IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

# Unravelling the Black Box: Explainable AI Approaches for Robust Hate Speech Detection

**Dipti Mittal [1], Harmeet Singh [2], Sita Rani [3]**

[1] Research Scholar, Department of Computer Applications, CT University, Punjab, India.
[2] Assistant Professor, Department of Computer Applications, CT University, Punjab, India.

[3] Department of Computer Science and Engineering, Guru Nanak Dev Engineering College,

Ludhiana-141006, Punjab, India

**ABSTRACT**

The potential of flexible and multifaceted characteristics in hate speech detection by deep learning models is found within Explainable Artificial Intelligence (XAI). The goal of this research was to comprehend the decision-making process through interpreting and explaining complex AI model decisions. Two datasets were chosen for demonstrating XAI's implementation into detecting cases of hate speech, which underwent data preprocessing with steps such as text cleaning, tokenization, lemmatization etc., followed by categorically simplifying them for training purposes. Exploratory analysis conducted on said dataset revealed patterns and insights that aided several pre-existing models from Google Jigsaw' including Decision Trees, K-Nearest Neighbors Multinomial Naïve Bayes Random Forest Logistic Regression Long Short-Term Memory among others where LSTM achieved an incredible accuracy rate at 97.6%. For explainability techniques like LIME or Local Interpretable Model Agnostic Explanations can be utilized upon using the HateXplain dataset while Variants were built atop BERT(Bidirectional Encoder Representations From Transformers) called BERT+ANN(Artificial Neural Network) with a result yieldting 93.l55% Accuracy Alongside By Using Benchmark ERASER(Evaluating Rationales And Simple English Reasoning), another variant labeled BETT + MLP(Multilayer Perceptron) yielded impressive results up to 93 .67 % accuracy performance metrics standardized provide good performance.

***Keywords:*** explainable artificial intelligence; hate speech detection; offensive languages; LIME; BERT; neural networks

## Introduction

The application of artificial intelligence is widespread in various fields, including science, education, finance and business. However, it's currently limited to its subset known as "machine learning" and has yet to reach its full potential. Machine learning enables computers to learn the relationship between input and output without requiring explicit programming using algorithms from previously provided data sets that can be used for predictive modelling with new datasets. Unfortunately, their ability comes up short when needing outcomes explained which traditional AI does not provide insight into how different features contribute results but rather a black box-type function making explainable AI emerging topic (XAI) provides more answers along with outputs enabling reasoning on human terms thereby having found use across diverse industries as an area of study."

The functioning of artificial intelligence is often concealed within a "black box", providing only the output without divulging its methodology. While in many scenarios, understanding the reasoning behind such outputs may not be crucial, it becomes critical in certain fields like medical research where answers to "how" and "why" are vital. The absence of knowledge pertaining to instances when models fail or succeed could lead to severe repercussions by failing to detect errors or rectify them appropriately. This might also raise doubts about the model's effectiveness altogether

### Need for Explainability

Explainable AI (XAI) is crucial in enabling users to comprehend the results produced by artificial intelligence, establish trust within algorithmic

decision-making and maintain an organized approach towards managing these outcomes. Regulatory considerations as well as ethical concerns carry significant importance for incorporating AI into everyday human interactions. XAI serves a fundamental role in fostering confidence amongst regulators and business partners using commercial benefits alongside ethically sound decisions creating better foundations for responsible practices especially concerning pivotal scenarios such as medical research. Implementing explanations regarding how models generate insights bolsters reasoning capabilities augmented with humans' cognitive abilities leading to improved quality of applications producing effective outcomes which minimize errors from arising without warning or scope when possible. It represents cutting-edge technology that provides novel solutions capable of addressing "why" type questions beyond what conventional methods can answer definitively; varied fields including health care systems along legal operations like law enforcement agencies all benefit immensely due to this emerging field's broad-based implication across diverse sectors ahead of its time in both theory & practice throughout forthcoming innovations shaping our world soon enough!

## Motivation

Artificial intelligence functions as a mysterious “black box” that produces output based on input without revealing its inner workings. Although machine learning has found practical uses in industries such as medicine, research, business, education and transportation (including self-driving cars), some of the models’ lack of clarity may make them difficult to comprehend or less effective altogether. Recently developed deep learning models have shown promising results but are still unable to explain their decisions accurately all the time—thus necessitating methods for eXplainable Artificial Intelligence (XAI) where explanations can be interpreted by humans with little knowledge required about how deep learning works. In particular, XAI lends itself well towards facilitating hate speech detection through deep neural networks. As these complex algorithms become more elaborate due to added parameters and optimizations over iterations making it even harder than usual validate model outputs precisely against real-life understandings expecting from such AI frameworks..

The aim of this paper is to gain insight into the decision-making process of complex artificial intelligence (AI) models that detect hate speech, and explain their decisions. To achieve this, pre-existing AI models were applied on Google Jigsaw dataset with a focus on improving prediction accuracy while using explainable methods such as LIME for interpreting results from HateXplain dataset. Additionally, variations of BERT model including BERT + ANN and BERT + MLP were developed specifically to optimize performance in terms of comprehensibility by utilizing ERASER benchmark (DeYoung et al., 2019).

## Literature Review

Recent studies have investigated hate speech detection through both traditional natural language processing (NLP) techniques and machine learning methods [1-3]. Success has been found in identifying bullies by extracting text-, user-, and network-based features and characteristics [4]. Additionally, deep learning techniques have been used to study abusive language detection, including identification of keywords commonly associated with hate speech, sexism, bullying, trolling, racism. These topics were explored in research papers such as those referenced in sources [1 and 5-7].

Recently, there has been a growing focus on explicating artificial intelligence methods such as machine learning and deep learning to comprehend their rationale in tagging text with hate speech or for other social media and medical purposes. A cutting-edge method of explanation founded upon LIME [3,8] was advanced [9], along with suggestions regarding the appropriate handling of these transparent machine-learning models and their use cases[10–14]. Further investigations into explanability were conducted through deep-learning techniques as well as active learning procedures documented in works such as [8,15–17].

Recently, there has been a surge in popularity for Explainable AI (XAI) as it aims to demystify the decision-making processes employed by artificial intelligence. With novel definitions introduced for explainable machine learning and deep learning [18], XAI techniques have been categorized based on factors such as their scope, methodology, algorithmic intuition and explanation capability [19]. A number of available models are discussed in literature including LIME, layer-wise relevance propagation and DeepLIFT alongside deployment strategies highlighted across various studies [20-23]. Furthermore,XAI applications span over industries like manufacturing where predictive maintenance scenarios are enabled through this technology[24]and even social science research[25].

