SUPERVISED SIGNAL-FEATURES LEARNING FROM CYBER-TEXT AND NOVEL DEEP LEARNING TECHNIQUES FOR FINE-GRAINED CYBERBULLY CLASSIFICATION

A Dissertation in
Electronics and Information Engineering

By
Junaid Iqbal Khan

Dissertation Submitted in Partial Fulfillment of the Requirements
for the Master's Degree

School of Electronics and Information Engineering
The Graduate School
Korea Aerospace University

SUPERVISED SIGNAL-FEATURES LEARNING FROM CYBER-TEXT AND NOVEL DEEP LEARNING TECHNIQUES FOR FINE-GRAINED CYBERBULLY CLASSIFICATION

A Dissertation in
Electronics and Information Engineering

Dissertation Submitted in Partial Fulfillment of the Requirements for the
Master's Degree

School of Electronics and Information Engineering
The Graduate School
Korea Aerospace University

SUPERVISED SIGNAL-FEATURES LEARNING FROM CYBER-TEXT AND NOVEL DEEP LEARNING TECHNIQUES FOR FINE-GRAINED CYBERBULLY CLASSIFICATION

This certifies that the dissertation of Junaid Iqbal Khan, entitled " Supervised signal-features learning from cyber-text and novel deep learning techniques for fine-grained cyberbully classification", is approved.

Prof. Dr. Sungchang Lee, Thesis Supervisor (Signature)

Prof. Dr. Junsun Ahn (Signature)

Prof. Dr. Jeonghee Han (Signature)

Korea Aerospace University,

South Korea

December,2022

SUPERVISED SIGNAL-FEATURES LEARNING FROM CYBER-TEXT AND NOVEL DEEP LEARNING TECHNIQUES FOR FINE-GRAINED CYBERBULLY CLASSIFICATION

A Dissertation to Korea Aerospace University

for Degree of Master

By
Junaid Iqbal Khan

Accepted on Recommendation of
Prof. Sungchang Lee
Professor of Korea Aerospace University, South Korea
Supervisor, Examiner

Abstract

Supervised signal-features learning from cyber-text and novel deep learning techniques for fine-grained cyberbully classification

Khan, Junaid Iqbal
Dept. of Electronics and Information
Engineering,
Graduate School,
Korea Aerospace University

(Advisor: Prof. Lee, Sungchang)

Cyberbullying detection and mitigation is one of the main challenges of current massive social-media popularity. The reduction of cyberbully detection task to a text classification problem is prone to computational and classification performance limitations, while the current trends has been centered around textual embedding-based approaches. In this dissertation, I tackle these challenges by proposing an alternate route to mapping textual embeddings toward classification probabilities. More specifically, I propose and establish a novel signal feature learning strategy for fine-grained cyberbully classification, by taking advantage of techniques like Word2Vec, Autoencoders, Multi-layer Perceptron (MLP) and Sinc-interpolation. In this direction, I also proposed new models in 1D and 2D input topology with innate several contributions for cyberbully classification. In 1D approaches, I introduced a new way of fusing signal features via Graph Convolution Network (GCN), increasing the temporal resolution of fused features via a novel deep learning layer called binding function, and mapping the resultant features to probabilities via attention and residual based systems. For my 2D approach, I make a novel utility of state-of-the-art PatchConvNet model for mapping the STFT spectrogram associated to signal features towards cyberbully classification probabilities. I rigorously

evaluate my approaches in the context of single-label and multi-label, multi-class cyberbully classification cases. I empirically show over different datasets and benchmark models to demonstrate the efficacy of proposed approaches in several domains.

Keywords

cyberbullying, deep learning, natural language processing, signal processing

“Science advances one funeral at a time.”

Max Plank

Acknowledgements

I acknowledge and thank Prof. Sungchang Lee for formally advising me during my Master's duration. I would also like to thank Prof. Farman Ullah and Dr. Jebran Khan for their interaction along the way. I owe my acknowledgement to my colleagues Furqan Ali, Abdul Wasay Sardar and Jamshid Bacha. Last but not least, I acknowledge my family for supporting me.

Contents

List of Figures

List of Tables

List of Equations

Symbols

Notation	Description
σ	Activation function
ϕ	Arbitrary input
$\mathbb{E}$	Trainable Embedding Matrix
ϕ_D	Dropout operation over ϕ
y	Output probability vector of MLP fine-tuner
X	Word2Vec vector
n_w	Dimensions of Word2Vec vector
V	Set of nodes of a graph
E	Set of edges of a graph
k	Output dimension of Autoencoder
T	Text embedding matrix
T_s	Signal Features
z	Output of Graph Convolution Network
Z	Fused signal features
$\hat{Z}$	Output of binding function
α	Thresholding function
N_{max}	Maximum word count
w	Window function for STFT
w_s	Window function for sinc-interpolation
D	Diagonal matrix
A	Adjacency matrix
ζ, Θ	Trainable parameters of binding function
W, b	Trainable parameters associated to Graph Convolution Network
N	Number of words per a given text

RSM	Relative sparsity metric
Z_{norm}	Fused signal features subjected to layer normalization
ξ	1D-convolution kernel
ω	Angular frequency
τ	Threshold associated to adjacency matrix
ϵ	Threshold associated to thresholding function
Q, K, v	Query, key and value matrices
H	Output of individual Attention head
Y	Output of multi-headed Attention head
Δ_{seq}	Sequence-wise change
Δ_{var}	Variational change
l	1D-convolution kernel size
s	Stride size

Chapter 1

Introduction

With current major engagement in social networking and ever increasing online presence, safety of online communities with several cyber-related crimes. With an array of social media websites like Facebook, Twitter, Instagram, Youtube, etc. the maximum number of visitors on these sites ranges from 0.3 billion to 2.3 billion per platform worldwide and plays a vital role in several sectors of society [1]. Among many problems that arise with information exchange on these platform, whether that be in the form of tweets, text messages, comments to posts or group chats, cyberbullying is one the most prominent problems. For example, a study in [2] found that for around 40,000 tweets extracted with "#cyberbully", nearly 60% were engaged in a cyberbully situation.

It is prone to challenge to control the number of cyberbully users from these social media websites due to the creation of fake accounts. Though there exist commercial softwares [3] and APIs [4] for automatic detection and blocking of repugnant on online social media, these systems are not robust to subtle forms of cyberbullying, either due to their reliance on keyword-based moderation strategy or inability to differentiate the extent of intimidation content may cause, leading to unreliability in their performance. An example comparison of this vocabulary between racist and non-cyberbully related texts is shown in figure 1.1. Alongside keywords, the context of sentence [5] has been identified as a critical factor for creating a bullying effect. Furthermore, keyword-based matching can be computationally complex due to its NP-hardness. Additionally, these systems can get limited by keyword dictionary size. Lastly, social networking is largely dependent upon human moderators to flag and ban abusive content, but large traffic volume can greatly reduce the efficiency of this system, followed by human error.

Deep learning classification models have been studied to produce an effective and efficient solution against these challenges. Whilst currently, cyberbully detection dissertation has spanned upon input data of textual, image [6] and video [7] format, my dissertation only focuses towards textual type data.

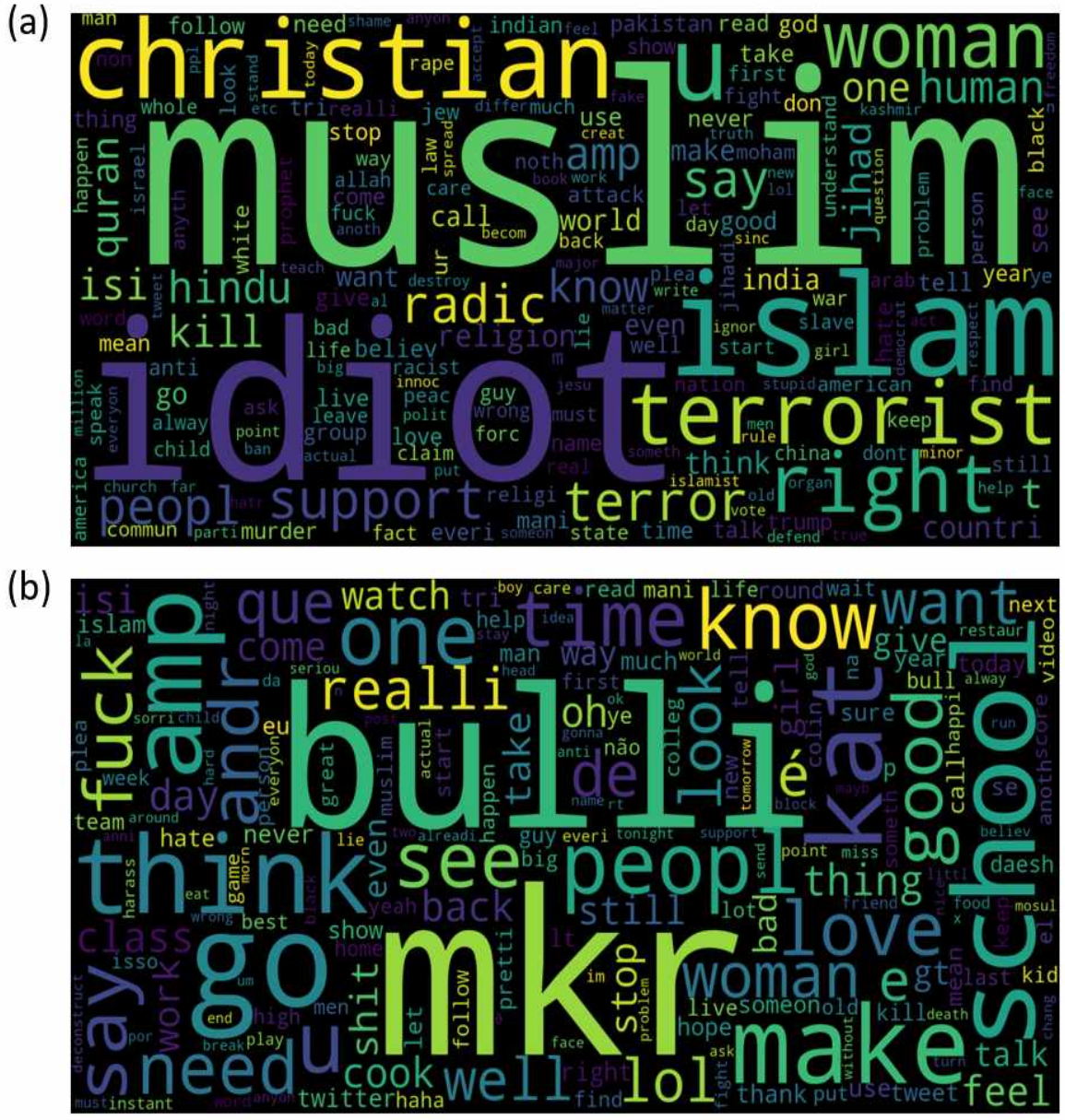

FIGURE 1.1: (a) word cloud representation associated to typical cyberbully text centered around racism (b) word cloud representation associated to typical texts not necessarily in context of cyberbullying

To the end of cyberbullying detection, the researchers have been actively developing deep learning systems to classify texts into either cyberbullying or not [5], fine-grained cyberbullying classification of these texts [8] and severity of cyberbullying content [4]. A common theme in all of the deep learning-driven cyberbully classification schemes has been centered around converting textual (and possibly other data modalities) into embedding features and mapping these features toward classification probabilities. An example demonstration of these textual embeddings is represented in figure 1.2, where the later sequential model learns the hyperplane to separate the cyberbully and non-cyberbully labeled text embeddings, thus leading to cyberbully classification.

