potential to greatly improve the ability to scale and precision for misinformation detection (Zannettou et al., 2020).

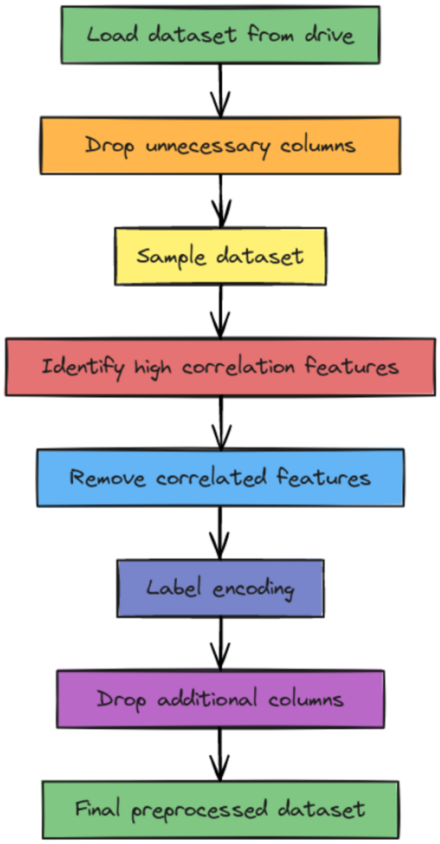
Identifying the Research Gap

While significant advancement has been obtained in the area of disinformation identification, the incorporation of a community-driven insights into machine learning models remain untapped. The current research aims to solve this problem by investigating at how the information offered via Community Notes may be used to boost the precision and effectiveness of misleading detection systems. This study extends to the broader effort to battle false information on social media by offering new perspectives on how interaction between communities might improve automated processes.

Chapter 3: Methodology

Design Philosophy

This study is based on three key concepts: robustness, transparency, and scalability. These guiding principles ensures that each analytical decision is thoughtful and in line with the overall objective of designing a credible and effective classification approach to Twitter Community Notes. By maintaining these standards, the study guarantees that the models are both reliable as well as easily understood, while still being able to manage complicated and complex information found on social networking networks.



Data Acquisition

The fundamental basis of every machine learning model relies on the accuracy and relevance of the data it is trained on. This research utilized data gathered through the Twitter Community Notes program, an openly accessible and regularly updated dataset comprised of user-supplied information specifically connected to the evaluation of community notes. This aligns with the principle of scalability, as the dataset is both extensive and constantly evolving, enabling the model to be retrained and modified as new data becomes available (Vaughan & Bogg, 2022).

The data was gathered from Twitter's Community Notes, a platform that regularly refreshes user-contributed evaluations. This collection of data provides annotations and ratings related to the accuracy and utility of community notes, making it a significant resource for researching misleading information and the contextualization of content on social media platforms. The dataset contains essential fields such as noteId,

participantId, tweetId, classification, and other factors that identify whether the community notes are misleading or not.

Column Name	Type	Information about the column
noteId	Long	The unique ID of this note
participantId	String	A Community Notes-specific user identifier of the user who authored the note.
createdAtMillis	Long	Time the note was created, in milliseconds since epoch (UTC).
tweetId	Long	The tweetId number for the tweet that the note is about.
classification	String	User-entered multiple-choice response to note writing question: “Given current evidence, I believe this tweet is:”
misleadingOther	Int	User-entered checkbox in response to question “Why do you believe this tweet may be misleading?”
misleadingFactualError	Int	User-entered checkbox in response to question “Why do you believe this tweet may be misleading?”
misleadingManipulatedMedia	Int	User-entered checkbox in response to question “Why do you believe this tweet may be misleading?”
misleadingOutdatedInformation	Int	User-entered checkbox in response to question “Why do you believe this tweet may be misleading?”
misleadingMissingImportantContext	Int	User-entered checkbox in response to question “Why do you believe this tweet may be misleading?”
misleadingUnverifiedClaimAsFact	Int	User-entered checkbox in response to question “Why do you believe this tweet may be misleading?”
misleadingSatire	Int	User-entered checkbox in response to question “Why do you believe this tweet may be misleading?”
notMisleadingOther	Int	User-entered checkbox in response to question “Why do you believe this tweet is not misleading?”
notMisleadingFactuallyCorrect	Int	User-entered checkbox in response to question “Why do you believe this tweet is not misleading?”
notMisleadingOutdatedButNotWhenWritten	Int	User-entered checkbox in response to question “Why do you believe this tweet is not misleading?”
notMisleadingClearlySatire	Int	User-entered checkbox in response to question “Why do you believe this tweet is not misleading?”
notMisleadingPersonalOpinion	Int	User-entered checkbox in response to question “Why do you believe this tweet is not misleading?”
trustworthySources	Int	Binary indicator, based on user-entered multiple choice in response to note writing question “Did you link to sources you believe most people would consider trustworthy?”
summary	String	User-entered text, in response to the note writing prompt “Please explain the evidence behind your choices, to help others who see this tweet understand why it is not misleading”
isMediaNote	Int	User-entered checkbox in response to question “Is your note about the Tweet or the image?”. New as of 2023-05-24.

Data pre-processing

Data cleaning is a method of detecting and eliminating errors and irregularities in the data to enhance its level of accuracy. For this study, columns which were no longer recommended or significant, such as credible, damaging, and validationDifficulty, were dropped. The utilisation of these columns was stopped in October 2022 due to an enormous amount of data being absent. This can potentially add disturbance and distortion into the prediction. Data cleaning keeps to the notion of resilience by providing that only appropriate and highly qualified data are used in the creation of models (Kandel et al., 2022).

Deprecated columns were removed due to their lack of usefulness or the presence of numerous missing values. This step ensures that the available data is optimized and narrowed down to include only the variables relevant to the classification problem. The dataset underwent a thorough examination to identify any errors, such as duplicate records or conflicting data entries. These irregularities were corrected to maintain the quality of the data.