**Table 1 Literature Review Summary**

| Ref. | Contribution | Key Findings | Limitation (s) |
|---|---|---|---|
| [1] | Automated hate speech detection and the problem of offensive language | Detection of offensive content and identification of potential offensive users. | The definition of hate speech is limited to language |

| | | | |
|---|---|---|---|
| | | Logistic regression, Naïve Bayes, decision trees, random forests, and SVM are tested using 5-fold cross-validation. | that threatens or incites violence, excluding a large proportion of hate speech. Lexical methods used are inaccurate at identifying hate speech. |
| [2] | A feature attribution method for explainability | Detecting bullying and aggressive behavior on Twitter. Random forest classifier using WEKA tool with 10-fold cross-validation. | The analysis is limited by the dataset's lack of variety of demographic groups. Results are presented only with respect to training time and performance due to limited space. Network-related metadata are not considered. |
| [3] | A unified deep learning architecture for abuse detection | Deep learning architecture for detection of abuse online. SHAP (Shapley Additive Explanations) framework for explaining complex ensemble and deep learning models. | SHAP model is not consistent with human intuition in some cases, leading to false positives or false negatives. Gradient-based sensitivity analysis used with this approach is not able to get accurate relevance scores when sentiment is decomposed into words. |
| [4] | LIME explanation with individual examples | LIME model to explain the predictions of any classifier. SP-LIME model for selecting representative and non-redundant explanations. | Some misclassification is observed in the case of non-toxic comments. |
| [5] | Explaining the predictions of any classifier | Technical foundations of explainable AI, presentation of practical XAI algorithms such as occlusion, integrated gradients, and LRP (Layer-wise Relevance Propagation), importance applications, challenges, and directions for future work. | The explanation revealed by the model in this research is difficult to interpret by a human observer due to limited accessibility of the data representation. |
| [6] | Interpretable machine learning models | Application-grounded, human-grounded, and functionally grounded approaches for evaluation of interpretability. | The research is focused only on the taxonomy to define and evaluate interpretability and not on methods to extract explanations. |
| [7] | Explainability of deep neural network models | Transparency of machine learning models, novel technological development for explainability, need for diverse metrics for | Only local explanations are presented, focusing on single samples without considering global |

| | | | |
|---|---|---|---|
| | | targeted explanations, suggestions for explainability of deep learning models. | explanations. |
| [8] | An active learning approach for labeling text | Attention network visualization for indirect and informal communication. Overview of explainable AI literature, review and taxonomy, implications, vision, and future of XAI. | The research does not focus on methods to evaluate natural language generation (NLG). Results are presented using a generic dataset, not real data. |
| [9] | Evaluation of explainable AI models for convolutional neural networks (CNN) | Proposed two proxy tasks (pattern task and Gaussian blot task) to evaluate LIME, layer-wise relevance propagation, and Deep LIFT, and discussed results. | The evaluation scheme has issues with cross-model evaluation and is less comprehensive. |
| [10] | Discussion of various explainable AI techniques | Survey on various XAI techniques and methodologies, need timeline applications, and future work of fuzzy systems for XAI. | The research fails to address the limitations of conventional AI and its combination with XAI. |
| [11] | Predictive maintenance case study based on explainable AI (XAI) | A machine learning model based on a highly efficient gradient boosting decision tree is proposed for the prediction of machine errors or any tool failure. | Results are presented using a generic dataset, not real data; however, the concept shows high maturity with promising results. |
| [12] | Insights from social sciences related to explainable artificial intelligence (XAI) | Why questions are diversified in explainable AI; explanations are biased and socially important. | Adopting the work of this research into explainable AI is not straightforward; models discussed need to be refined and extended to provide good exploratory agents. |
| [13] | Explainability of deep neural network models | Overview of interpretability of machine learning models. | The study only focuses on abstract overview of explainability without diving deep into explanation metrics. |
| [14] | Enhancing interpretability of tree-based machine learning models | Method for computation of the game theoretic Shapley values; local explanation method tools for explainability using a combination of local explanation methods. | Only local explanations are presented that focus on single samples without considering global explanations. |
| [15] | A unified framework for machine learning interpretability | An open-source package InterpretML for glass-box and black-box explainability. | Computational performance for models across datasets is not consistent for the explainable boosting machine (EBM) model discussed in |

| | | | |
|---|---|---|---|
| | | | this research. |
| [16] | An active learning approach for labeling text | Semantic embeddings and lexicon expansion techniques discussed. | The semantic embeddings and lexicon expansion techniques lack detailed explanations. |
| [17] | Explainable artificial intelligence (XAI): categorization, contributions, suggestions, and issues in responsible AI | Explainable AI methods give explanations that are not aligned with what the original method calculates. | Some functions are proprietary and are not exposed to the public in this research. |
| [18] | Opportunities and challenges in explainable artificial intelligence (XAI) | Human attention is not able to arrive at XAI explanation maps for decision-making. | Quantitative measures of completeness and correctness of the explanation map are not available. |
| [19] | Explainable artificial intelligence (XAI): categorization, contributions, suggestions, and issues in responsible AI | The explanation revealed by the model in this research is difficult to interpret by a human observer due to limited accessibility of the data representation. | The research does not focus on methods to evaluate natural language generation (NLG). |
| [20] | Discussion of various explainable AI techniques | The explanation revealed by the model in this research is difficult to interpret by a human observer due to limited accessibility of the data representation. | The research does not focus on methods to evaluate natural language generation (NLG). |
| [21] | Fuzzy systems for explainable artificial intelligence | Survey on various XAI techniques and methodologies, need timeline applications, and future work of fuzzy systems for XAI. | The research fails to address the limitations of convention al AI and its combinatio n with XAI. |
| [22] | A literature survey on explainable artificial intelligence (XAI) terminology | Background, terminology, objectives of explainable artificial intelligence (XAI), natural language generation approach. | The survey does not explain how to evaluate natural language generation (NLG). |
| [23] | Predictive maintenance case study based on explainable artificial intelligence (XAI) | A machine learning model based on a highly efficient gradient boosting decision tree is proposed for the prediction of machine errors or any tool failure. | Results are presented using a generic dataset, not real data; however, the concept shows high maturity with promising results. |
| [24] | Insights from social sciences related to explainable artificial intelligence (XAI) | Why questions are diversified in explainable AI; explanations are biased and socially important. | Adopting the work of this research into explainable AI is not straightfor ward; models discussed need to be refined and extended to provide good exploratory agents. |
| [25] | Evaluation of explainable AI models for convolutional neural networks (CNN) | Proposed two proxy tasks (pattern task and Gaussian blot task) to evaluate LIME, layer-wise relevance propagation, and Deep LIFT, and discussed results. | The evaluation scheme has issues with cross-model evaluation and is less comprehens ive. |

## Materials and Methods

In this section, we discuss the two datasets utilized for hate speech detection through explainable artificial intelligence. Both datasets exclusively contain text in English language. The Jigsaw dataset was employed to compare linear and complex models such as decision trees and LSTM on a hate speech dataset extracted from user discussions of talk pages of English Wikipedia provided by Google's Jigsaw platform. This particular set has been used to train various semi-interpretable linear models but does not feature human annotations making it unsuitable for evaluating against ERASER benchmarks unlike HateXplain - another dataset containing annotated posts sourced from Twitter and Gab that allows assessment across these metrics while retaining its suitability due to its accessible interpretability features based upon annotation guidance within each post analyzed throughout all stages during exploration so potential issues with bias can be adjusted accordingly if necessary at any point beforehand or afterward without impacting results produced when running queries using criteria specified either prior exposure/specific context were given about what constitutes "hate".