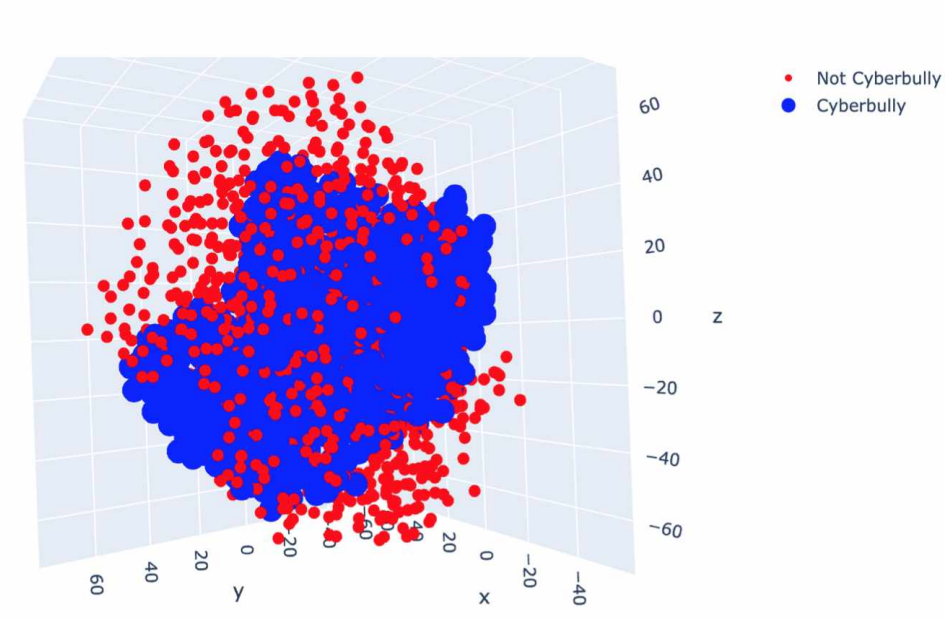

FIGURE 1.2: Text embedding visualization of a cyberbully dataset based on Swivel co-occurrence matrix factorization, which is pre-trained on English Google News 130GB corpus

In this dissertation, I propose a novel methodology of classifying textual data into different cyberbully related classes, which boils down to either a single-label multi-label or multi-label multi-class classification problem, in a unified manner. In this methodology, I take a digression from textual embeddings and propose to extract signal features from textual data, which hold characteristic morphology in pertinence to the cyberbully class. Realizing the potential of these signal features, I make novel use of a few deep learning architectural elements like Graph convolution [9], Multi-headed attention [10] and Residual Wavenet Block [11], to construct 1D classification models. Not only this, but I also propose an additional novel 2D classification model, by converting the signal features into 2D short-time Fourier transform (STFT) spectrograms.

1.1 Contribution

The contributions of this research work are as follows:

- Contrary to previous approaches of extracting text embeddings, I employ extraction of novel signal features, which are later classified by a learning model. More precisely, I extend the standard Skip-Gram based Word2Vec model [8, 12] by a Deep Autoencoder, followed by Multi-Layer Perceptron (MLP) based fine-tuning. The resultant vectors are resampled by Sinc-interpolation to produce signal features over channels

- I apply a novel 1D classification approach, by fusing the signal features along channels via a Graph Convolutional Network (GCN). Then, the temporal representation of the resultant fused signal feature is improved via a proposed binding function. Finally, the resultant sequence is mapped towards the probability vector either via proposed attention or residual system
- I propose another novel 2D-approach to exploit the signal characteristic of proposed feature, by converting signal features into Short-Time Fourier-Transform (STFT) and mapping the resultant image data to classification probabilities via PatchConvNet [13].
- I empirically show the precedence of classification performance of the proposed scheme over state of the art benchmark models, while evaluation over two datasets in context of single and multi-label multi-class classification scenario. Specifically, I show that the proposed scheme performs well over small sized and noisy datasets and the performance rapidly improves over increasing the datasets. This is further complimented with the fact that overall parameter count of proposed model significantly less as compared to embedding based models, which I considered as benchmarks.

1.2 Dissertation Organization

The remaining part of this dissertation is organized as follows. Chapter II provides a brief overview of related papers and context of cyberbullying classification in textual data. The details of the proposed framework are given in Chapter III. Chapter IV provides experimental validation over two datasets. Finally, I discuss conclusion and future works in Chapter V and Chapter VI respectively.

Chapter 2

Literature Survey

Previously, researchers have employed standard rule-driven Natural Language Processing (NLP) approaches [14, 15] for cyberbully identification from textual data with considerable success. With the pursuit of more competitive classification models for cyberbully-based texts, the current state of art models can be decomposed into feature extractors and classifiers. The feature extractors determine effective embeddings of text in some arbitrary dimensional space which effectively capture the semantic value of each text, while the classifiers map these embeddings toward cyberbully classification probability.

Notable feature extractors can be further subdivided into word-based embeddings and text-based embeddings. In the case of word-based embedding, notable examples include Bag-of-Words (BoW) [12, 16], Word2Vec (either Continuous Bag-of-Words or Skip-gram models) [12], GloVe [12], FastText [17] and term frequency-inverse document frequency (Tf-Idf) [12]. On the other hand, the text-level embedding techniques include tokenization [18, 19], character embedding network [20, 21], Latent Semantic Features [12], Latent Dirichlet Allocation [12], Doc2Vec [22], stacked denoising autoencoder [23], Gated Recurrent Units (GRU) [24, 25] and Transformers [21]. Transformers have already known to be greatly overhead in NLP related tasks, and their future generations like Bi-directional Encoder Representation of Transformers (BERT) [8, 26, 27], DistillBERT [8, 27], Sentence-BERT [8], RoBERTa [26, 27] and XLNet [27] have empirically shown the state of the art results in the context of cyberbully classification. These models have been mostly utilized in pretrained settings for extracting text features.

The classifiers considered in cyberbully classification literature have ranged from K-nearest neighbors (KNN) [28], support vector machine (SVM)[16, 22, 22], Naive Bayes [28], XGBoost (XGB), Logistic regression (LR) [22], Decision tree [16, 22, 28], Random forest [22, 28], Gradient boosting

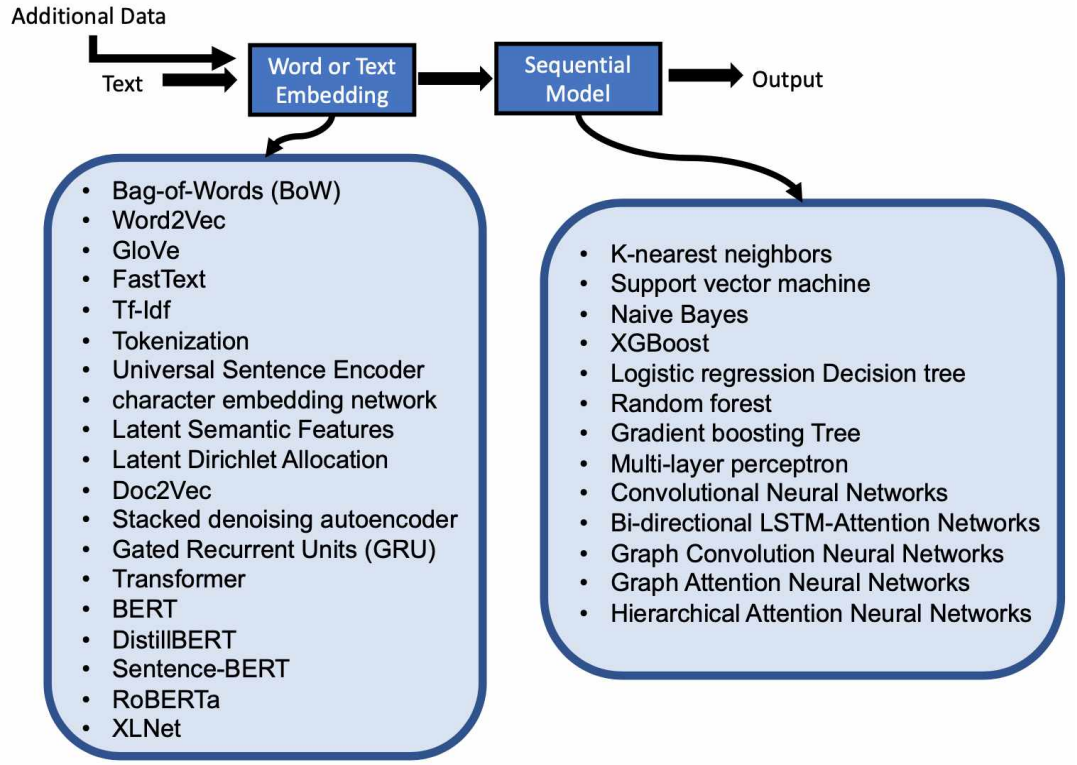

FIGURE 2.1: Abstraction of Cyberbully detection in Literature

regression tree[22] to multi-layer perceptron (MLP) [22, 27, 29], Convolutional Neural Networks [18], Bi-directional LSTM-Attention Networks [19], Graph Convolution Neural Networks (GCN) [8], Graph Attention Neural Networks (GAN) [25, 30] and Hiearichal Attention Neural Networks (HAN) [24]. Furthermore, researchers have also considered alternative features like temporal [31], sentimental [32], media [29] and user-based information of the texts [29, 32], or even combining them with the textual features [33], based on provided datasets for more competitive classification benchmarks. Additionally, ensembling approaches [16, 18, 27] over mentioned models has been considerably studied. A summary of the cyberbully detection literature is demonstrated in figure 2.1. For the purpose of text classification, the word embeddings in themselves might not be lucrative for classification problem, but requires a mapping of the word embeddings of the individual words of a text, towards a embedding vector associated with that text. These vectors can then be mapped towards classification probabilities. An example demonstration is shown in figure 2.2, where pretrained word embeddings (Word2Vec) and pretrained textual embeddings (Universal Sentence Encoder) are represented in 3D Euclidean space via dimensionality reduction technique called t-distributed stochastic neighbor embedding (t-SNE). It is apparent that both kind of embeddings projects words or sentences onto specific points in Euclidean space, and the relative distance between these points is analogous to their similarity.