Handling Missing Values

The presence of missing data is common issue in datasets that occur in real-world scenarios, and managing these missing values leads to in models that are biased and predictions that are inaccurate. The study utilised mean imputation as an approach for handling missing data for the numerical columns. Mean imputation is a method of filling the data points that are missing by an average value of the column that corresponds. This helps maintain the broad distribution of the results and avoids the chances for creating bias. The technique proves particularly helpful whenever there is a minimal number of data that is missing. It ensures that the model stays robust while the findings stays simple to follow (Zhou et al., 2022).

Mean imputation is applied to address missing values in numerical fields by calculating the mean value of each column and using it as a substitute for the missing data. This approach helps maintain the statistical properties of the dataset and ensures that the model can effectively learn from the data. In cases where categorical variables have missing values, the most frequent category is used to fill in the gaps, ensuring that the distribution of categories is preserved.

Correlation Analysis

The correlation analysis is used in this research project as a starting point of looking into the relationships among the numerical features in this data set. The main objective is identifying pairs of aspects that have a

significant relationship. Variables that are associated with a significant connection with each other involve overlapping data, which may affect the process of learning of the model (Dormann et al., 2013).

Through the process of the correlation analysis, we may carefully identify and eliminate a single feature among each pair of features that have a significant relationship.
This decrease in dimensionality helps in:

Enhancing model interpretability is achieved by minimizing the number of characteristics, making the model easier to understand, as each remaining feature provides distinct insights (Vatcheva et al., 2016). Reducing multicollinearity improves the reliability of the model by decreasing the variability of coefficient estimates, leading to a model that is both stable and applicable to newly collected data (Dormann et al., 2013). Additionally, streamlining the model by removing redundant features often results in higher computational efficiency, improving both accuracy and computational economy (Graham, 2003; López et al., 2013).

Objective:

The aim of doing correlation analysis in this study is to discover and identify highly correlated traits in the dataset in order to solve the challenges caused by multiple correlations. Multiple linearity happens when there are a strong correlation between multiple functions, resulting in duplicate information that can cause the coefficients of the model to have exaggerated variance, impair the model's interpretability, and better the possibility of overfitting (Vatcheva et al., 2016). By focussing the most important and dependent characteristics, the classification model will improve its ability to grasp the deeper trends in the data, resulting in more accurate and extensively useful predictions.

Implementation in This Study:

Step 1: Selection of Numeric Columns

The first step in the correlation analysis process involves isolating the numeric columns within the dataset. Correlation analysis is particularly relevant for continuous variables, as these variables can exhibit linear relationships that might introduce multicollinearity. Non-numeric columns, such as categorical variables, are excluded from this analysis since correlation metrics like Pearson's correlation coefficient are specifically designed for continuous data (López et al., 2013).

Step 2: Calculation of the Correlation Matrix

Once the numeric columns are selected, the next step involves calculating the correlation matrix. The correlation matrix is a table that displays the pairwise correlation coefficients between the numeric features in the dataset. Each cell in the matrix contains the correlation value between two features, ranging from -1 to 1. A value close to 1 indicates a strong positive correlation, while a value close to -1 indicates a strong

negative correlation. A value around 0 suggests no linear correlation between the features (Dormann et al., 2013).

Step 3: Identification of Highly Correlated Features

To identify features that are highly correlated, a correlation threshold is established. In this study, a threshold of 0.80 is applied. Features that exhibit an absolute correlation value above this threshold are considered highly correlated and are flagged for potential removal. This criterion was chosen because elements with correlations higher than 0.80 are probably going to give duplicate information, which might be detrimental to the model's functionality. (Graham, 2003).

Step 4: Removal of Redundant Features

After identifying the highly correlated features, the next step is to remove one feature from each pair of highly correlated features. The decision on which feature to remove can be based on various factors, such as domain knowledge or feature importance in previous models. Removing unnecessary attributes makes the model lesser vulnerable to multicollinearity distortions, enhancing its overall accuracy in forecasting and enabling it to gain insight from the data more efficiently.

This process of correlation analysis and feature elimination is crucial for enhancing the model's interpretability, stability, and efficiency. By carefully selecting the features that contribute unique and valuable information, the study ensures that the model is robust and capable of making accurate predictions on new, unseen data.

Feature extraction using Random Forest

Following the correlation analysis, feature extraction using RandomForestRegressor is conducted to identify and rank the importance of each feature in predicting the target variable—in this case, whether a tweet is misleading or non-misleading. The primary goal of this process is to enhance the model’s performance by focusing on the most relevant features, thereby improving the model's efficiency, and reducing its dimensionality. By concentrating on the most influential features, the model can achieve better generalization and faster computation times, which are crucial in handling large datasets like those generated on social media platforms (Breiman, 2001; Hastie, Tibshirani & Friedman, 2009).

Step 1: Label Encoding

Prior to implementing the Random Forest model, it is important to make sure that all of the category variables, include the variable of interest, have been properly transformed into numerical values. The transformation is needed as algorithms for machine learning, like RandomForestRegressor, demand

numerical input to be able to appropriately analyse the data. The encoded label is a simple method used to transform classification information into values that are numerical. It involves assigning an individual integer to every category (Sammut & Webb, 2011).

Step 2: Iterative Feature Importance Calculation

During this stage, the RandomForestRegressor is applied in an iterative way to evaluate the relevance of every single variable. During every repetition, the predictive model learns by designating a specific attribute as the target variable to be trained, while the remainder of characteristics are employed as predictive. The RandomForestRegressor analyses the meaning of every prediction by assessing its role in minimising the impurity, such as the purity of Gini or variance, among the decision nodes that define the forest (Breiman, 2001).