### Google Jigsaw Dataset

We utilized a dataset provided by Google Jigsaw for the initial portion of our analysis, which was released as part of a Kaggle competition. This dataset includes various fields such as comment, toxic, severe_toxic, obscene, threat insult and identity_hate. The discussions included in this specific data set are extracted from Wikipedia pages. Moreover,the labeling standards allow for instances where one text belongs to multiple classes simultaneously - also known as multinomial classification.The specifics regarding the Google Jigsaw data set can be further explored through Table 2 provided below

### HateXplain Dataset

The HateXplain dataset, comprising of posts from both Twitter and Gab, was utilized as the second data source. By merging these two platforms, a comprehensive compilation of over 20,000 instances containing labels that denote hateful content alongside offensive or plain text was achieved.

For our dataset, we collected 1% of tweets from January 2019 to June 2020 on Twitter through a random selection process. We obtained the Gab dataset mentioned in [26]. Reposts were excluded and duplicates were eliminated to only include textual data with significant emotional value contributed by emojis that remained untouched. All usernames have been redacted and replaced using tokens instead.

### Extracting the Dataset

The data collected was in CSV format, which stores tabular information as plain text separated by commas. Each line corresponds to a row and the first row contains attribute or column names. The files were loaded into a Pandas data frame using Python's Pandas library known for its extensive use in analyzing and manipulating datasets for machine learning and data science applications.

### Data Preprocessing and Cleaning

Data preprocessing is an essential phase that affects the effectiveness of a model. Data obtained from online sources such as Twitter are often contaminated with noise and void or incomplete values, including images, audio files and videos. Preprocessing guarantees data cleanliness by eliminating any unwanted information while retaining key details for meaningful analysis. It's worthy to note that no preprocessing was carried out on BERT-based models since they have been pre-trained language representation models which incorporate every piece of information in a sentence like punctuations and stop words. Nonetheless, Python libraries plus functions were utilized for cleaning followed by analyzing unrelated contents regarding this research project where we deployed various statistical learning approaches except those anchored on BERT technology too.

**A summary of the steps performed for preprocessing and cleaning of the dataset is given below.**

1. Rows with missing labels were dropped as they do not contribute to the learning process.
2. Using the natural language toolkit (NLTK) library, tokenization was performed, i.e., tokens of the sentences were created.
3. Stop words (if, then, the, and, etc.) were removed to keep only the text that would contribute to the learning process.

Before training the model, conducting data cleaning is crucial given its several advantages. Removing incorrect or inconsistent information enhances the quality of data as shown in Figure 1 where common steps are followed for effective cleansing. The process involves deleting unwanted observations and rectifying structural errors present within dataset entries. Structural error refers to discrepancies such as feature name misspelling, varying attribute names for one item, misclassifications etc., that occur due to irregular sentence structure including extra spaces and newline characters. To eliminate outliers like excess spacing and address missing fields in datasets; other subsequent actions follow thereafter stated comprehensively below [27].

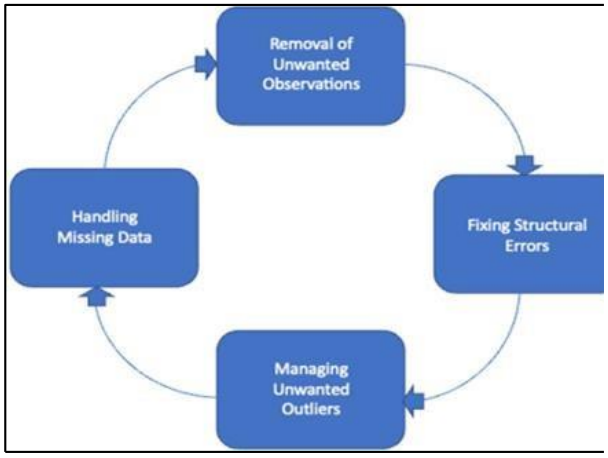


**Figure 1 Data cleaning.**

1. Firstly, a regular expressions module was imported to help with data cleaning tasks. Regular expressions are sequences of characters that are used for matching with other strings in search. Patterns and strings of characters can be searched using regular expressions. Python has a “re” module that can help to find patterns and strings using regular expressions. Regular expressions can be used to remove or replace certain characters as part of data cleaning and preprocessing.
2. Any newline characters or additional spaces were removed.
3. Any URLs were also removed as they do not contribute to the learning process.
4. Similarly, any other alphanumeric characters that included punctuation were removed for the same reason, including the following strings: !"#$%& ’()*+,-./:;<=>?@[ ]^_‘{|}~. Only uppercase and lowercase letters along with digits 0–9 were kept.
5. Stopwords such as “the”, “and”, “then”, and “if” were also removed as they are not a part of the learning process. Python’s NLTK library has stopwords in about 16 different languages. We imported English stopwords to remove them from our dataset. These words were removed as they do not add any additional information to the learning process.
6. The outputs of these tasks were stored in a separate column, resulting in a column of tokenized words.

### Tokenization, Sentence Padding, and Lemmatization

Tokenization involves dividing sentences into smaller parts referred to as tokens, which serve as the foundation for stemming and lemmatization. It can also assist in identifying patterns within text. The Python library NLTK offers functions designed specifically for word tokenization, with character or subword output options available. For instance, "clearer" could be broken down into either "clear" and "er," or even spelled out entirely (c-l-e-a-r-e-r). In order to improve efficiency during learning processes related to this study endeavored upon using character tokenization methods that convert words into integer-based arrays. To achieve our objectives we utilized a tokenizer object made from a pre-existing model imported via the TensorFlow libraries alongside Keras technologies fitted towards utilization of HateXplain Dataset analysis efforts.

To meet the requirement of same length inputs for neural networks, we applied padding to ensure uniformity. The initial raw text comprised varying lengths of words and sentences which were observed during exploratory data analysis. After scrutinizing maximum sentence length predominantly ranging up to 200, longer ones were cropped while shorter ones got padded.