Works done in literature have been either to order-wise stack the word embeddings into higher rank

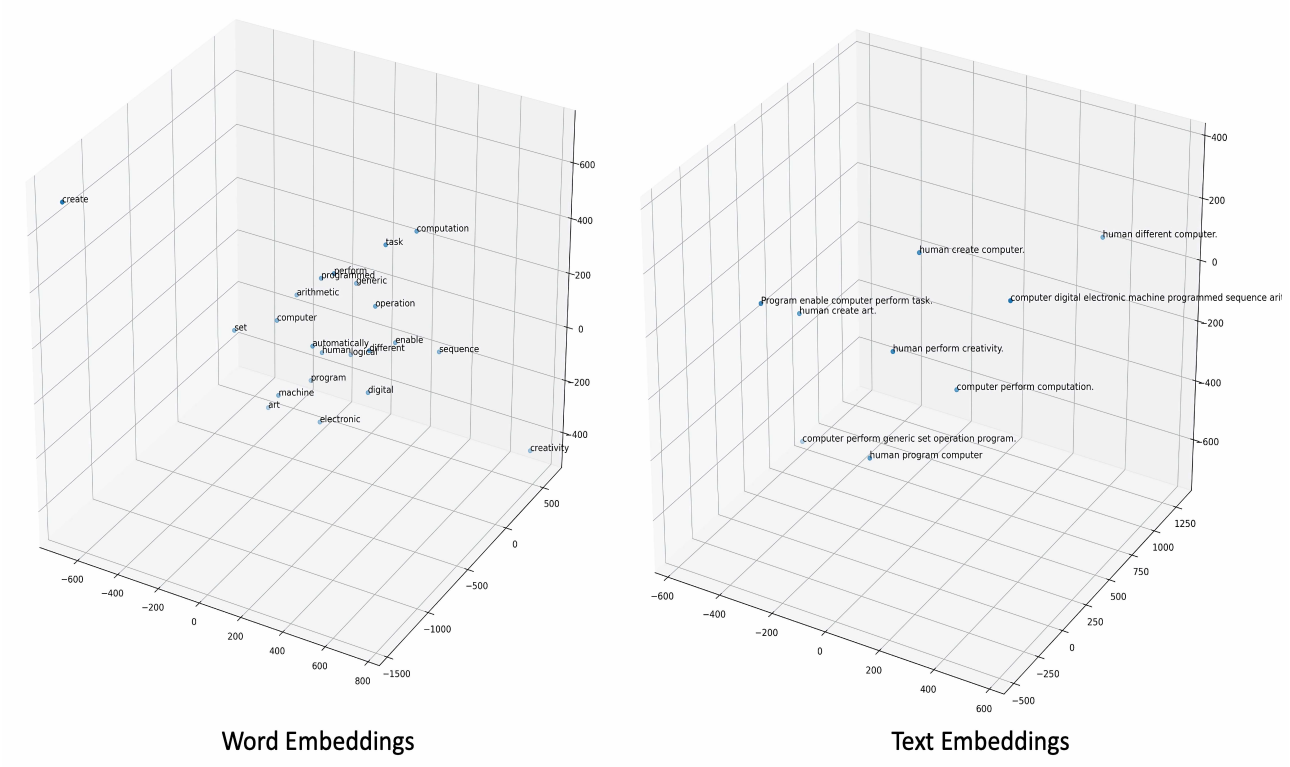

FIGURE 2.2: Comparison of two main types of NLP embeddings

matrix [34], formation of graphs based on similarity metrics [35] or to produce embeddings in the same rank as word embeddings by combining these embeddings with an unsupervised learning technique [36, 37]. Furthermore, dimensionality reduction [12, 35] upon embeddings in stacking case or graph formation case has been studied to be effective in classification performance. In either cases, the fundamental nature has been unsupervised learning schemes, the end product of which have been vectors or matrices in an embedding space, the distance between vectors in this space imply the extent of similarity between vectors.

Chapter 3

Methodology

My proposed methodology divides into tasks of text pre-processing, feature (signal) extraction, 1D and 2D classification approaches. A detailed flow diagram of my proposed scheme is shown in figure 3.1. I further provide details of my framework as follows.

3.1 Text Preprocessing

The raw textual data is filtered from possible symbols like punctuation, emoji, emoticons and other miscellaneous symbols, alongside with numbers, URLs and stop words during initial cleaning phase. Additionally, all words in the sentences are lowercased in this stage as well. This is followed by stemming operation via PorterStemmer[38] to convert derived words of the text to base form. Next, I utilize WordNet's morphy function [39] to perform sequent lemmatization operation. These set of operations lead to lesser noise, and subsequent lesser computations, leading to improved performance by classifiers [38].

3.2 Signal Feature Extraction

Within the pipeline of signal feature extraction, I start with extracting word embeddings via Word2Vec model, followed by dimensionality reduction by Deep Autoencoder, readjustment of reduced vectors by MLP based fine-tuning. Finally, the readjusted vectors are subjected to Sinc-interpolation to produce final signal features. These steps are described below.

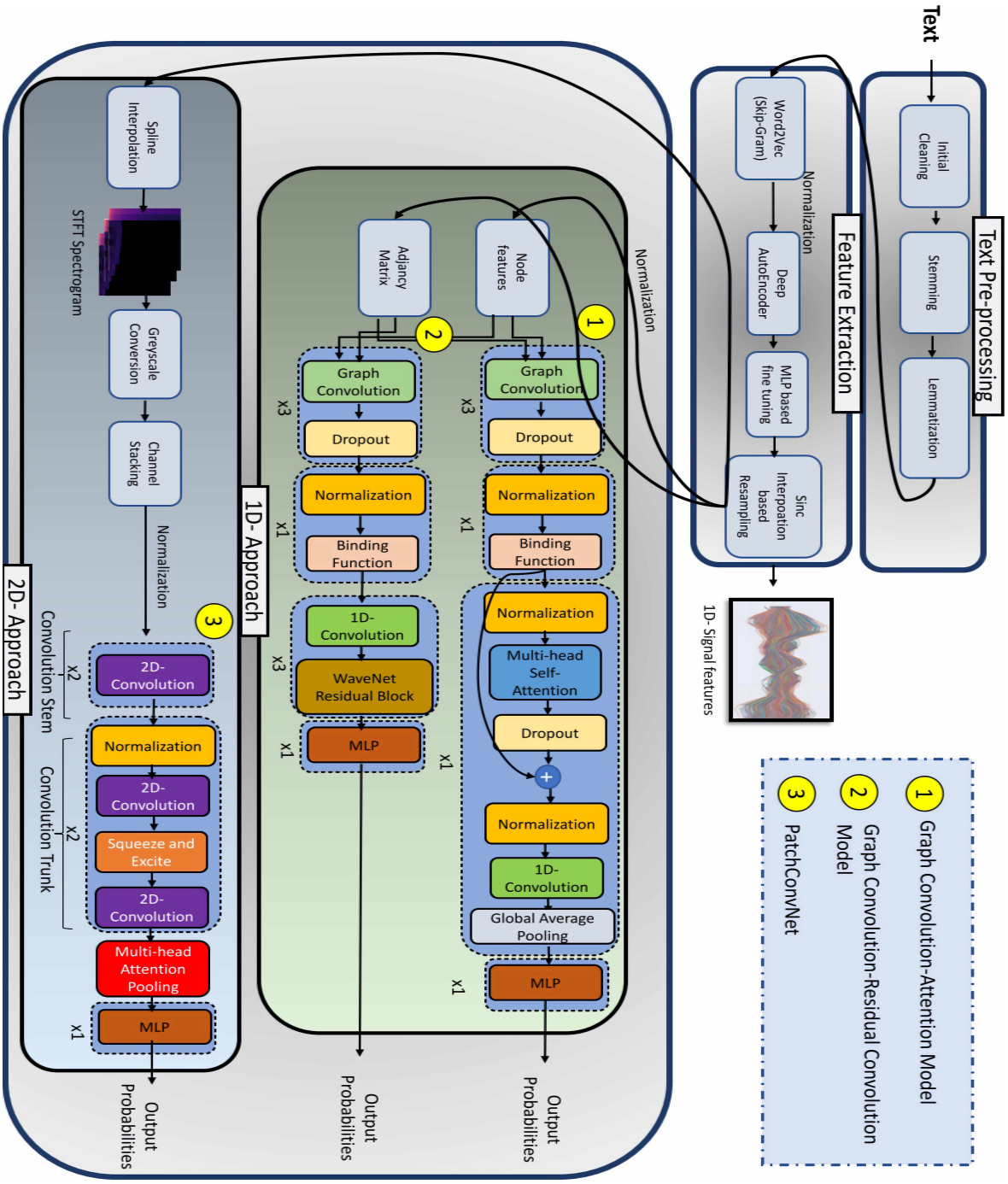

FIGURE 3.1: Landscape view of proposed scheme for fine-grained cyberbully text classification

3.2.1 Word2Vec

The refined textual dataset after the text preprocessing phase, is used to train a Word2Vec Skip Gram model [8, 12, 16] over certain epochs to learn word embeddings as $X \in \mathbb{R}^{n_w}$, where n_w is the dimension of Word2Vec vectors. The choice of Word2Vec revolves around it great capacity to learn relationship between words and phrases in unsupervised manner, which identically translates to euclidean distance between corresponding word embedding vectors, at the cost of in-vocabulary learning limitation.

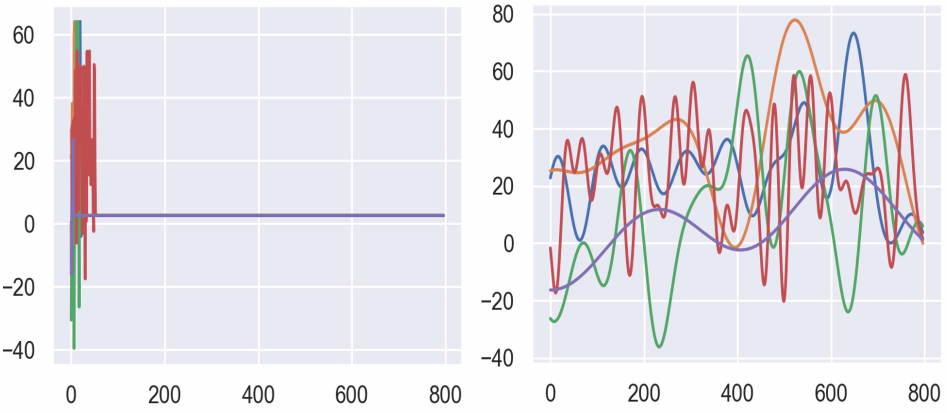

FIGURE 3.2: Left plot show first channel of text embedding matrix corresponding to 5 random cyberbully related texts. Right plot shows corresponding signal features via Sinc-interpolation based resampling

3.2.2 Deep Autoencoder

I construct deep autoencoder [23], by stacking encoder and decoder architectures, which in my case would be considered as Multi-layer Perceptron (MLP) models, with a bottleneck in between. A brief description of MLP is as follows.