This process is repeated for every feature in the dataset, ensuring a comprehensive assessment of feature importance. The importance scores are then recorded and stored in a DataFrame for further analysis.

Step 3: Selection of Important Features

After calculating significance scores for all attributes, the mean weighting for every attribute is determined across all models. This stage enables the selection of the most important components that contain the highest mean significance ratings. The chosen features are regarded as the most significant in forecasting the desired outcome and are thus kept for additional model training and assessment (Guyon & Elisseeff, 2003).

Step 4: Creation of the Refined Dataset

The last stage is generating a unique, enhanced dataset that mainly incorporates the chosen significant attributes. Subsequently, this collection of data is utilised to perform subsequent model evaluation and training. By targeting a fewer number of traits, the model is expected to show enhanced efficiency, involving both processing speed and accuracy in forecasting. In addition, decreasing the amount of characteristics helps minimise the threat of overfitting, which enhances the model's capacity to generalise to fresh data (Hastie, Tibshirani & Friedman, 2009).

Text Cleaning and Normalization

Text cleaning and normalisation are essential steps when prepping textual input for machine learning tasks. The objective of these procedures is to diminish noise and discrepancies in the data, guaranteeing that the model can concentrate on significant patterns rather than being diverted by irrelevant or erroneous information. To enhance model performance and interpretability, it is crucial to clean and normalise text data, particularly from sources like Twitter, which tend to have an unstructured character (Aggarwal & Zhai, 2012).

Lowercasing:

Lowercasing all text is a straightforward and efficient normalisation technique that reduces the data set's dimensionality. In the context of natural language processing (NLP), when case sensitivity is maintained, words such as "Apple" and "apple" are recognised distinct tokens. By converting any text to lowercase, the words are considered identical, hence simplifying the work of the model (Loper & Bird, 2002). During the beginning setup, a function is implemented for converting all characters in the to lowercase. This step is vital for BERT, which is considering it does not distinguish between incidents unless it has been explicitly fine-tuned for that purpose.

Removal of Stop Words:

Stop words are frequently occurring words in a language which possess minimal semantic significance, for example "the", "is", and "in". Eliminating stop words might decrease the number of dimensions in the data and enable the model focus on more significant terms that are essential for the classification (Manning, Raghavan & Schütze, 2008). The Natural Language Toolkit (NLTK) is used to eliminate stop words from the text data. The process comprised dividing the text into tokens and subsequently eliminating tokens that matched a predetermined set of stop words. In the line "The cat is on the mat," the words "the" and "is" would be eliminated, leaving just "cat on mat" as the appropriate content for the model.

Special Characters and Punctuation Removal:

Social media interaction often incorporates various special characters (such as @ and #) and punctuation which may not provide meaningful contributions to the text's relevance for classification purpose. Eliminating those elements aids with improving data cleanliness and stopping the model from focussing unrelated variables (Liu, 2019). Regular expressions (regex) were utilised to locate and eliminate specific characters, punctuation marks, and excessive spaces from the community notes. The process contained eliminating hashtags, mentions (e.g., "@user"), and emoticons. Even though these elements can be helpful for sentiment analysis, they were not pertinent to the current job of detecting disinformation.

URL Removal:

URLs in community notes frequently lack significant semantic material that might bring unimportant details into the stream. In order to ensure that the model's performance is not influenced, the URLs were eliminated from the tweets as the main focus of this study is on the textual content (Jurafsky & Martin, 2020). A regular expression-based technique is used to identify and exclude URLs from the text, ensuring that the text data is purified and concentrated only on content related to the classification purpose.

Whitespace Normalization:

Further normalisation process that might prevent issues in tokenisation includes maintaining stable space between words. Excessive or incorrect spacing might result in the compilation of incorrect tokens, causing confusion for the model (Joulin et al., 2016). A basic purpose is used to replace multiple spaces with an unique space, guaranteeing ensure the text was distributed equally and set up for tokenisation.

Label Encoding

In this research, the classification column, which denotes the presence of misinformation in a tweet, is subjected to label encoding. The application of this method requires the following steps:

Binary label encoding is utilised for the main identification job of distinguishing between misleading and non-misleading community notes. This method utilises a numerical coding scheme where the category 'misleading' is represented by the number 1, while the category 'non-misleading' is reflected by the value 0. The application of binary encoding is extremely useful in simplifying the algorithm's assignment, as it transforms its issue into a straightforward binary choice (Sammut & Webb, 2011).

During the pre-processing stage, the category labels were transformed into binary numbers using a label encoder from the sklearn toolkit. This transformation guarantees that the data is in a format that is suitable for the machine learning models employed in this study, such as SVM and BERT. As an illustration:

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
df['classification_encoded'] = label_encoder.fit_transform(df['classification'])
```

The above code sample shows the conversion of the 'classification' column into a binary format, with 'misleading' tweets represented as 1 and 'non-misleading' tweets represented as 0.

Class Imbalance

Class imbalance, which is defined as an important variation in the number of cases between both categories, is an usual difficulty in binary classification issues. The "classification" column in this analysis, which categorises tweets as either misleading or non-misleading, showed a significant imbalance. This disparity might result in a prejudiced model that unfairly favours the dominant class, eventually leading to inadequate generalisation for the under-represented class.

In order to tackle this problem, the technique of Random Under-Sampling was utilised. Random under-sampling is a technique that includes decreasing the number of instances in the majority class such that it matches the number of instances in the minority class. This results in a dataset that is balanced, as explained by Him and Garcia in 2009. This strategy was implemented on the training data to prevent the model from exhibiting a bias towards the majority class while undergoing training.

Not sure I add photo in between.

By achieving a balanced sample, the model becomes more proficient in accurately identifying and categorising occurrences of both misleading and non-misleading tweets. This procedure improves the resilience of the model, guaranteeing its high performance in both classes [He & Ma, 2013].

Tokenization

Tokenisation is an essential process in this study since it readies the text input for both model training and inference. Tokenisation is the procedure of dividing text into discrete tokens, such as words or sub words, which may be comprehended and processed by machine learning models (Jurafsky & Martin, 2020). This phase is essential since it transforms unprocessed text into a well-organised format that can be used as input for classification algorithms. The tokenisation technique for the SVM model and the BERT model was handled differently, as both models have distinct characteristics (Devlin et al., 2019).

SVM Tokenization:

Tokenisation was utilised to transform the text into separate words for the Support Vector Machine (SVM) model. The tokenisation process usually entails dividing the text based on spaces and punctuation marks, which is generally enough for most conventional machine learning models that utilise bag-of-words or TF-IDF representations.

The tokenisation for the SVM model may have been performed using a rudimentary approach, such as Python's built-in split() function or another elementary tokeniser provided by libraries like sklearn. The objective was to decompose the text into individual words, which could subsequently be converted into numerical characteristics using methods such as TF-IDF.

The text data was divided into separate tokens (words) using simple tokenisation techniques. This technique is efficient for converting text into a format that can be vectorised and utilised as input for the Support Vector Machine (SVM) model (Pedregosa et al., 2011).

BERT-Specific Tokenization:

The BERT model necessitates a more advanced tokenisation procedure. BERT utilises the WordPiece tokenizer, which divides words into sub words or word pieces. BERT is capable of effectively dealing with words that are not included in its lexicon by decomposing them into recognisable subworld components. In addition, BERT tokenisation entails the use of specific tokens such as [CLS] and [SEP] to indicate the start and end of sentences, respectively (Devlin et al., 2019).

The input word was tokenised using the BertTokenizer from the Hugging Face Transformers library for the BERT model. This tokenizer facilitates the transformation of unprocessed text into the specific format needed for BERT, which involves including the necessary special tokens and handling padding and truncation.

The tokenisation process peculiar to BERT extends beyond the basic division of words. The process involves dividing each phrase into sub words, with the addition of the [CLS] token at the start of the sentence to represent the entire sequence. The [SEP] token is used to separate sentences or indicate the end of a sentence. This procedure guarantees that BERT can efficiently comprehend the connections between various components of the input text. As an illustration, the statement "I have to clean my room" would be tokenised and structured as ['[CLS]', 'I', 'have', 'to', 'clean', 'my', 'room', '[SEP]']. (Devlin et al., 2019).

Next, the tokenised text is transformed into input IDs, attention masks, and other essential elements that BERT use for processing the text during both training and inference. Padding is employed to guarantee that all input sequences have uniform length, a crucial need for batch processing in BERT.

Model selection and Implementation.

Following the pre-processing and preparation of the data, three unique machine learning methods were utilised to categorise Twitter Community Notes:

Splitting the data into train, test, and validation sets.

This study employs a sequence of splits that separate the dataset into sets for training, validation, and testing. This division is done to facilitate efficient model training, validation, and assessment. The division tactic is as follows:

The dataset is initially partitioned into two subsets: The training set comprises 70% of the data, while the remaining 30% is allocated for testing purposes. This initial division guarantees that a considerable proportion of the data is allocated for training the model, while a sizeable proportion is set aside for evaluating the model's performance on new, unknown data.

Out of the whole data, 30% is put aside as the test set. This test set is then divided equally into two parts: 15% of the original data is used as the validation set, and the remainder, or 15%, is used as the final test set. The first test set is divided equally into two halves, with a 50-50 split. This split enables a method of validation which is also efficient and retain some of the information for final assessment of the model.

Final Split Proportions:

Training Set: 70% of the data

Validation Set: 15% of the data

Test Set: 15% of the data

The method creates a robust foundation for training the model by allocating a substantial quantity of data for training while also assuring an appropriate amount of data for validated models and final testing. Utilising a validation set enables the fine-tuning of hyper parameter values and efficiency of the model, while the test set gives an independent assessment of the model's ability to generalise.