Lemmatization was utilized for word normalization using natural language processing (NLP) to reduce all words to their base/root forms. This included reducing "go, going, gone, and goes" to "go," "read, reading, and reads" to "read," and "hated, hating,and hates"to"hate."

### Simplification of Categorical Values

To streamline the training and learning process, the original dataset's seven columns were reduced to three: text, category, and label. The "tweet" column was converted into a "text" column while deriving labels from values in hate_speech, offensive language, and neither categories. Consequently , 0 now represents hate_speech; 1 indicates offensive_language whereas 2 signifies neither . Thus resulting data is ideal for effective training & learning with only essential information contained within it including Text (actual message), Category (hate speech/offensive/neither) as well as Labels representing each of these respective groups numbered accordingly from their individual categories mentioned before - all represented across just three distinct columns overall!

### Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) involves investigating data to identify patterns and insights, which helps with understanding the dataset's attributes. This process also enables one to determine how these various features contribute towards achieving a target variable while detecting any inconsistencies or incomplete information. By serving as the foundation of pre-processing and cleaning steps, EDA ensures that assumptions align with reality in machine learning endeavors. As such,

it is an essential stage for making intelligent decisions throughout this entire process. Figure 2 summarizes what exploratory data analysis entails succinctly.

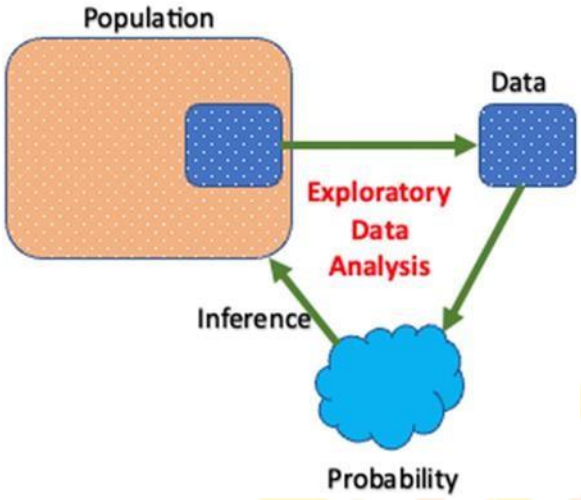


**Figure 2 Exploratory data analysis**

### Feature Extraction Methods

Once the data has been cleaned and preprocessed, it needs to be transformed into a form that can be comprehended by the model. This entails converting all variables to numerical values - an operation known as feature extraction or vectorization. In addition to contributing towards dimensionality reduction, this process serves to retain only crucial features which enhance model accuracy. A variety of techniques can be leveraged for carrying out feature extraction such as gauging word importance in datasets and eliminating redundant information whilst simultaneously forming new attributes from existing ones. Through these strategies, vital features are retained while novel characteristics are generated resulting in an improved version of the original dataset itself. During our research project we employed Count Vectorizer- designed specifically for transforming textual content into vectors [29].

The TF-IDF, which stands for term frequency-inverse document frequency, evaluates the importance of a word within a series of documents by combining two measurements. The first measurement involves calculating how often a specific word appears in one particular document while the second measures that same word's inverse-document-frequency across all other documents contained in the collection. Utilized extensively in endeavors such as natural language processing (NLP) and automatic text analysis, this statistic has multiple potential applications ranging from scoring words to machine-learning techniques.

### Classification Methods and Explainable Techniques

Various classifiers were utilized for the forecast of hate speech on Google Jigsaw data set, which include artificial neural network (ANN) [29], multilayer perceptron (MLP) [30], decision trees, KNN, random forest, multinomial naive Bayes, logistic regression and long short-term memory(LSTM). The explanation method was clarified through BERT and LIME techniques using HateXplain Dataset. This section presents a brief rundown on LSTM technology along with an introduction to BERT and LIME methodologies.

### Deep Learning Model—Long Short-Term Memory (LSTM)

An artificial recurrent neural network (RNN) architecture, known as LSTM, is utilized in the domain of deep learning. Unlike conventional feedforward neural networks, LSTM contains feedback connections and can handle complete data sequences rather than just individual data points.

Incorporating the entire dataset comments as was made possible by designing the LSTM's input layer with a capacity of 30,000 elements each having a size of 128 (equivalent to a total parameter count of 3,840,000). Prior to processing, lemmatization and removals operations were carried out on stop words and punctuation marks. The topmost relevant 30k words were subsequently utilized for this purpose. Using standard ASCII encoding using seven-bit characters ensures that all numbers or letters are uniquely represented per every subset group consisting essentially up until number '27' which corresponds perfectly with element quantity fetched from this set during run-time usage through tokenized inputs augmenting entities captured optimally at scale accordingly.

Dropout layers are utilized to lessen the amount of data being processed while simultaneously increasing the number of extracted features from input. The typical rate at which dropout occurs in LSTM models is 0.2, and it can be observed that the parameter count (131,584) indicates a significant reduction in entities following recurrent layer processing; specifically lowering them from 3,840,000 down to this value. In order to associate each input word with its respective class label via dimensionality reduction through a dense layer outputting roughly 774 units (equivalent to around 128 multiplied by six), ascertaining their classification becomes possible.

A partition of 70% and 30% was applied to the data, assigning that amount for training and testing respectively. The defined model employed binary cross-entropy as a loss function and Adam optimizer. Following this, it underwent fitting on the training set using batch size equaling128.

The LSTM model achieved an accuracy of 97.6%, a precision of 0.85, a recall of 0.83, a macro F1-score of 0., and specificity was recorded as being at 82%.

### BERT (Bidirectional Encoder Representation from Transformers)

In 2018, Google introduced the BERT model as a novel language model with exceptional performance in natural language processing [31]. Unique to this approach is bidirectionality. Through leveraging transformer encoder components, word representation can be accurately furnished by using BERT models. Consequently, diverse objectives are now possible through these versatile language representations created via BERT's functionality.

Pretraining gives BERT a foundational "knowledge" base upon which further training can build adaptations to specific tasks. Unlike other models that rely on one-dimensional understanding, BERT's transformer considers the relationships between all words in a given sentence and thereby grasps contextual meaning. While Angry BERT excels at detecting hate speech paired with emotion classification, our study aimed instead to use explainable AI for evaluating high-performing black-box algorithms' ability to identify this harmful language accurately despite its complex nature (e.g., sarcasm). Thusly we chose standard rather than specialized variants of BERT so explanations could be informative for decision-making by recipients.