Multi-layer perceptron (MLP) is a recursion over a single layer perceptron (SLP) upto a finite number n. Each i^{th} SLP can be fundamentally represented by following equation.

$$\phi_{i+1} = \sigma_i(M_i\phi_i) \tag{3.1}$$

where $M_i \in \mathbb{R}^{l\times m}$ is a matrix with trainable weights, $\phi_i \in \mathbb{R}^m$, $\phi_{i+1} \in \mathbb{R}^m$ are arbitrary input and output vectors respectively. σ_i is an apporpiate activate function corresponding to i^{th} SLP. Representing $\sqcup_i$ as the recursion operator, then output of a MLP Ω can be defined as $\phi_n = \Omega(\phi_0) = \sqcup_{i=0}^{n}\sigma_i(M_i\phi_i)$.

In my approach, the word2Vec embeddings are subjected to min-max normalization, and then the dimensionality of these vectors is reduced via a deep autoencoder (MLP based) setup which efficiently learns non-linear subspace on which the original vectors can be projected in an unsupervised manner. The fundamental equation of this setup is represented by equation 3.2.

$$X = \Omega_D(\Omega_E(X)) \tag{3.2}$$

For a given input Word2Vec vector X, Ω_E representing the encoding architecture (which might as well be a MLP or other deep learning architecture) transforms it to latent space $\mathbb{R}^k$, such that $k < n_w$.

The decoder Ω_D is the complimentary neural network architecture that transforms the representations from this latent space to get back original input. At the end of associated unsupervised trained, I would avail the encodings as $\Omega_E(X)$. For my study, I only employ MLP based encoder and decoder architectures, with final activation function as *sigmoid*. Binary Crossentropy is used as the loss to train the weights associated to equation 3.2.

3.2.3 MLP based Fine-tuning

The dimensionality reduced vectors, represented by $\Omega_E(X)$, is followed by fine tuning the resultant word vectors. To achieve this, an embedding matrix $\mathbb{E}$ is constructed from the dimensionality reduced word vectors associated with the textual dataset. $\mathbb{E}$ is allowed to be trainable, and is adjoined by a MLP with softmax (or sigmoid in case of multi-label classification) as the final activation function. The input to this network would be the word tokenized texts from the textual dataset. The resultant network is fitted on training text with corresponding cyberbully labels in a supervised manner. In my dissertation, I will use MLP as a SLP. Therefore, the setup can be reduced to following formulation.

$$y = \sigma(M.\mathbb{E}X_T) \tag{3.3}$$

Where X_T represents the tokenized text, M is a weight matrix, y is output probability vector and σ is the appropiate activation function for classification (*sigmoid* or $softmax$). $\mathbb{E}$ and M are trainable matrices and the learning rate is kept low, in order to prevent overfitting and un-memorization of previous learned embeddings. In this way, I introduce the additional context of classes into the embeddings.

After training, the fine-tuned word vectors corresponding to each word, of the sentence are orderly stacked get text embedding matrices. These fine-tuned vectors of the trained equation 3.3 model can be calculated as $\mathbb{E}X_T$, where X_T would correspond to each individual word of a text. The dimension of these matrices are kept fixed via zero padding to hold same dimensions as that of maximum word count of all sentences. However, it can be zero padding can be extended to any number greater than the maximum word count as well. These text embedding matrices can be represented as $T \in \mathbb{R}^{k \times N_{max}}$, where k is the original reduced dimension, and N_{max} is the maximum number of words in a text.

3.2.4 Sinc Interpolation

Text embedding matrix $T_{k \times N_{max}}$ can be realized as 1D feature vectors over channels. I will refer to k as the 'channel' dimension and N_{max} as the feature dimension. However, one concern regarding this text embedding matrix is that T is almost zero over feature dimension for short sentences, the information associated to it is confined within few adjacent samples. This characteristic can certainly degrade the performance of sequential model that model pattern over whole sequence range. Additionally, if the dataset has greater proportion of shorter sentences as compared to longer sentences, then the model might overfit onto longer sentences, degrading the classification performance of the sequent models.

In this course, I take inspiration from a well-known filed of signal processing, the main subject of which is 'signals'. Most of the signal processing techniques, revolve around extracting signals over raw time-series, which exhibits certain properties like being smooth and dense over time domain, have a well-structured, and sparse spectrum in frequency domain. Realizing this, I attempt to extract signal features from T by making a fundamental observation of identifying cyberbullying in textual context, such that the meaning delivered by shorter sentences is same as the meaning delivered by longer sentence.

Based on these principles, propose a signal processing technique to resample the text embedding matrices over feature dimension corresponding to all texts. In this context, I employ Kaiser-window based Sinc-Interpolation [40] to stretching the low support feature vectors (and have large zero padding) over maximum support, i.e. N_{max}. Considering T_i as i^{th} vector of T along 'channel' dimension, having support N_i, I can represent this Sinc-interpolation operation as.

$$\hat{T}_i(t) = \sum_{u=0}^{N_i - 1} T_i(u) w_s(t-u) sinc(t-u) \tag{3.4}$$

where $w_s(t)$ is the Kaiser-window function and $t \in \{0, 1, .., N_{max} - 1\}$. In equation 3.4, I assume that the original vector $T_i(u)$ has sampling frequency as one, which is more logical since it confirms to interpretation of one word per position in a sentence. Through this formulation, I arrive at 1D vectors, that have much definite Fourier spectrum and have smoothness properties by the virtue of *sinc* function. In this way the resultant signal features are stacked to form final text embedding matrices $T_s = stack(\{T_i\}_{i=1}^{k}) \in \mathbb{R}^{k \times N_{max}}$. One of the main characteristics of these signal features is sparsity in Fourier domain, due to its smoothness which directly links to decay and sparsity of Fourier transform [41], in this course the resultant individual signal feature $\hat{T}_i$ can be approximated by a

trigonometric polynomial as equation 3.5.

$$\hat{T}_i(t) \approx \sum_{j=-N}^{N} a_{ij} e^{\iota b_{ij} t} \tag{3.5}$$

where $a_{ij} \in \mathbb{C}$, $b_{ij} \in \mathbb{R}$ and $N << N_{max}/2$. ι is the imaginary unit. This representation has definite temporal patterns which can be more efficiently learned by an arbitrary sequential model.

An important thing to point out that in my experiments, I set N_{max} equal to the maximum number of words found in sentences for a given dataset. In order to illustrate this, I take a small dataset of texts of which the maximum number of words per text is 800, while mostly the words per text are much lower. I plotted first channel of text embedding matrix corresponding to first 5 texts of this dataset, which is shown as left plot of figure 3.2. The first channel of consequent signal features are plotted in right plot of figure 3.2. An obvious interpertation of the Sinc-interpolation operation vividly shows that it stretches the feature confined to low support (which is less that 100) towards maximum support (800 in this case).

I may remark that signal features conform well to equation 3.5. However, an important point to be noted is that these signal feature may change depending upon the N_{max} assigned. So if I increased the considered dataset with more longer word sentences, with same assumption over N_{max} to be the maximum word count per text, then signal features would resample across broader range, and N in equation 3.5 would reduce. This will be by virtue of Fourier transform property that stretching in time domain leads to contraction of frequency spectrum in frequency domain. In this course logic, letting N_{max} to be relatively small would allows high frequency sinusoidals to be added in signal features, thus leading to noisy characteristic signal features that might deteriorate the performance of the later sequential model to learn characteristic 1D features from them.

3.2.4.1 Relative Sparsity Metric

The purpose of relative sparsity metric (RSM) is to define a metric to measure the sparsity of a given signal vector in Fourier domain.

I first of define a thresholding function $\alpha : \mathbb{R} \to \mathbb{R}$ by equation 3.6.

$$\alpha(\phi) = \begin{cases} 0, & \text{if } \phi \leq \epsilon \\ \phi, & \text{otherwise} \end{cases} \tag{3.6}$$

Then I define RSM as the amount of sparsity in a signal feature, that is invariant to length of the signal feature over a particular channel $\hat{T_i}$ in Fourier spectrum, and is defined by equation 3.7.

$$RSM(x) = \frac{\|\alpha(|FFT(\hat{T_i})|)\|_0}{N_{max}} \tag{3.7}$$

where FFT stands for Fast Fourier Transform, ϵ is some threshold to associate negligible magnitudes to zero. The RSM metric for all channels would be average to determine the overall RSM metric of signal features associated with a text. For my dissertation, I choose $\epsilon = 2 * mean(s)$. I make utility of this metric to scrutinize results in "Evaluation" chapter.

3.3 1D Approach

The 1D approach is progressed by building a classifier model whose first part performs signal fusion, i.e. to combine information from multiple channels into a singular channel, and the second part extract features from the resultant fused signal and then maps them to classification probabilities. A further resolution to 1D approach is described through next subsections.

3.3.1 Signal Fusion

The signal features of text are fused into a single signal feature via a Graph Convolution Network (GCN) architecture. A typical GCN is described below.

GCN [42] is a data-driven way to dissertation around graph data to map them towards desired outputs. The graph data is representable as $G = (V, E)$, where V denotes the set of nodes and E implies the set of edges between nodes. Each node is further associated with node feature matrix $\phi \in \mathbb{R}^{n \times m}$, where $n = |V|$ and m is the feature dimension. Each graph's connection topology can be further represented by an adjacency matrix $A \in \mathbb{R}^{n \times n}$, whose entries represented edges between nodes. Finally, the GCN extracts the output features as $z \in \mathbb{R}^{n \times d}$ from this data structure via equation 3.8, where d is the output feature dimension.

$$z = \sigma((D + I)^{-\frac{1}{2}}(A + I)(D + I)^{-\frac{1}{2}}\phi W + b) \tag{3.8}$$

Where D represents the diagonal node matrix of A (where $D_{ii} = \sum_{j=1}^{n} A_{ij}$, and $W \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^{n \times d}$ represent trainable weights which are learned from training dataset. σ represents a choice activation

function, which can range across typical activation functions like $ReLU$, $Tanh$ or $sigmoid$ functions. For the sake of brevity, I will represent equation 3.8 as $z = GCN_\sigma(x, A)$ for later consideration.