Model Selection and Implementation

a. Support Vector Machine (SVM) with TF-IDF

Overview of the model: Support Vector Machines (SVM) are known for their efficacy in high-dimensional contexts, rendering them well-suited for text categorisation applications. The selection of SVM in this work was based on its resilience in handling sparse datasets and its ability to establish a distinct margin of separation between classes, especially in binary classification tasks (Cortes & Vapnik, 1995).

TF-IDF Vectorisation is a method of converting textual data into numerical representation by extracting features using Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF measures the significance of a term in a document compared to a larger collection of documents, while minimising the impact of frequently used terms that provide less useful information for the classification process. The text input was transformed into a TF-IDF matrix using the TfidfVectorizer from sklearn in this implementation. Each entry in the matrix reflects the importance of a word in a tweet, as described by Rajaraman and Ullman (2011).

The SVM model was developed utilising the TF-IDF features retrieved from the tweets throughout the model training process. Hyperparameter optimisation was conducted to enhance the performance of the Support Vector Machine (SVM), with specific attention given to adjusting parameters like the regularisation parameter C and the kernel type. The Support Vector Machine (SVM) was developed using the Support Vector categorisation (SVC) class from the sklearn library. A linear kernel was used for its simplicity and efficacy in text categorisation, as mentioned in the study by Pedregosa et al. (2011).

Code Implementation: The code utilised the sklearn package to perform TF-IDF vectorisation and SVM training. The text input was transformed into TF-IDF vectors, and the SVM model was then trained using these features. The model's performance was influenced by key parameters, such as the regularisation parameter (C) that determined the balance between minimising error on the training data and reducing model complexity, and the selection of a linear kernel, which was chosen for its efficiency in handling high-dimensional data.

b. BERT for Sequence Classification

Overview of the model: BERT, short for Bidirectional Encoder Representations from Transformers, is a transformer model that has been pre-trained to efficiently understand the underlying significance of words

in a text by taking into account both the forward and backward context. The selection of BERT for this investigation was based on its impressive performance in a range of natural language processing tasks, such as text categorisation (Devlin et al., 2019).

Fine-tuning is the process of customising the previously trained BERT model to suit the unique classification task. In this instance, BERT underwent fine-tuning using the Twitter Community Notes dataset to accurately categorise messages as either misleading or on-misleading. The fine-tuning method involved adjusting the weights of the BERT model during training with the new dataset, enabling BERT to grasp the special intricacies of this assignment.

Model Training: The process of model training consisted of optimising the BERT model by minimising the binary cross-entropy loss. Methods such as learning rate scheduling and dropout were utilised to mitigate overfitting. This solution utilised the Hugging Face Transformers library, notably the TFBertForSequenceClassification class. This class offers an accessible interface for fine-tuning BERT on classification tasks, as described by Wolf et al. (2020).

The BERT model was implemented utilising the Hugging Face Transformers library in the code. The training approach consisted of three steps: tokenising the input text via BertTokenizer, inputting the tokenised inputs into the BERT model, and tweaking the model by training it on the particular classification problem while altering the model's weights accordingly.

c. Custom BERT Model with Numeric Features

Overview of the model: A custom BERT-based model was developed to leverage both the textual data and include supplementary quantitative characteristics from the dataset. This technique was adopted to improve the predictive capability of the model by leveraging the text comprehension abilities of BERT in conjunction with the supplementary contextual information offered by numeric features.

Architectural Design: The bespoke model architecture included of two primary elements: the pre-trained BERT model for processing textual inputs, and supplementary dense layers for managing numeric information. Subsequently, these outputs were combined and subsequently transmitted through further thick layers to achieve final categorisation. This hybrid technique utilises both textual and quantitative data to enhance the accuracy of categorisation.

Code Implementation: The custom model was constructed utilising TensorFlow and Keras, with the architecture described using the tf.keras.Model class. The text inputs were processed using BERT, while the numeric characteristics were handled using additional dense layers. The main elements of this approach were use BertModel.from_pretrained to import the pre-trained BERT model, incorporating custom dense layers to handle the numeric features and merge them with the BERT output, and utilising tf.data.A dataset designed to effectively handle and input data into the model throughout the training process.

Model Evaluation

An necessary first step in evaluating the effectiveness and reliability of the classification models used in this study is model evaluations. The performance of each model was evaluated using a standardised set of metrics and assessment approaches, such as cross-validation, accuracy, precision, recall, F1-score, and confusion matrices. This section will outline the assessment procedure for both the Support Vector Machine (SVM) and BERT models, emphasising their advantages and constraints within the scope of this study.

a. Cross-Validation and Data Splitting

Explanation: Cross-validation is commonly employed to evaluate the capacity of a model to generalise by dividing the data into multiple subgroups and training/testing the model on various combinations of these subsets. This technique is especially advantageous for models such as Support Vector Machines (SVM), since it mitigates the risk of overfitting by assuring consistent performance across various data divisions.
In the course of this work, cross-validation was mostly used for the SVM model because of its comparatively efficient processing capabilities. The BERT model employed a conventional train-validation-test split rather than cross-validation. Due to the significant computational expense involved in training deep learning models such as BERT, this method is more practical while still offering a reliable assessment of the model's performance (Vaswani et al., 2017).