**BERT uses the following two semi-supervised models for pretraining [33]:**

1. Masked language model (MLM): In this task, BERT learns a featured representation for each of the words present in the vocabulary. About 85% of the words are used for training, and the remainder are used for evaluation. The selection of the training and evaluation sets is random and in iterations. Through this process, the model learns featured representation in a bidirectional way i.e., learns both the left and the right contexts of the words. In this task, some of the tokens from each sequence are replaced with the token [Mask]. The model is trained to predict these tokens using other tokens from sequence.
2. Next sentence prediction (NSP): In this task, BERT learns the relationship between two different sentences. This task contributes to aspects such as question answering. The model is trained to predict the next sentence. It is similar to the textual entailment task where there are two sentences; it is a binary classification task to predict whether the second sentence succeeds the first sentence.

### Local Interpretable Model—Agnostic Explanations (LIME)

Local Interpretable Model-Agnostic Explanations, known as LIME, provides interpretable explanations for supervised learning models. It computes important features and attributes of the data point by providing weights to its rows and using feature selection techniques. LIME is versatile in handling all types of text, image or video data with local fidelity that accurately reflects the classifier's behavior on a given instance being predicted. LIME offers an agnostic model that can explain predictions regardless of domain-specific knowledge without compromising interpretability since it can be understood by humans. Like other surrogate models such as SHAP and counterfactual explanations used in Explainable Artificial Intelligence (XAI), LIME aims at approximating black-box prediction transparently to people seeking information under novel concepts across domains. However, one significant difference between these methods lies within how they select relevant variables; whereas others recognize salient predictors via more advanced algorithms like Shapley Additive Explanation (SHAP), yet may produce lengthy results not easily readable from laypersons who are often social media arbitrators looking for concise explanation aids- Lime works best here!

Initially, our focus is on interpretability. Certain classifiers incorporate representations that consumers may find difficult to comprehend (e.g., word embeddings). However, with LIME's approach, these classifiers can be described in interpretable terms using familiar language (i.e., words), even if this differs from the original classifier representation.

The concept of model agnosticism pertains to LIME's capability to justify predictions made by any kind of supervised learning algorithm, regardless if the data is comprised of images, text or videos. This approach can accommodate all types of models used in supervised learning and provide compelling explanations for them. By assessing relevant features within its immediate vicinity, LIME generates local optimal interpretations without accessing the inner workings (or "peeking") into a particular model--a key requirement in being truly agnostic about the methods employed. To achieve this objective, we manipulate interpretable inputs surrounding our target instance so that these disturbances reflect on what portions may be contributing towards predictions produced by other machine-learning algorithms under consideration. We then weigh each new set against their proximity from originally provided examples until such point when an explainable pattern emerges based on different related projections analyzed through feature-target selection technologies like PCA/Lasso techniques. Lime has gained

considerable traction as one significant toolset among many available options helping XAI practitioners expand beyond standard tabular databases toward more complex mediums - notably imagery and specific textual manifestations./ Textual analysis relies heavily upon vectorization/embedding operations which form at least basis-level considerate sampling units while image-powered systems fragment critical sections before forwarding potential training samples onwards with weighting handled similarly via assessment aligned alongside pertinent reflections initially measured during pre-processing phases!

## Results

### Model Training and Evaluation for Google Jigsaw Dataset

The LSTM model boasted an impressive accuracy rate of 97.6%, outperforming both multinomial naïve Bayes (96%) and logistic regression (97%). In terms of precision levels, random forest proved to be most effective with a score of 90% while KNN classifier exhibited slightly lower figures but still performed well with an accuracy level reaching up to 88%.

### Model Training and Evaluation for HateXplain Dataset

#### BERT + MLP

In this section, we examine the application of BERT and other techniques to train a dataset while maintaining explainability. The BERT model is designed for NLP tasks and utilizes context from surrounding text to comprehend complex language structures. To implement this approach, we selected both a preprocessor model and a BERT machine learning framework from TensorFlow Hub (2021). As with most datasets, ours included imbalanced data which required addressing through weight optimization or bias setting rather than using resampling or augmentation techniques commonly employed for unbalanced data management. Accordingly, suitable weights were calculated proportional to each class's representation in the dataset then applied during training so that these factors could remove any potential biases between classes.

Out of the 29,027,844 parameters in total, a staggering majority of 29,027,843 were trainable while only one parameter remained nontrainable. The configuration consisted of an input and pre-processing layer, as well as a Keras-based BERT encoder. To reduce parameters and amplify features passed on to the next step, a dense layer followed this encoding process. A dropout method served its purpose in preventing overfitting issues before directing results into another dense classification problem-solving phase. Following these adjustments, we compiled our model utilizing sparse categorical cross-entropy loss function alongside Adam optimizer preferences.

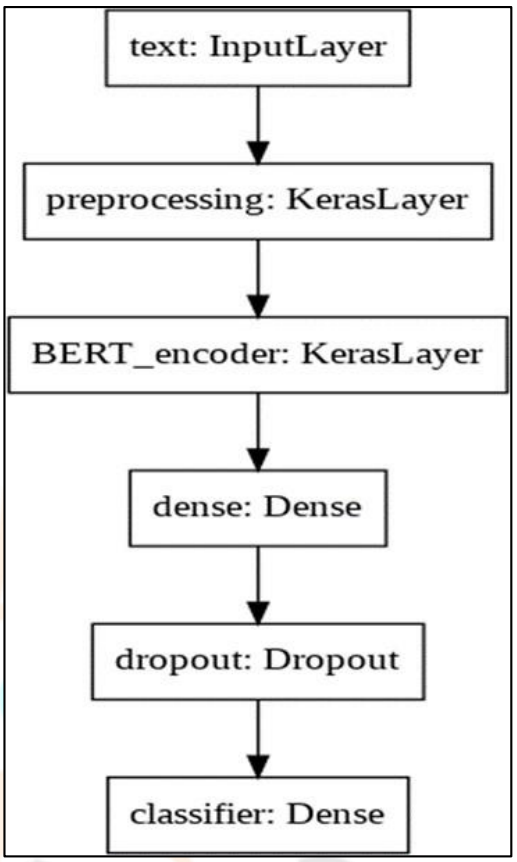


**Figure 3 BERT + MLP model architecture.**

#### BERT + ANN

Afterwards, the BERT + ANN model was trained and its performance evaluated. The architecture comprised an input and preprocessing layer, coupled with a keras-based BERT encoder. To optimize performance, convolution layers were incorporated in conjunction with a 1D global max-pooling layer that computed maximum inputs across channels. Subsequently, for parameter reduction purposes while propagating more features to subsequent layers after pooling was done through adding dense-layer following it. Overfitting avoidance was ensured using dropout followed by another dense-layer at lastly added Thus developed model underwent compilation defined as sparse categorical cross-entropy loss function combined with Adam optimizer methodology applied during execution phase of aforementioned model optimization techniques.