For my methodology, the node features would be the transposed signal features $T_s^\top$, and the adjacency matrix takes into account the relative positional context between segments of the sentence. Following this, I propose the adjacency matrix $A \in \mathbb{R}^{N_{max} \times N_{max}}$ in equation 3.9.

$$A_{ij} = \begin{cases} \exp(-|i-j|), & \text{if } \exp(-|i-j|) \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{3.9}$$

As per the pipeline in figure 3.1, I stack three GCN layers complimented with dropout, to avoid overfitting, alongside with learning more effective fused representation of node features as compared to one layer. While representing dropout operation (randomly thresholding some enteries of vector to zero) over arbitrary input ϕ as ϕ_D, and representing the resultant fused signal feature vector $Z \in \mathbb{R}^{N_{max}}$ would be represented via equation 3.10.

$$Z = GCN_\lambda(GCN_\gamma(GCN_\gamma(\hat{T}_D, A)_D, A)_D, A) \tag{3.10}$$

where γ, λ represent $ReLU$ and linear activation functions respectively.

3.3.2 Binding Function

I considered a trainable system, called 'binding' function, the first objective of which is to improve the temporal resolution of the resultant fused signal feature, in terms of regularizing the sharp transition regions of a sequence vector, as well increasing the density (support) of this feature temporally. This improved representation would effectively ameliorate the subsequent classifier's ability to learn temporal features.

The second objective of this function is to homogenize the temporal representation of two or more fused features associated to texts belonging to same class, that may show sharp differentials in some portion of the sequence. This second objective essentially ensures better classification accuracy and robustness during training and inference phase respectively.

In this regard, I develop a transformation, represented in equation 3.11, with trainable parameters. The name is derived in consideration of the fact this function binds two different architectures (in my

case, the feature fusion architecture and classifier architecture) serving different purposes.

$$\hat{Z} = Z_{norm} + \zeta \cos(2\pi Z_{norm} + \Theta) \tag{3.11}$$

where $\zeta \in \mathbb{R}$, $\Theta \in \mathbb{R}^{N_{max}}$ are trainable parameters. Z_{norm} represents when Z is subjected to layer normalization. In order to elaborate binding function's regularization property, I can consider following approximation under chain rule, given that ζ, Θ would be constant during inference phase.

$$\begin{aligned} \Delta_{seq}\hat{Z} \approx \Delta_{seq} Z_{norm} - 2\pi\zeta \sin(2\pi Z_{norm} + \Theta) \\ \Delta_{seq}(Z_{norm} + \frac{\Theta}{2\pi\zeta}) \end{aligned} \tag{3.12}$$

Here Δ_{seq} represents the sequence-wise change. On the right hand side of equation 3.12, if the sequence wise change of Z_{norm} is positively large, then it would be countered by the term $2\pi\zeta \sin(2\pi Z_{norm} + \Theta)\Delta_{seq}(Z_{norm} + \frac{\Theta}{2\pi\zeta})$, for trained ζ, Θ. This argument can be extended for the case of negatively large change as well, to conclude that $\Delta\hat{Z}$ is essentially stabilized. Additionally, the presence of trigonometric function in equation 3.11, increases the support of $\Delta\hat{Z}$. This possible due to smoothing property and range of cosine function over $2\pi Z_{norm}$, thus allowing to extend the locally confined information globally. This would serve beneficial in terms of reduced parameters both for later classifier chapter for Multi-headed Attention case (which would required lesser attention heads) and Convolution-Residual Wavenet Block (which would require lesser number of filters). More profoundly, there is an imposition of stationary characteristic into the resultant fused signals, to homogenise the same property over whole time range. An interesting direction to consider would be to impose constraint on decaying property of Fourier transform of $\hat{Z}$, that would regularize the total loss. But for the sake of brevity, both of these things are not considered further.

In order to elaborate the second objective of the binding function, I consider a variational approach, as to consider a set of all possible close variations of a single Z_{norm} (that maybe close enough up to small $\mu \in \mathbb{R}$ in L_1 space), which correspond to same class. Then, the change of Z_norm to its most close variation can be approximately computed from equation 3.11, via first order Taylor expansion as follows.

$$\Delta_{var}\hat{Z} \approx \Delta_{var} Z_{norm} - 2\pi\zeta \sin(2\pi Z_{norm} + \Theta)\Delta_{var} Z_{norm} \tag{3.13}$$

Here Δ_{var} represents variational change. Following the same logic as that of equation 3.12, $2\pi\zeta \sin(2\pi Z_{norm} + \Theta)\Delta_{var} Z_{norm}$ essentially regularizes any confined sharp fluctuation between Z_{norm} and its close variation (which is possible in L_1 space), thus homogenizing the temporal representation of variations with

respect to same class. This would be largely independent of learned parameter Θ, since it is only enclosed in sine function.

3.3.3 Classifier

The local and global dependencies of the output of binding function are consequently learned by two architectures approaches:

3.3.3.1 Attention System

Multi-headed attention was proposed as a countermeasure to reduced resolution to singular attention mechanism [43] by parallelizing the computation to produce independent attentions scores which are later concatenated. In this course, query (Q), key (K) and value (v) matrices are calculated from encoded sequence, which are later used to compute scaled dot-product attention via equation 3.14.

$$AT(Q, K, v) = softmax(\frac{QK^T}{\sqrt{d}})v \tag{3.14}$$

where d is the length of individual query and key vectors. Equation 3.14 computation is independently distributed in to h attention heads, where i^{th} attention head computes the attention pooling H_i by equation 3.15.

$$H_i = AT(W_i^q Q, W_i^k K, W_i^v v) \tag{3.15}$$

W_i^q, W_i^k, W_i^v are the associated trainable linear projections to Q, K and v for i^{th} attention head. The resultant outputs of each attention heads are concatenated and fed into a MLP Ω to generate final representation Y as.

$$Y = \Omega(Concat(\{H_i\}_h) \tag{3.16}$$

The Multi-headed Attention is engulfed in a ResNet [13] topology, followed by 1D-Convolution layer and Global Average Pooling [44] to produce reduced features, which are mapped towards classification probabilities via a MLP capped with $sigmoid$ or $softmax$ layer, depending upon type of classification. This scheme is also illustrated in figure 3.1.

Here 1D-convolution layer [45] operating over an input $\phi \in \mathbb{R}^n$, having one channel, can be represented by equation 3.17.

$$\phi_*(j) = \sum_{i=0}^{l} \phi(i)\xi(j - i - s + 1) \tag{3.17}$$

where $\xi \in \mathbb{R}^l$ is the trainable convolution kernel of size l, and $s \geq 1$ is the stride size.

3.3.3.2 Residual System

In this approach, the 1D-Convolution layer and Wavenet residual block are stacked over three times, and the final reduced features are mapped towards classification probabilities via a MLP. This pipeline is also demonstrated in figure 3.1.

The WaveNet residual block is the main primary constituent of the WaveNet model proposed in [46] for audio generation. The main operation in this methodology is 1D dilated causal convolutions, where the temporal dependencies are respected in convolution operation alongside with increasing the receptive field with relatively lower parameters by skipping inputs at certain time steps. The output of this convolution is followed by gated activation units (which are known to perform better than standard *ReLU* operation for audio representation) in residual topology to prevent gradient vanishing, which would allow stacking more these blocks to increase the depth of the model with expected proportional accuracy. Realizing the potential of residual block to gain features of 1D inputs, I utilize the same architecture of residual block as proposed in [46].

3.4 2D Approach

For 2D approach, I employ a different strategy than the 1D approach, where a computer vision model us used to classify a spectrogram representation of the signal features. The corresponding pipeline is represented in figure 3.1. I elaborate this strategy further through following sections.

3.4.1 STFT Spectrogram Tensor

Considering a non-stationary signal, Short-time Fourier Transform (STFT) works well in capturing the frequency domain information by extracting small stationary segments of the signal and applying Fourier transform via window function. Henceforth, for a given signal ϕ, its STFT is defined by equation 3.18.

$$STFT\{\phi(n)\}(m,\omega) = \sum_{n=-\infty}^{\infty} \phi(n)w(n-m)e^{-j\omega n} \tag{3.18}$$

where $w[n]$ is window function, more specifically Hanning window function [47] is considered in this dissertation. I construct spectrogram plot from this STFT information, by first converting the STFT

matrix into decibal scale for improve representation. These spectrogram plots are save as an $256 \times 256 \times 3$ RGB image.

In order to increase the frequency range associated to Fourier transform of a fixed sized window for this STFT spectrum, as well as inducing further smoothness into the signal features while preserving the morphology of the signals, cubic spline interpolation is performed on the original signal, to increase the sampling frequency of signal features by a factor called sampling ratio. Therefore sampling ratio can be defined as ratio of samples after cubic spline interpolation and before cubic spline interpolation. For my dissertation, I will mainly consider sampling ratio as 2.

Before availing the STFT spectrogram images, I first perform cubic spline interpolation, as per the pipeline in figure 3.1. This will have two advantages. One, that it will increase the frequency range associated to Fourier transform of a fixed sized window for this STFT spectrum thus allow the STFT representation to be more compact and distinctive with respect to classes. Second, it will induce further smoothness properties into the signal features, thus allowing a cleaner STFT spectrum, conforming to morphology of the signal features.

Since there would be k channels associated with the signal features, therefore a total of k RGB images are produced. The dimensionality of these RGB images is reduced via greyscale transformation, and the resultant greyscale images are stacked to form the final STFT spectrogram tensor $I \in \mathbb{R}^{256 \times 256 \times k}$ to be classified by an image classifier model, discussed in next section.

3.4.2 Image Classifier

I employ PatchConvNet [13] to learn features from the final image tensor I, and the classification probabilities are determined via final softmax or sigmoid layer depending upon the type of classification.

PatchConvNet proposed in [13] replaces the previous pyramidal scheme of CNNs to arrive at a globally inferring solution from image data. In context of classification, this model is divided into first preprocessing by "convolution stem", spatial processing by "convolution trunk" and final aggregation of projected patches based on their inter-similarity and class information for determining logits vector by "attention pooling". This architecture allows competitive accuracy in relationship with the peak memory, but at cost of greater floating-point operations as compared to contrary hierarchical vision models. Lastly, PatchConvNet's capability to effectively encode the relationship between feature-dense regions of image and probability vectors, by virtue of its attention pooling, proves a suitable choice for learning characteristic shapes emergent in STFT spectrograms.

Chapter 4

Evaluation

4.1 Datasets

I consider following two datasets for my experimentation study, and they are also described in table 4.2.

4.1.1 Single-Label Multi-Class

I utilize dataset proposed in [8] that utilized dynamic query expansion to get text samples from Twitter with respect to several cyberbully contexts targetting topics like age, ethnicity, gender and other domain. In this course, the dataset has a total of six labels namely 'age', 'gender', 'ethnicity', 'religion', 'other_cyberbully' and 'other_cyberbully'. The fitting of a model over this dataset boils down to single-label multi-class classification problem. As discussed in Chapter 3, since N_max plays a great role in performance of the proposed scheme, the histogram of word count per text is shown for dataset-1 in figure 4.1.