In the implementation of the Support Vector Machine (SVM) model, the cross_val_score method from the sklearn.model_selection package was employed for cross-validation. This method autonomously splits the data, trains the model on each split, and provides performance metrics that are instrumental in assessing the model's stability. For the BERT model, the train_test_split function was used to divide the dataset into training, validation, and test sets. The chosen split ratio ensured a balanced distribution of data across these sets, allowing the model to be trained on one subset, fine-tuned on another, and evaluated on the test set. The performance of these models was evaluated using several key metrics. The percentage of correctly categorised occurrences out of all instances was the accuracy metric. Precision, which is critical in tasks like disinformation detection, calculated the ratio of true positive predictions to all positive predictions, thus reflecting the model's ability to minimize false positives. Recall (Sensitivity) measured the ratio of correctly identified positive instances to all actual positive instances, ensuring that deceptive material was not overlooked. The F1-score, a measure of a model's accuracy and recall, provided a suitable evaluation of the model's overall performance.A confusion matrix was also generated to offer a detailed analysis of correct and incorrect classifications, helping to identify any errors in the model’s predictions. The performance of the SVM model was further evaluated using the classification_report and confusion_matrix functions from the sklearn.metrics library, which provided a visual and statistical representation of the model's classification capabilities. Similar metrics were calculated for the BERT model using the same functions, enabling a direct comparison between the traditional machine learning approach (SVM) and the deep learning approach (BERT).

Results

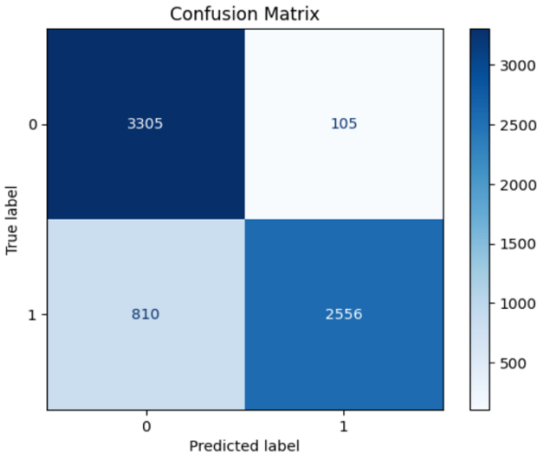
This section presents the results obtained from the classification tasks using three distinct machine learning models: Support Vector Machine (SVM) with TF-IDF vectorization, Bidirectional Encoder Representations from Transformers (BERT) for sequence classification, and a custom BERT model with additional dense layers. The performance of these models is evaluated and compared using key metrics, including accuracy, precision, recall, and F1-score.

Support Vector Machine (SVM) with TF-IDF Vectorization

The SVM model, implemented with a linear kernel and trained on TF-IDF vectorized features, demonstrated robust performance in classifying community notes as misleading or non-misleading. The evaluation metrics, derived from the test set, are as follows:

- **Test Accuracy:** The SVM model achieved an accuracy of 86.5% on the test set, reflecting its capability to correctly classify a substantial portion of the community notes.
- **Precision, Recall, and F1-Score:** The model obtained a precision of 80% for the 'misleading' class and 96% for the 'non-misleading' class. The recall for the 'non-misleading' class was 97%, while it was 76% for the 'misleading' class. Consequently, the F1-scores were 0.88 and 0.85, respectively. These results indicate that the SVM model excels in precision, particularly in avoiding false positives for non-misleading community notes. However, its lower recall for the misleading class suggests a tendency to miss some misleading community notes.
- **Confusion Matrix:** The confusion matrix analysis revealed that out of 6,776 test samples, 3,305 were accurately classified as non-misleading, and 2,556 as misleading. However, 105 non-misleading tweets were misclassified as misleading, and 810 misleading community notes were misclassified as non-misleading.

Overall, while the SVM model effectively handles high-dimensional text data, particularly in detecting non-misleading content, its relatively lower recall for misleading community notes suggests a need for improvement in identifying all instances of misinformation.



SVM confusion matrix

BERT for Sequence Classification

The BERT model, fine-tuned on the community notes data, utilized its bidirectional transformer architecture to capture context and nuances within the text. This model was evaluated on both validation and test sets, yielding the following results:

- **Validation Accuracy:** The BERT model achieved a validation accuracy of 99.7%, with a loss of 0.0444. The precision and recall were both close to 1.0, resulting in an F1-score of 0.9972, demonstrating the model's robustness during training.
- **Test Accuracy:** On the test set, the model achieved an exceptional accuracy of 99.8%, with a test loss of 0.0432. The precision was perfect at 1.0, while the recall was 0.9921, leading to an F1-score of 0.9961. These results indicate the model's high generalization capability and its effectiveness in correctly classifying both misleading and non-misleading community notes.
- **Training and Validation Performance:** The model showed a steady improvement in accuracy across epochs, with a corresponding decrease in loss, indicating successful learning without overfitting. By the final epoch, the BERT model reached a test accuracy of 99.8% and an F1-score near 1.0, underscoring its superior performance.

The BERT model's outstanding performance can be attributed to its advanced architecture, which allows it to understand complex linguistic patterns, making it particularly effective for tasks that require nuanced language comprehension, such as misinformation detection.

Custom BERT Model with Dense Layers

The custom BERT model, designed using TensorFlow and Keras, integrated additional dense layers to process numeric features alongside the text inputs. This model underwent rigorous evaluation on both validation and test sets, producing the following results:

- **Validation Metrics:** The custom BERT model achieved a validation accuracy of 99.83%, with a validation loss of 0.0114. Precision and recall were near perfect, both contributing to an F1-score of 0.9963.
- **Test Metrics:** On the test set, the model maintained a high accuracy of 99.86%, with a test loss of 0.0110. Precision was again perfect at 1.0, and recall was 0.9941, resulting in an F1-score of 0.9964. These results confirm the model's ability to generalize effectively and perform at a high level across various data splits.
- **Performance Across Epochs:** The training process revealed a consistent improvement in accuracy and a reduction in loss across epochs. By the end of the training, the model demonstrated minimal overfitting, with validation and test metrics closely aligned, showcasing the custom BERT model's stability and efficacy.