The BERT + ANN and BERT + MLP models underwent 50 epochs of training, resulting in an increase in accuracy as the number of epochs rose. The parameters employed to determine the quantity of training steps and warmup steps were defined by setting the following variables: number of epochs at 50, number of total training steps equal to the

product between steps per epoch and epoch count, while warmup step value was set at ought point one times that same figure for total trainining numbers necessary.

The BERT + MLP model and the BERT + ANN model achieved accuracy rates of 93.67% and 93.55%, respectively, suggesting little difference in conventional evaluation metrics. Nevertheless, regarding explainability metrics discussed later on in this section, it was determined that the BERT + ANN outperformed slightly over the BERT+MLP choice .

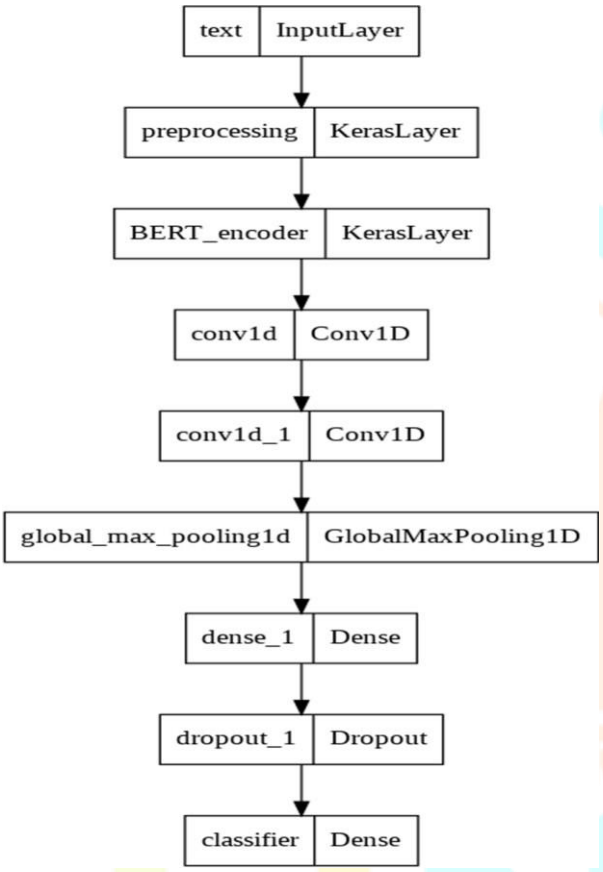


**Figure 4 BERT + ANN model architecture.**

## LIME with Machine Learning Models

The following section explores how the LIME model can be incorporated with other linear machine learning models to offer better understanding and clarity.

The labeled dataset utilized to train BERT with ANN and MLP was also employed for training the LIME model, which used linear noncomplex machine learning models like random forest, naïve Bayes, decision tree,and logistic regression.

### Explainability with Random Forest

The LIME explainer and random forest are used to demonstrate explainability for a specific tweet. The helpful words in the comment were assigned weights by the LIME explainer to show their importance in decision making.

### Explainability with Gaussian Naïve Bayes

The results showcase how significant each used word was in determining the final decision-making process by assigning weights to them through LIME explainer. As evident from Figure 8, words such as "full" and "excus," with their respective weights of 0.08 and 0.07, played a crucial role in contributing to the overall prediction probability. Interestingly enough, for Gaussian naïve Bayes classifier case study showed that including the term 'retard' resulted in reducing hate speech predictions possibility significantly (downgrading its predictive importance) eventually led towards an increase of at least up-to twenty percent regarding text not being labeled negative social abuse or discrimination kind discussion content. Highlighting all textual data sources that contributed either positively or negatively has been done on one side of this figure while estimating parameters within two modeling approaches -speaks volumes itself concerning which model type performed better-: Hatred probability estimation turned out eight times higher when applying gaussian forms instead!

### Explainability with Decision Tree

The LIME explainer assigned weights to each pertinent word, indicating their significance in the overall decision-making process. Based on Figure 9, it appears that certain words like "full," "excus," and "retard" were highly weighted and contributed heavily towards an overall prediction probability of 0.07, 0.06, and o.o6 respectively for hate speech classification. In contrast, none of these same words were given any weight by the decision tree classifier when predicting non-hate speech comments. Therefore, we can discern a pattern using highlighted text wherein the latter achieved 100% predictability for identifying content as hateful using a Decision Tree Classifier model algorithm.

### Explainability with Logistic Regression

The LIME explainer assigns weights to relevant words in the comment as a measure of their contribution towards the final decision. Notably, "excus" and "second" had high weights contributing to a prediction probability of 0.03 and 0.04 respectively while terms like "retard" and "full," with respective weights at 0.04 and 0.03 were associated with non-hateful language usage. Overall classification accuracy for hate speech stood firmly at around 95% based upon logical regressions carried out over this dataset sample being analysed herein".

### Summary of Results for the HateXplain Dataset

Notably, BERT variants demonstrated significantly superior performance compared to other linear explainable models: BERT + MLP achieved an impressive accuracy rate of 93.67%, trailed closely by BERT + ANN with an accrual rating of 93.55%. Moreover, measures such as precision, recall and macro F1 also indicated that the BERT variants outperformed other linear alternatives; logistic regression with LIME earned recognition among these options displaying an outstanding level in terms of both accuracy (88.57%) and macro-F1 score (93-75%). Ultimately outlined via bar chart format within Figure 11 above mentions outcomes prevail present accords based upon comprehensive study assessment techniques applied across differing frameworks observed herein regarding data sets evaluated throughout analyses noted at outset hereof.

### Explainability Metrics

To assess the trained models' explainability, we utilized the ERASER benchmark [35], which evaluates rationalized NLP models based on their agreement with human rationales. DeYoung et al.'s (2020) proposal avoids excessive rigidity by assessing plausibility and faithfulness rather than exact matches between predicted and reference rationales. Any overlaps in word predictions count as a match, while token level calculations are compared to human annotations for accuracy. We employed various measures from the ERASER benchmark to derive these comparisons, including IOU F1-score at both token-level precision-recall curve area under its score along intersection-over-union categories having more than 50% overlap above ground truth rationale prediction; high scores across all metrics indicate strong plausibility alongside faithfulness representing an accurate reasoning process of model respectively therein evaluated measure explained them well enough overall during our experimentation too without many practical issues encountered so far!

In order to gauge the accuracy of the models, calculations were conducted for comprehensiveness and sufficiency. The assessment of comprehensiveness involves determining how much probability changes in relation to the initially predicted class once significant tokens have been removed from consideration. A higher score on this measure points towards a more dependable interpretation. Sufficiency evaluates whether important tokens are adequate enough to support predictions made by models; it measures if snippets within exact rationales suffice for accurate forecasting. Lower scores here indicate greater faithfulness exhibited by a model.