4.1.2 Multi-Label Multi-Class

I also utilized the Kaggle's toxic comment classification challenge dataset, where the labels of datasets are 'toxic', 'severe_toxic', 'obscene', 'threat', 'insult' and 'identity_hate'. I add the additional label of 'non-toxic', if all other categories are nill corresponding to a text. This dataset comprises of Wikipedia talk page edit evaluated through human moderator to different levels of toxicity, as identified by the labels. The fitting of a model over this dataset is equivalent to multi-label multi-class classification

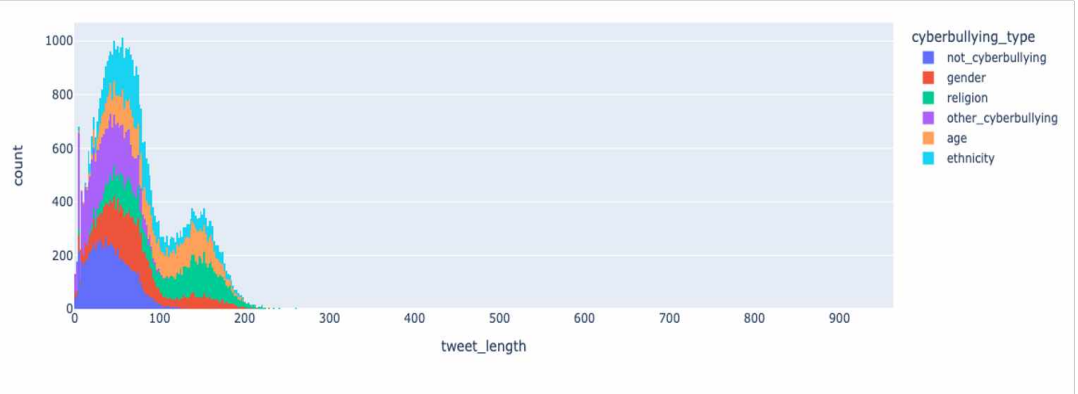

FIGURE 4.1: Histogram of word count per text in dataset-1

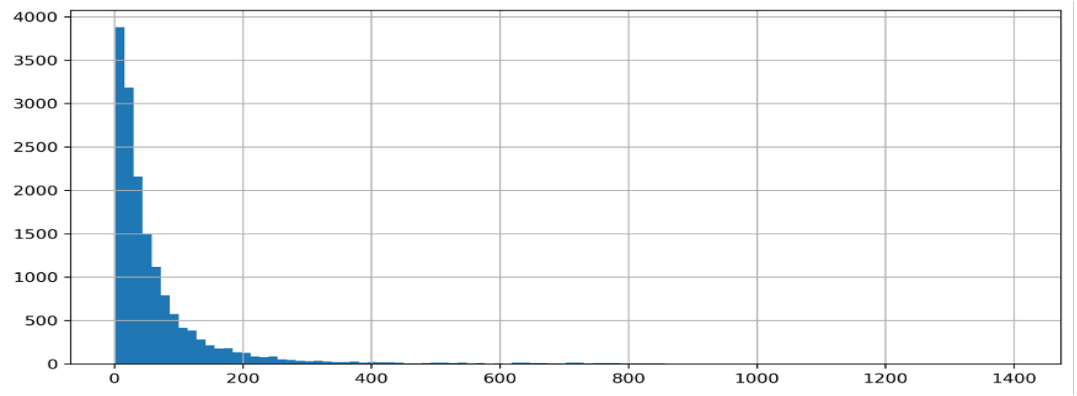

FIGURE 4.2: Histogram of word count per text in dataset-2

TABLE 4.1: Datasets utilized for study

No.	Name	Nature	Classes	Ref
1	Fine Grained Cyberbullying Dataset	Single-label, Multi-class	6	[48]
2	Toxic Comment Classification Dataset	Multi-label, Multi-class	7	[49]

problem, since each text in this dataset can have different levels of toxicity severity. The histogram of word count per text is shown for dataset-2 in figure 4.2.

4.2 Performance Metric

I measure the classification performance of the proposed approaches by utilizing scores like classification accuracy [8], precision [4], recall [4] and F_1 [4] scores for both single-label and multi-label classification cases. For multi-label case, I also adopt an additional metric called Hamming loss, that accounts for fractions of labels incorrectly predicted, which is more suitable for multi-label, multi-class classification problem.

TABLE 4.2: Description of models used in comparison. TP stands for trainable parameters count, PP stands for pre-trained parameters count, and LR stands for learning rate

ID	Description	TP	PP	LR	Epochs	Ref
M_1	BERT-MLP	$5.26\text{x}10^5$	$1.09\text{x}10^8$	0.001	50	[27]
M_2	DistillBERT-MLP	$5.26\text{x}10^5$	$6.68\text{x}10^7$	0.001	50	[27]
M_3	SentenceBERT-GCN	$1.35\text{x}10^4$	$2.25\text{x}10^7$	0.001	50	[8]
M_4	SentenceBERT-GAN	$8.31\text{x}10^6$	$2.25\text{x}10^7$	0.001	50	[30]
M_5	GCN-Attention (1D)	$1.38\text{x}10^5$	0	0.001	50	-
M_6	GCN-Residual (1D)	$2.31\text{x}10^6$	0	0.001	50	-
M_7	PatchConvnet (2D)	$1.38\text{x}10^6$	0	0.001	50	-

TABLE 4.3: Accuracy (A), Precision (P), Recall (R) and F_1 (F) percentages comparison for methods described in table 4.1 over subsets of dataset-1

Metric	M_1	M_2	M_3	M_4	M_5	M_6	M_7
250 samples per label (N_{max} = 40)							
A	70.95	72.97	66.67	73.66	76.35	74.66	57.72
P	72.85	78.96	65.67	72.92	76.95	74.91	57.99
R	70.55	73.34	66.89	74.15	75.2	73.51	58.23
F	71.18	70.64	64.28	73.24	75.3	73.83	56.74
500 samples per label (N_{max}= 144)							
A	73.46	73.46	68.8	76.24	83.80	83.08	68.16
P	72.74	76.22	71.5	75.39	84.17	83.52	66.31
R	73.76	72.06	69.91	75.57	84.29	83.58	66.15
F	72.92	72.10	69.97	75.42	84.87	83.43	64.41
1000 samples per label (N_{max} = 252)							
A	71.97	73.41	64.21	75.82	85.41	84.22	82.01
P	75.04	73.30	65.16	76.00	85.14	85.11	86.22
R	72.16	72.47	65.74	75.91	85.15	84.24	81.71
F	71.05	72.28	64.91	75.78	84.46	84.25	82.01
1500 samples per label (N_{max} = 252)							
A	72.94	68.76	67.47	79.14	88.30	87.36	88.82
P	74.29	69.62	66.41	79.75	88.05	87.17	88.95
R	73.78	68.09	66.81	79.46	88.38	87.45	88.85
F	73.18	67.92	65.02	79.57	88.15	87.28	88.47
2000 samples per label (N_{max} = 252)							
A	72.63	73.49	70.19	77.82	87.42	86.37	87.96
P	72.84	74.5	71.69	77.71	87.46	86.18	88.08
R	73.05	74.42	70.39	77.75	87.39	86.38	87.79
F	72.60	74.56	69.55	77.64	87.52	86.18	87.73

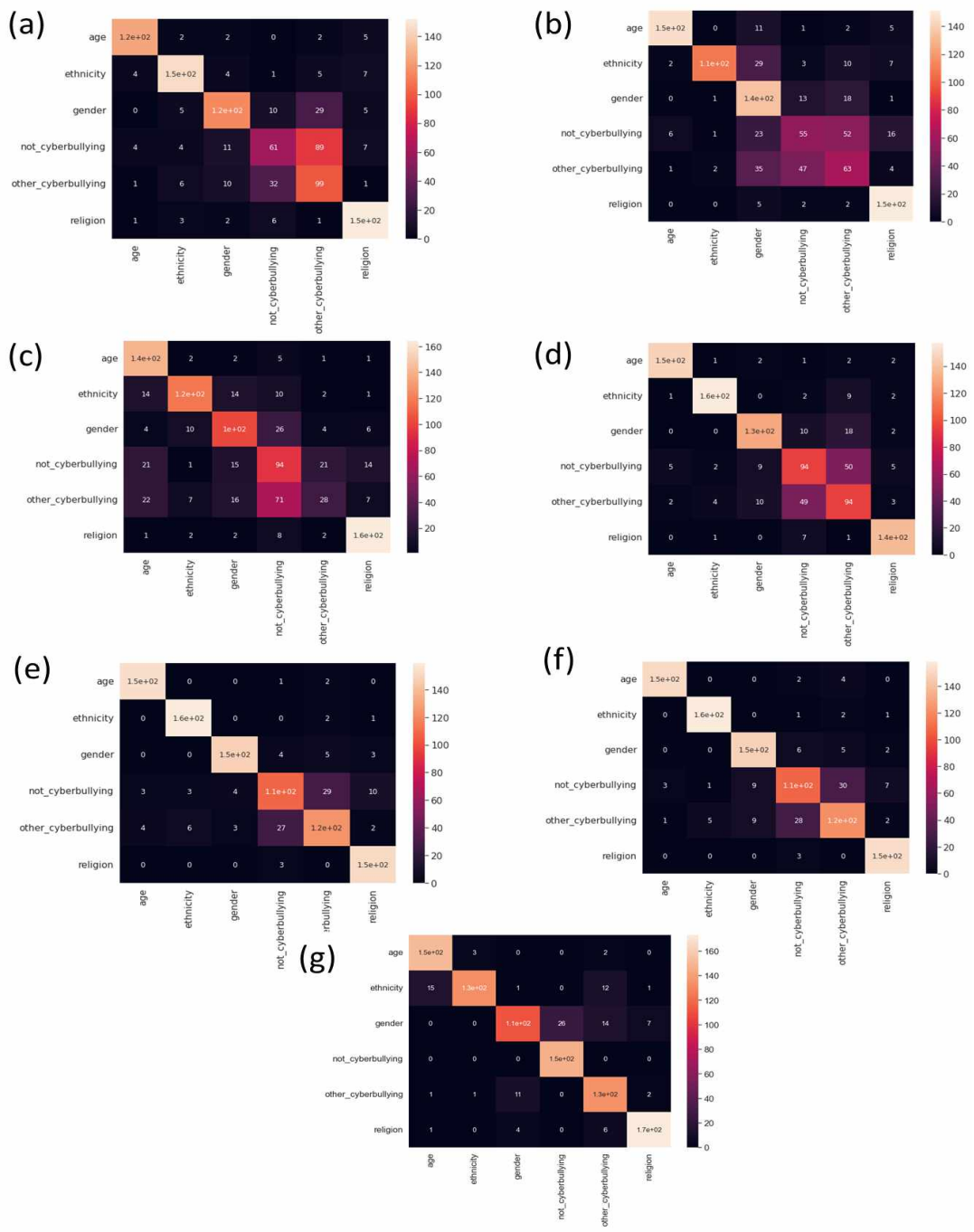

FIGURE 4.3: Confusion matrices of predictions of (a) M_1, (b) M_2, (c) M_3, (d) M_4, (e) M_5, (f) M_6 and (g) M_7 models over test dataset, when trained on 1500 samples per label subset of dataset-1

4.3 Comparison Models

The state of the art cyberbullying classification models in literature can be decomposed into feature extractors (responsible for determining textual embeddings) and classifier (responsible for mapping textual embeddings towards classification probabilities) sections.