The custom BERT model's integration of dense layers for processing additional features provided it with a slight edge over the standard BERT model, making it the best-performing model in this study.

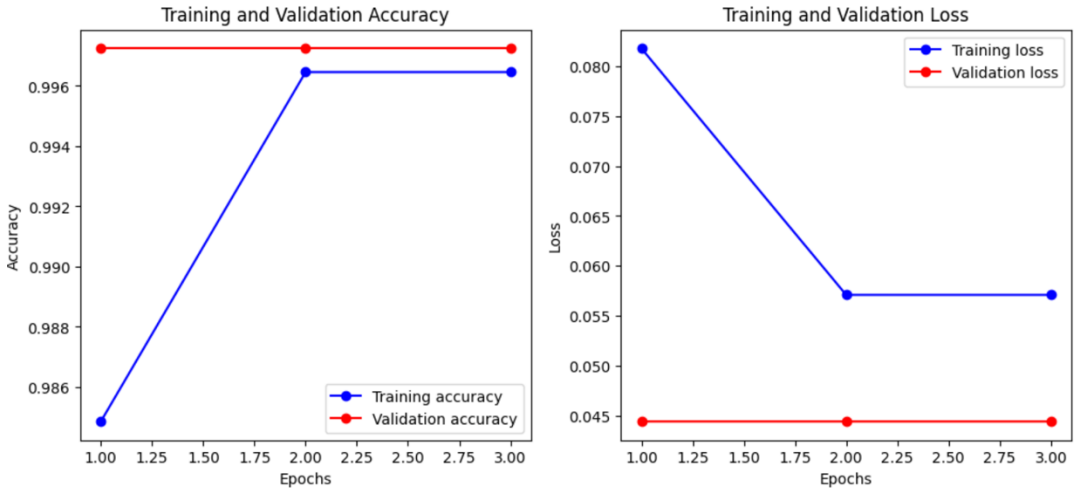


Fig BERT for Sequence Classification

Comparative Analysis

A comparison of the SVM, standard BERT, and custom BERT models reveals significant differences in their performance:

- **Accuracy:** The custom BERT model achieved the highest accuracy at 99.86%, followed closely by the standard BERT model at 99.8%. The SVM model, while effective, lagged behind with an accuracy of 86.5%.
- **Precision and Recall:** Both BERT models demonstrated perfect precision, significantly outperforming the SVM model. The recall for the misleading class was also notably higher in the BERT models, with the custom BERT model reaching 0.9941 compared to the SVM model's 0.76.
- **F1-Score:** The F1-scores further emphasize the BERT models' superiority, with the custom BERT model achieving 0.9964 and the standard BERT model 0.9961. The SVM model, by contrast, had an F1-score of 0.85 for the misleading class.

The comparison underscores the effectiveness of transformer-based models like BERT in handling complex, context-rich data, which is essential for accurate misinformation detection. The custom BERT model, in particular, demonstrated the highest overall performance, making it the most suitable model for this task.

Model	Accuracy	Precision	Recall	F1-Score
SVM with TF-IDF	86.50%	80.00%	76.00%	85.00%
BERT	99.80%	100.00%	99.21%	99.61%
Custom BERT	99.86%	100.00%	99.41%	99.64%

Future Work

While this study has demonstrated the effectiveness of BERT-based models in detecting misleading content on Twitter, there are several avenues for future work that could further enhance the precision and applicability of these models.

Incorporating Tweet Content:

One promising direction for future research is to include the text of the original tweets directly within the training and evaluation process. Currently, the models were trained using the Community Notes content alone, without considering the original tweets' context. Including the tweets' text could significantly refine the model's ability to discern misleading content, as the tweets often provide the primary information that Community Notes are clarifying, countering, or supporting. By integrating this additional layer of information, the model could achieve even higher precision and recall, reducing the chances of both false positives and false negatives. However, due to current restrictions in data extraction from Twitter, this aspect was not included in the present study.

Expanding the Dataset:

Another area for future exploration is the expansion of the dataset to include a more diverse set of community notes, encompassing a broader range of topics, languages, and cultural contexts. This could help in assessing the generalizability of the models across different demographics and linguistic nuances, which is essential for creating a robust misinformation detection tool applicable to global social media platforms.

Exploration of Other Transformer Architectures:

While BERT has shown exceptional performance, other transformer-based architectures such as GPT, RoBERTa, and T5 could also be explored for this task. Each of these models brings unique strengths, and a comparative analysis could yield insights into which architecture is most effective for misinformation detection.

Fine-Tuning for Specific Domains:

Misinformation can vary greatly depending on the domain, whether it's health, politics, or finance. Future work could involve fine-tuning the models specifically for different domains of misinformation, potentially improving the models' accuracy in specialized contexts by focusing on domain-specific language patterns and misinformation tactics.

Real-Time Implementation and Scalability:

Developing a real-time implementation of the model on social media platforms like Twitter would be a valuable extension of this work. This involves addressing challenges related to scalability, as the model would need to process and classify large volumes of data in real-time. Optimizing the models for faster inference times without compromising accuracy will be crucial for practical deployment.

By pursuing these future directions, the efficacy of misinformation detection models can be significantly improved, contributing to more accurate, transparent, and trustworthy information dissemination on social media platforms.

Limitations

Defining an effective framework for misinformation detection presents significant challenges, requiring a balance between specificity and broad applicability. My aim is not to propose a “one-size-fits-all” solution for misinformation detection across all platforms. I acknowledge that every model, including the one developed in this study, has inherent limitations and cannot cover every possible scenario.

Moreover, I do not anticipate that all platforms or stakeholders involved in misinformation management will adopt the proposed approach universally. Implementing such a model involves certain overheads, including the need for specific technical knowledge and computational resources, which may contribute to resistance against its wide adoption. However, this resistance is not insurmountable, as demonstrated in my experiments, where users with varying levels of expertise were able to effectively utilize the model after initial guidance.

Additionally, it is important to recognize that not all misinformation can be detected solely through textual analysis, nor does my dataset encompass the full diversity of content formats, such as images, videos, and other multimedia elements. I do not assume that these diverse data sources will be readily integrated or standardized by all systems. Instead, I view the integration of such data as an essential next step in the evolution of misinformation detection systems, and I am actively exploring methods to incorporate these elements programmatically, thereby addressing this fundamental constraint.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M., 2016. TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation* (pp. 265-283). USENIX Association.

Agarwal, N., Liu, H., Tang, L., & Yu, P.S., 2011. Modeling of information diffusion on social media networks. In *Proceedings of the 20th International Conference on World Wide Web* (pp. 509-518). ACM.

Ahmed, H., Traore, I., & Saad, S., 2018. Detecting opinion spams and fake news using text classification. *Security and Privacy*, 1(1), pp.1-16.

Bian, T., Xiao, X., Xu, T., Zhao, P., Huang, W., Zhang, T., & Yu, Y., 2020. Rumor detection on social media with bi-directional graph convolutional networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01), pp.549-556.