BERT + MLP demonstrated superior plausibility performance, scoring highest in IOU F1, token F1, and AUPRC compared to other models. Regarding faithfulness, the BERT + ANN model achieved top results with a comprehensiveness score of 0.4199. These outcomes represent an improvement over Mathew et al.'s (2020) original paper as evidenced by human interpretability favouring various forms of BERT variants; specifically due to its simpler architecture than MLP's complex structure resulting in slightly higher comprehensiveness scores for the same parameter trends observed previously mentioned study edition(s).

### Bias-Based Metrics

The detection models for hate speech have the potential to unfairly target certain groups that are already victims of abuse, as noted in studies by Sap et al. (2019) and Davidson, Bhattacharya, and Weber (2019). To determine if there are any inadvertent biases within these models, we utilized Borkan et al.'s AUC-based metrics from 2019. This included computations for subgroup AUC (area under ROC curve), background positive/subgroup negative AUC or "BPSN," and background negative/subgroup positive AUC or "BSNP." The results of the subgroup analysis illustrate how well the model can differentiate between toxic comments versus those that lack toxicity; higher values indicate greater accuracy on this front for detecting each respective group's content – be it normal conversations only being discriminated against due merely their affiliation with a specific community rather than because they contain violent language aimed at members outside said minority classification/self-identity category - while lower numbers suggest more erroneous differentiation occurs instead when identifying such harmful posts.

The summarized bias-based metrics for all implemented models are presented in Table 10. Analysis reveals that BERT variants outperformed other linear models significantly, with highest subgroup AUC, BPSN AUC and BSNP AUC values. Specifically, it was found that the combination of BERT + MLP achieved the most accurate results; recording scores of 0.8229, 0.7752 and 0.8077 respectively for these three metrics.

## Conclusions

The aim of this research was to showcase the detectability of hate speech with explainable artificial intelligence (XAI) through the analysis of two datasets. In order to achieve this, exploratory data examination was conducted on both sets in an effort to uncover patterns and insights.

Subsequently, a variety of XAI models were used for training purposes on each dataset resulting in useful interpretable findings extracted from both samples. Further discussion regarding study outcomes can be found within this section.

### Conclusions of the Study on the Google Jigsaw Dataset

The Google Jigsaw dataset, released by the company of the same name, encompasses user discussions from English Wikipedia talk pages. We utilized various interpretable models (decision tree, KNN, random forest, multinomial naïve Bayes logistic regression and LSTM) trained on this data to compare their performance. Our findings show that in terms of accuracy (97.6%) and recall scores (83%), LSTM outperformed all other models tested. Random forest displayed superiority with regards to precision 90% outcomes while specificity was highest at 87%. The decision trees implementation was found accurate having achieved an output figure equivalenting 89%, while randomly structured forests reached 91%. It can be observed subsequently shown results indicate LSTMs overall better quality in measurement for respective attributes studied viz-a-viz accuracy, precision, and recall along with macro F1-score compared against analysis made previously by Risch et al.(2020).

### Conclusion of the Study on the HateXplain Dataset

The HateXplain dataset contains posts from Twitter and Gab that have been annotated by human annotators. Several state-of-the-art models were tested on this dataset to evaluate hate speech detection using varying levels of explainability. Models incorporating LIME with interpretable decision trees, random forest, logistic regression, and naïve Bayes methods were used to extract significant word weights for the model's decisions. Additionally, BERT variants optimized performance in detecting hate speech with BERT + ANN proving slightly better than BERT + MLP overall according to three subsets of evaluation metrics: performance (including accuracy), bias-based analysis as well as plausibility and faithfulness measures outlined in previous research cited here.Mathew et al.(2020) had previously divided these metrics into third-party groups while also showcasing examples where LIME could provide black-box textual explanations when needed during testing stages too! For the HateXplain dataset, we utilized explanation metrics based on DeYoung et al.'s ERASER benchmark. These metrics determined how well models could identify hateful comments compared to other existing ones.

Therefore, it can be concluded that the BERT versions utilized in the study demonstrated outstanding results over its base model. The combination of BERT and ANN yielded optimal outcomes with regard to interpretability while the amalgamation of BERT with MLP resulted in overall superior performance when compared against classical models like logistic regression, KNN, naïve Bayes, decision trees and random forests.

**Author Contributions:** Conceptualization, D.M. and H.S.; methodology, D.M. and H.S.; software, D.M.; validation, D.M., H.S., and S.R.; formal analysis, D.M.; investigation, D.M.; resources, D.M. and H.S.; data curation, D.M.; writing—original draft preparation, D.M.; writing—review and editing, H.S. and S.R.; visualization, D.M.; supervision, H.S.; project administration, H.S.; funding acquisition, H.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The datasets are publicly available as follows: Google Jigsaw dataset: https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data (accessed 5 January 2022); HateXplain dataset: https://github.com/hate-alert/HateXplain/tree/%20master/Data (accessed 7 January 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## REFERENCES

1. Davidson, T.; Warmsley, D.; Macy, M.; Weber, I. Automated Hate Speech Detection and the Problem of Offensive Language. Available online: http://arxiv.org/abs/1703.04009 (accessed on 11 August 2022).
2. Chen, Y.; Zhou, Y.; Zhu, S.; Xu, H. Detecting offensive language in social media to protect adolescent online safety. In Proceedings of the 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust and 2012 ASE/IEEE International Conference on Social Computing, SocialCom/PASSAT, Amsterdam, The Netherlands, 3–5 September 2012; pp. 71–80. [CrossRef]
3. Balkir, E.; Nejadgholi, I.; Fraser, K.C.; Kiritchenko, S. Necessity and sufficiency for explaining text classifiers: A case study in hate speech detection. arXiv 2022, arXiv:2205.03302.
4. Chatzakou, D.; Kourtellis, N.; Blackburn, J.; de Cristofaro, E.; Stringhini, G.; Vakali, A. Mean birds:

Detecting aggression and bullying on Twitter. In WebSci 2017—Proceedings of the 2017 ACM Web Science Conference; Association for Computing Machinery: New York, NY, USA, 2017; pp. 13–22. [CrossRef]
5. Founta, A.M.; Chatzakou, D.; Kourtellis, N.; Blackburn, J.; Vakali, A.; Leontiadis, I. A Unified Deep Learning Architecture for Abuse Detection. In WebSci 2019—Proceedings of the 11th ACM Conference on Web Science; Association for Computing Machinery: New York, NY, USA, 2018; pp. 105–114. [CrossRef]
6. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. Adv. Neural Inf. Process. Syst. 2017, 30.
7. Arras, L.; Montavon, G.; Müller, K.R.; Samek, W. Explaining recurrent neural network predictions in sentiment analysis. In EMNLP 2017—8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA 2017—Proceedings of the Workshop; Association for Computational Linguistics: Copenhagen, Denmark, 2017. [CrossRef]
8. Mahajan, A.; Shah, D.; Jafar, G. Explainable AI approach towards toxic comment classification. In Emerging Technologies in Data Mining and Information Security; Springer: Singapore, 2021; pp. 849–858.
9. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In Proceedings of the NAACL-HLT 2016—2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144. [CrossRef]
10. Samek, W.; Montavon, G.; Lapuschkin, S.; Anders, C.J.; Müller, K.-R. Toward Interpretable Machine Learning: Transparent Deep Neural Networks and Beyond. Available online: https://doi.org/10.48550/arXiv.2003.07631 (accessed on 11 August 2022). [CrossRef]
11. Doshi-Velez, F.; Kim, B. Towards A Rigorous Science of Interpretable Machine Learning. 2017. Available online: https://doi.org/10.48550/arXiv.1702.08608 (accessed on 11 January 2022). [CrossRef]
12. Hind, M.; Wei, D.; Campbell, M.; Codella, N.C.F.; Dhurandhar, A.; Mojsilovic´, A.; Natesan Ramamurthy, K.; Varshney, K.R. TED: Teaching AI to explain its decisions. AIES 2019. In Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society; Association for Computing Machinery: New York, NY, USA, 2019; pp. 123–129. [CrossRef]
13. Montavon, G.; Samek, W.; Müller, K.-R. Methods for interpreting and understanding deep neural networks. Digit. Signal Process.
14. 2018, 73, 1–15. [CrossRef]
15. Gilpin, L.H.; Bau, D.; Yuan, B.Z.; Bajwa, A.; Specter, M.; Kagal, L. Explaining explanations: An overview of interpretability of machine learning. In Proceedings of the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics, DSAA, Turin, Italy, 1–3 October 2018. [CrossRef]
16. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S. Explainable AI for Trees: From Local Explanations to Global Understanding. arXiv 2019, arXiv:1905.04610. Available online: http://arxiv.org/ abs/1905.04610 (accessed on 11 May 2022). [CrossRef] [PubMed]
17. Nori, H.; Jenkins, S.; Koch, P.; Caruana, R. InterpretML: A Unified Framework for Machine Learning Interpretability. arXiv 2019, arXiv:1909.09223. [CrossRef]
18. Ahmed, U.; Lin, J.C.-W. Deep Explainable Hate Speech Active Learning on Social-Media Data. IEEE Trans. Comput. Soc. Syst.
19. 2022, 1–11. [CrossRef]
20. Barredo Arrieta, A.; Díaz-Rodríguez, N.; del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies,

opportunities and challenges toward responsible AI. Inf. Fusion 2020, 58, 82–115. [CrossRef]
21. Das, A.; Rad, P. Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. arXiv 2020, arXiv:2006.11371.
22. [CrossRef]
23. Kanerva, O. Evaluating Explainable AI Models for Convolutional Neural Networks with Proxy Tasks. Available online: https://www.semanticscholar.org/paper/Evaluating-explainable-AI-models-for-convolutional_Kanerva/d91062a3e13ee034af6807e1819a9ca3051daf13 (accessed on 25 January 2022).
24. Gohel, P.; Singh, P.; Mohanty, M. Explainable AI: Current STATUs and Future Directions. Available online: https://doi.org/10.1109/ACCESS.2017 (accessed on 30 January 2022). [CrossRef]
25. Fernandez, A.; Herrera, F.; Cordon, O.; Jose Del Jesus, M.; Marcelloni, F. Evolutionary fuzzy systems for explainable artificial intelligence: Why, when, what for, and where to? IEEE Comput. Intell. Mag. 2019, 14, 69–81. [CrossRef]
26. Clinciu, M.-A.; Hastie, H. A Survey of Explainable AI Terminology. In Proceedings of the 1st Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence (NL4XAI 2019); Association for Computational Linguistics: Copenhagen, Denmark, 2019; pp. 8–13. [CrossRef]
27. Hrnjica, B.; Softic, S. Explainable AI in Manufacturing: A Predictive Maintenance Case Study. In IFIP Advances in Information and Communication Technology, 592 IFIP; Springer: New York, NY, USA, 2020; pp. 66–73. [CrossRef]
28. Miller, T. Explanation in artificial intelligence: Insights from the social sciences. Artif. Intell. 2019, 267, 1–38. [CrossRef]
29. Mathew, B.; Saha, P.; Yimam, S.M.; Biemann, C.; Goyal, P.; Mukherjee, A. HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. arXiv 2020, arXiv:2012.10289. Available online: http://arxiv.org/abs/2012.10289 (accessed on 14 June 2021).
30. ML|Overview of Data Cleaning. GeeksforGeeks. 15 May 2018. Available online: https://www.geeksforgeeks.org/data-cleansing-introduction/ (accessed on 3 April 2022).
31. Pearson, R.K. Exploratory Data Analysis: A First Look. In Exploratory Data Analysis Using R; Chapman and Hall/CRC: New York,
32. NY, USA, 2018.
33. Using CountVectorizer to Extracting Features from Text. GeeksforGeeks. 15 July 2020. Available online: https://www.geeksforgeeks.org/using-countvectorizer-to-extracting-features-from-text/ (accessed on 3 April 2022).
34. Bisong, E. The Multilayer Perceptron (MLP). In Building Machine Learning and Deep Learning Models on Google Cloud Platform;
35. Apress: Berkeley, CA, USA, 2019; pp. 401–405. [CrossRef]
36. Kamath, U.; Graham, K.L.; Emara, W. Bidirectional encoder representations from transformers (BERT). In Transformers for Machine Learning; Chapman and Hall/CRC: New York, NY, USA, 2022; pp. 43–70. [CrossRef]
37. Awal, M.R.; Cao, R.; Lee, R.K.-W.; Mitrovic, S. AngryBERT: Joint Learning Target and Emotion for Hate Speech Detection. arXiv
38. 2021, arXiv:2103.11800. Available online: http://arxiv.org/abs/2103.11800 (accessed on 16 July 2022).
39. Nair, R.; Prasad, V.N.V.; Sreenadh, A.; Nair, J.J. Coreference Resolution for Ambiguous Pronoun with BERT and MLP. In Proceedings of the 2021 International Conference on Advances in Computing and Communications (ICACC), Kochi, India, 21–23 October 2021; pp. 1–5. [CrossRef]
40. Biecek, P.; Burzykowski, T. Local interpretable model-agnostic explanations (LIME). In Explanatory Model Analysis; Chapman and
41. Hall/CRC: New York, NY, USA, 2021; pp. 107–123. [CrossRef]