TABLE 4.4: Hamming Loss (H), Precision (P), Recall (R), F_1 (F) and Accuracy (A) scores comparison for methods described in table 4.2 over subsets of dataset-2 comprising total 2000 and 4000 samples

Models	H	P	R	F	A
2000 Samples ($N_{max} = 797$)					
M_1	0.0675	0.8218	0.8472	0.8282	0.8078
M_2	0.0534	0.853	0.8587	0.8533	0.8177
M_3	0.0802	0.7988	0.8054	0.7981	0.7832
M_4	0.0679	0.818	0.8806	0.8333	0.7851
M_5	0.0953	0.8004	0.8004	0.8004	0.8005
M_6	0.0891	0.8122	0.8322	0.8128	0.8208
M_7	0.0879	0.8226	0.8226	0.8226	0.8226
4000 Samples ($N_{max} = 834$)					
M_1	0.0431	0.8781	0.9034	0.8841	0.8659
M_2	0.0398	0.8950	0.9102	0.9003	0.8686
M_3	0.0798	0.8071	0.8049	0.7989	0.7908
M_4	0.0494	0.8668	0.9056	0.8764	0.8455
M_5	0.0337	0.9262	0.9305	0.9244	0.8726
M_6	0.0360	0.9341	0.9320	0.9262	0.8512
M_7	0.0459	0.8940	0.9101	0.8976	0.8659

4.3.1 Feature Extractors

I consider following state of the text embedding extraction models considered in cyberbullying literature.

4.3.1.1 BERT

Bi-directional Encoder Representation form Transformers (BERT) have known to be successful in extracting robust context aware embeddings from text. Likewise to cyberbullying literature, I utilize BERT to get textual embeddings in a pretrained setting, where the model has been pretrained on uncased English langauge dataset (Toronto BookCorpus and Wikipedia [27]) for masked-language modelling and next sentence prediction. The total number of parameters of this model are around 109 million.

4.3.1.2 DistilBERT

DistilBERT was proposed as a less parameter version of BERT (around 40% less), thus allow better FLOP efficiency, at advantage of nearly equivalent performance to that of BERT. DistilBERT is pretrained on same dataset as BERT in my study. The total number of parameters of this model are around 66 million.

4.3.1.3 SentenceBERT

SentenceBERT was proposed due to unsuitability of BERT embeddings for semantic similarity search, by performing modification of pre-trained BERT via simaese network. The model considered in my dissertation produces 384 dimensional embeddings, and the description for pretraining and fine-tuning is provided in [50]. The total number of parameters of this model is more than 22 million.

4.3.2 Classifier

I consider following state of the art classifiers utilized in cyberbullying literature to map text embeddings towards classification probabilities.

4.3.2.1 MLP

I consider a 3-layered multi-layer perceptron to map the learned features, from the selected feature extractor, towards output. In my dissertation, I consider BERT and DistilBERT as the augmented feature extractors to MLP.

4.3.2.2 GCN

I employ the GCN in the same framework as [8], where each node of input graph data is represents the the text, and edges are represented by cosine similarity between the text embeddings associated to each node. In this course, I use GCN layer to extract graph features, and a final softmax (or sigmoid layer for multi-label classification) to map the features to probabilities. For determining textual embeddings, I use SentenceBERT.

4.3.2.3 GAN

I employ Graph Attention Network (GAN) in the node classification setting, in the same context as [30], except I only take into account the textual data and not user data as well due to lack of its availability in utilized datasets. Likewise to GCN case, the final classification probabilities are mapped by a softmax or sigmoid layer depending on the multi-class classification case.

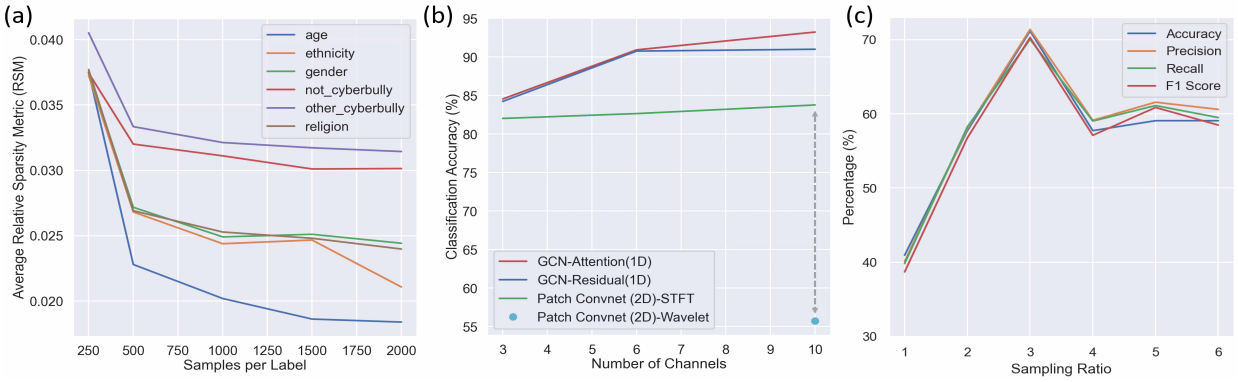

FIGURE 4.4: (a) Average RSM with respect to classes of dataset-1 vs size of subsets of dataset-1 (b) Categorical Classification accuracy for proposed model over variation in number of channels (c) Impact of classification performance of proposed 2D approach by changing the sampling ratio

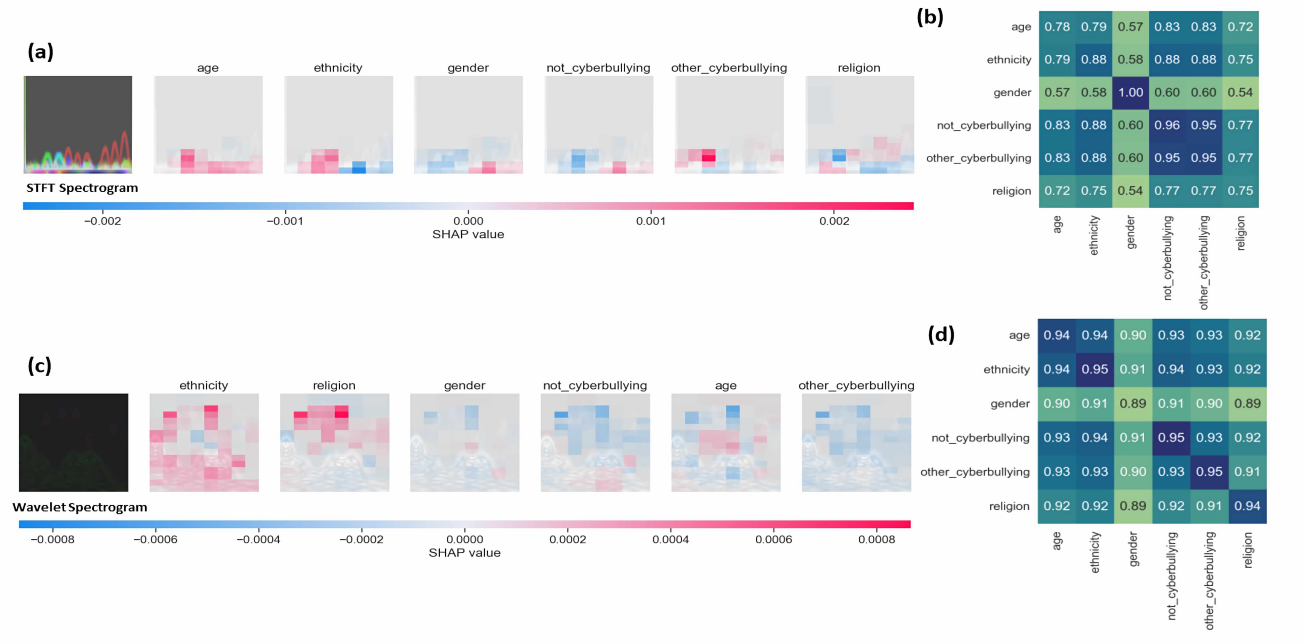

FIGURE 4.5: (a) STFT image tensor and its SHAP values plots over pixel with respect six classes of dataset-1 (b) Average MS-SSIM matrix showing similarily of STFT image tensors between classes of dataset-1(c) Wavelet image tensor and its SHAP values plotted over pixels with respect six classes of dataset-1 (b)) Average MS-SSIM matrix showing similarily of Wavelet image tensors between classes of dataset-1

I perform the combination of feature extractor and classifier model to compose benchmark models that are tabulated in table 4.3, where the benchmark models are represented with ID M_1, M_2, M_3 and M_4.

4.4 Results

I evaluated the comparison of models mentioned in table 4.3 over both datasets mentioned in table 4.2. The main comparison are tabulated in table 4.2 and 4.3. Due to limited available computation resources, I did not computed and compared performance over whole dataset, the size of which is more than 4.8×10^4 and 1.6×10^5 for dataset-1 and 2 respectively. Instead in my experimentations, I focused on evaluated how these models perform over lower size datasets, and how this performance scales as the dataset is incremented.

4.4.1 Single-label, Multi-class classification

To scrutinize single class, multi-class classification performance, I trained my proposed approach and benchmark models mentioned in table 4.3 on 50 epochs each, adjusted learning rate to be 1×10^{-3} and used ADAM as the optimizer for minimizing Categorical Crossentropy loss. Subsets of dataset-1 of sizes 250, 500, 1000, 1500 and 2000 samples per label. In this course, I trained and tested the models by dividing the dataset into training, validation and testing categories in 70, 20 and 10 % division respectively with random shuffling to ensure balanced classes. Furthermore, I made sure to prevent overfitting in all of these models, by adjusted the amount of dropout and other hyperparameters.