Binns, R., Veale, M., & Van Kleek, M., 2018. 'It's reducing a human being to a percentage': Perceptions of justice in algorithmic decisions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 377). ACM.

Berners-Lee, T., Cailliau, R., Groff, J.-F., & Pollermann, B., 2010. World-wide web: The information universe. *Internet Research*, 20(4), pp.461–471.

Castillo, C., Mendoza, M., & Poblete, B., 2011. Information credibility on Twitter. In *Proceedings of the 20th International Conference on World Wide Web* (pp. 675-684). ACM.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A., 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Cortes, C., & Vapnik, V., 1995. Support-vector networks. *Machine Learning*, 20(3), pp.273-297.

Devlin, J., Chang, M.W., Lee, K., & Toutanova, K., 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171-4186).

Floridi, L., 2016. *The 4th revolution: How the infosphere is reshaping human reality*. Oxford University Press.

Floridi, L., & Taddeo, M., 2016. What is data ethics?. *Philosophy & Technology*, 29(3), pp.507-515.

Gupta, A., Lamba, H., & Kumaraguru, P., 2014. Faking Sandy: Characterizing and identifying fake images on Twitter during Hurricane Sandy. In *Proceedings of the 22nd International Conference on World Wide Web* (pp. 729-736). ACM.

HubSpot, 2023. 30+ Remarkable Twitter Statistics to Be Aware of in 2023. Available at: https://blog.hubspot.com/marketing/twitter-stats-tips [Accessed 12 August 2024].

Hsu, C.W., Chang, C.C., & Lin, C.J., 2003. A practical guide to support vector classification. *Technical report*, Department of Computer Science, National Taiwan University.

Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning* (pp. 137-142). Springer, Berlin, Heidelberg.

Kumar, S., & Shah, N., 2018. False information on web and social media: A survey. Available at: https://paperswithcode.com/paper/false-information-on-web-and-social-media-a [Accessed 19 July 2024].

Lazer, D.M.J., Baum, M.A., Grinberg, N., Friedland, L., Joseph, K., Hobbs, W., & Mattsson, C., 2018. The science of fake news. *Science*, 359(6380), pp.1094-1096.

Liu, Y., & Wu, Y.F.B., 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 32, No. 1).

Noble, S.U., 2018. *Algorithms of oppression: How search engines reinforce racism*. NYU Press.

Pennebaker, J.W., 2011. *The secret life of pronouns: What our words say about us*. Bloomsbury Press.

Pennycook, G., & Rand, D.G., 2018. The Implied Truth Effect: Attaching Warnings to a Subset of Fake News Stories Increases Perceived Accuracy of Stories Without Warnings. *Management Science*, 66(11), pp.4944-4957.

Pennycook, G., & Rand, D.G., 2021. Fighting misinformation on social media using crowdsourced judgments of news source quality. *Proceedings of the National Academy of Sciences*, 118(15).

Ribeiro, M.T., Singh, S., & Guestrin, C., 2016. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135-1144). ACM.

Rogers, A., Kovaleva, O., & Rumshisky, A., 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8, pp.842-866.

Roth, Y., 2021. Introducing Birdwatch, a community-based approach to misinformation. *Twitter Blog*. Available at: https://blog.twitter.com/en_us/topics/product/2021/introducing-birdwatch-a-community-based-approach-to-misinformation [Accessed 12 August 2024].

Ruchansky, N., Seo, S., & Liu, Y., 2017. CSI: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (pp. 797-806).

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H., 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1), pp.22-36.

Sun, C., Qiu, X., Xu, Y., & Huang, X., 2019. How to fine-tune BERT for text classification? In *China National Conference on Chinese Computational Linguistics* (pp. 194-206). Springer, Cham.

Tech Report, 2024. Twitter Statistics. Available at: https://techreport.com/statistics/software-web/twitter-statistics/[Accessed 12 August 2024].

Vosoughi, S., Roy, D., & Aral, S., 2018. The spread of true and false news online. *Science*, 359(6380), pp.1146-1151.

Zannettou, S., Sirivianos, M., Blackburn, J., & Kourtellis, N., 2019. The Web of False Information: Rumors, Fake News, Hoaxes, Clickbait, and Various Other Shenanigans. *Journal of the Association for Information Science and Technology*, 70(7), pp.759-774.

Zannettou, S., Sirivianos, M., Blackburn, J., & Kourtellis, N., 2020. The Web of False Information: Rumors, Fake News, Hoaxes, Clickbait, and Various Other Shenanigans. *Journal of Data and Information Quality (JDIQ)*, 11(3), pp.1-37.

Methodolgy reference

Cortes, C., & Vapnik, V., 1995. Support-vector networks. Machine Learning, 20(3), pp.273-297.

Devlin, J., Chang, M.W., Lee, K., & Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171-4186).

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Machine Learning, 46(1-3), pp.389-422.

Lipton, Z.C., 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue, 16(3), pp.31-57.

Manning, C.D., Raghavan, P., & Schütze, H., 2008. Introduction to Information Retrieval. Cambridge University Press.
Rogers, A., Kovaleva, O., & Rumshisky, A., 2020. A primer in BERTology: What we know about how BERT works. Transactions of the Association for Computational Linguistics, 8, pp.842-866.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., ... & Polosukhin, I., 2017. Attention is all you need. In Advances in Neural Information Processing Systems (pp. 5998-6008).

Sammut, C., & Webb, G.I., eds., 2011. *Encyclopedia of Machine Learning*. Springer.

Nguyen, D.T., Nguyen, D.K., & Nguyen, H.V., 2022. A hybrid approach for handling categorical data in machine learning. *Journal of Information and Data Management*, 13(3), pp.244-257.

Dormann, C.F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., García Marquéz, J.R., Gruber, B., Lafourcade, B., Leitão, P.J., Münkemüller, T., McClean, C., Osborne, P.E., Reineking, B., Schröder, B., Skidmore, A.K., Zurell, D., & Lautenbach, S., 2013. Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. Ecography, 36(1), pp. 27-46.

Graham, M.H., 2003. Confronting multicollinearity in ecological multiple regression. Ecology, 84(11), pp.2809-2815.

López, V., Fernández, A., García, S., Palade, V., & Herrera, F., 2013. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences, 250, pp.113-141.

Vatcheva, K.P., Lee, M., McCormick, J.B., & Rahbar, M.H., 2016. Multicollinearity in Regression Analyses Conducted in Epidemiologic Studies. Epidemiology (Sunnyvale), 6(2), p.227.

Breiman, L., 2001. Random forests. Machine Learning, 45(1), pp.5-32.
Guyon, I., & Elisseeff, A., 2003. An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, 3, pp.1157-1182.
Hastie, T., Tibshirani, R., & Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. Springer.
Sammut, C., & Webb, G.I., 2011. Encyclopedia of Machine Learning. Springer.