The benchmark models (M_1 to M_4) show nearly saddle performance as the dataset size is increasing. The proposed 1D model's (M_5 and M_6) excel in performance not only in small size datasets, but also the performance improves significantly as the dataset size increases. Remarkably, M_7 provides less evaluation scores at lowest size dataset, but the improvement in performance is most significant as compared to 1D counterparts, such that the 2D approach shows best performance at larger size datasets. One common trend has been that the performance scores for 1500 samples per label case are greater than the 2000 samples per label case, which I attribute is due to annotation error [8].

I demonstrated the confusion matrices for table 4.3 models, corresponding to 1500 samples per label subset of dataset-1, in figure 4.3. A common trend among these models is the confusion of "not_cyberbully" and "other_cyberbully" classes, the causality of which is restricted to the annotation error existent in the dataset [8]. Even in this scenario, the proposed models shows lesser extent of confusion in this case, with overall much more diagonalized matrices as compared to benchmark models. Remarkably, the 2D approach confuses "not_cyberbully" and "other_cyberbully" classes with other classes, while 1D approaches confuses these classes among themselves.

4.4.1.1 Effect of Fourier Sparsity of Signal Features

As discussed in "Sinc-Interpolation" that the sparsity of signal features plays a vital role in allowing a model to approximate it towards output, as in greater Fourier domain sparsity would lead to more robust model predictions and the contrary case reduces robustness due to addition of noise characteristics. I make use of information of N_max provided in table 4.4 (as discussed in "Methodology" chapter, I choose N_{max} as the maximum number of words per text), to empiricially demonstrate this fact. Since N_{max} is overall increasing, therefore the Sinc-interpolation operation allows expansion in time domain, which implies contraction of frequency spectrum along frequency axis. Therefore, increasing the size of dataset, fundamentally increases the sparsity of the signal features in frequency domain, which consequenlty increases the performance of proposed models.

I further empirically illustrate this principle by plotting average RSM for each class of dataset-1 over different subsets and the results are shown in figure 4.4(a). As the average RSM per label is decreasing over increasing the size of dataset, correspondingly correlates with the boost of the classification scores of 1D-models.

4.4.1.2 STFT Spectrogram Interpretability

For analysis of proposed 2D-model, I utilize SHAP (SHapley Additive exPlanations) values [51], which are one of the most ubiquitous model interpretability techniques, for explaining how the features of images are impacting model's outputs. The relationship between how much pixels of images are impacting the model's output probability vector is metricized by SHAP values.

I focus on the 500 labels per sample case of dataset-1, and for the sake of comparison also include replacement STFT spectrogram with wavelet (second order generalized Morse wavelet) [52] spectrograms and analyze the impact on PatchConvNet performance. I plotted SHAP values for a random test STFT and wavelet spectrogram based Image tensor (comprising of three channels) over 50 epochs trained PatchConvNet model with ADAM used to optimize the parameters at learning rate of 1×10^{-3}. The results are shown in figure 4.5(a) and (c) for STFT and wavelet cases respectively. I also determined the average Multi-scale Structural Similarity Index (MS-SSIM) [53] between the STFT and wavelet spectrogram image tensors to determine the average amount of correlation within the class and between classes, and the corresponding correlation matrix is shown in figure 4.5(b) and (d) respectively.

It should be noted that the spectrograms represented in figure 4.5(a) and (b) are infact stacked-greyscale tensors, and not the original RGB images that act as direct input to the PatchConvNet model. In the case of STFT spectrogram, the 2D-model gains macro precision, recall and F_1 scores of 0.6681, 0.6832 and 0.6723 respectively. On the other hand, same classifier models achieves respective scores of 0.5077, 0.5377 and 0.4778 for the wavelet spectrogram case.

It is evident from figure 4.5(a) and (c) that the STFT spectrogram provides much reduced vision field as compared to wavelet spectrogram. On the other hand, figure 4.5(b) and (d) show that wavelet spectrograms are much more correlated not only within the same class samples, but also between different class samples. That explains relative low classification scores of the model over wavelet spectrogram case, that the model struggles identify distinct features with respect to classes.This would be imminent, since the signal features exhibit smootheness (no transient characterisitcs) and sparse frequency spectrum properties (confinement in frequency space), therefore STFT spectrogram would excel in producing distinct representations of such features. But the reduced vision field of STFT spectrogram provides room for less learnable features for 2D-model to map towards outputs, which fundamentally highlights why the M_7 model somewhat lags in classification scores for 500 samples per label case, in comparison table 4.4.

4.4.1.3 Sampling Ratio Impact

Sampling ratio is one of the free hyperparameters of proposed scheme in relationship to the 2D approach. Since it was already described that cubic spline interpolation increases the frequency range of the FFT spectrum, associated to corresponding to STFT plot matrix, therefore it certain control the confinement of class respective features on the 256×256 grid image. Increasing the sampling ratio associated to the cubic spline interpolation should allow the correpnding decibel scale STFT spectrum to achieve more distinct and compact representation. However, this trend could only persist until the distinctive features converge over minimal pixel portion for the same 256×256 grid, and this will certainly lead to deterioration of the PatchConvNet classifer. I empirically depict this fact by plotting the macro-classification accuracy M_7 by over course of changing sampling ratio as argument to the cubic spline interpolation in figure 4.4(c).

4.4.1.4 Number of Channels Impact

It is documented in literature that increasing the number of channels of input feature increases the accuracy of deep learning models like convolution models [54, 55]. Since my proposed approach allows

the number of channels k as a free hyperparameter, therefore an understanding of impact of channel variations to model's accuracy is need. In this regard, I utilized the 1000 samples per label subset of dataset-1, and plotted the classification accuracy over validation set, for the three proposed models trained over 50 epochs. The corresponding plot is shown in figure 4.4(b). It can be observed that for proposed models, the classification accuracy is nearly proportional to the number of channels. However, the 2D approach shows relatively lower accuracy to channel ratio, as compared to 1D approaches. However, increasing number of channels comes at cost of greater FLOP cost.

I also added the classification accuracy over 10-channel for the case of wavelet spectrogram (second order generalized wavelet) alongside STFT spectrogram to assert that the lower accuracy to channel ratio is purely dependent upon the STFT spectrogram plots as representation of the 1D signal features. These performance controversy with respect to spectrograms is already illustrated in second previous chapter.

Additionally for the sake of demonstration, the intermediate outputs of the proposed feature extraction scheme are shown in figure 4.6.

4.4.2 Multi-label, Multi-class classification

Likewise to dataset-1, I considered subsets of dataset-2 of increasing size (original dataset size goes to around 160000 texts). In this course, I considered dataset of size 2000 and 4000 samples, and divided the dataset into training, validation and testing sets in 70%, 20% and 10% respectively, after random shuffling. I documented Hamming loss, Precision, Recall, F_1 and Accuracy scores for the models described in table 4.3, and the results are tabulated in table 4.4.

While for 2000 samples, the performance scores of proposed models (M_5, M_6, M_7) are in proximity of benchmark models, with proposed 2D approach (M_7) performing the best. This partly due to the fact that N_{max} is relatively small. On increasing the dataset size to 2000 samples, there is performance jump by atmost 13%, atleast for the proposed 1D-models (M_5 and M_6), leaving behind benchmark models. On the other hand, the 2D approach shows better performance than most benchmark models on par with M_2.

4.4.2.1 Effect of Binding Function

In order to illustrate the effectiveness of binding function (a main constituent of 1D model approaches), I plotted the intermediate output of trained GCN-Attention Model (M_5) before and after the binding

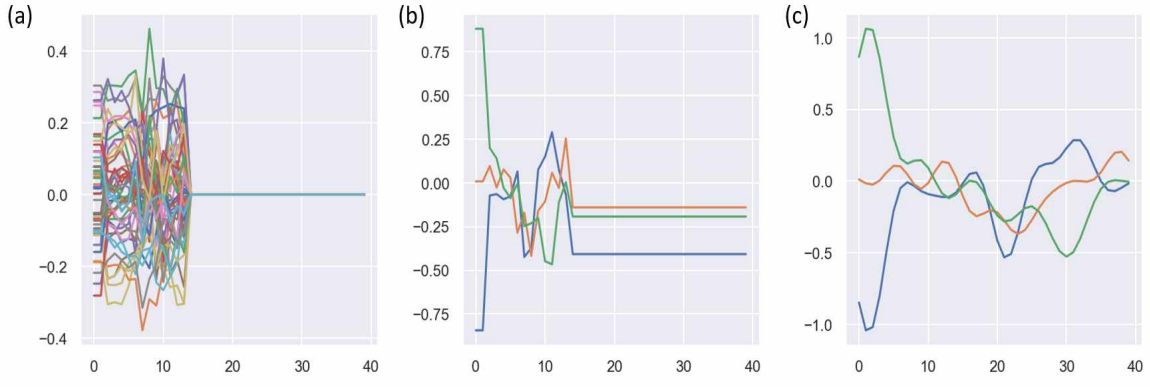

FIGURE 4.6: (a) Stacked Word2Vec vectors with zero padding corresponding to an arbitrary cyberbully text (b)Text Embedding Matrix along channels (c) Signal features

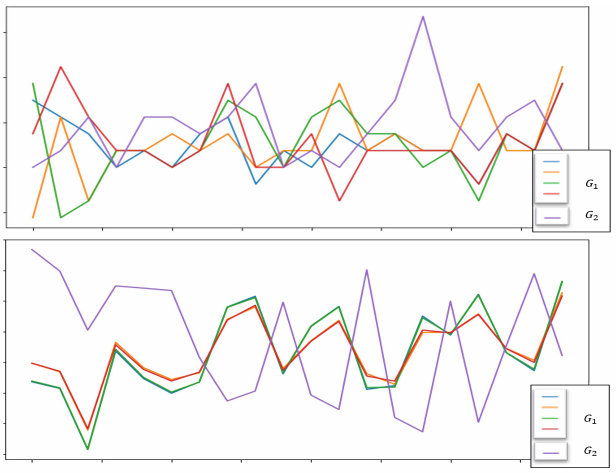

FIGURE 4.7: Top plot shows intermediate input to the binding function (of trained M_1 over 2000 samples subset of dataset-2). Bottom plot show intermediate output of the binding function

function, and the results are shown in figure 4.7. The left plot shows several sequences, either belonging to G_1 or G_2, to the output of Normalization layer (as input to binding function). Here G_1 and G_2 represent groups with same multi-label vector, and thus associated sequences to G_1 and G_2 belong to same class. It is clear from left plot that the characteristic morphology of particular class is not evident to allow, lets say a MLP, to find a non-linear projection where they are separable by a hyperplane. Additionally, some sequences shows sharp transient response for the case of G_2, which causes the associated temporal information confined to a small region.

The outputs of binding functions are represented in right plot of figure 4.7. The characteristic morphology for G_1 and G_2 is clearly visible and distinct, with the morphology in individual classes being uniform. This is the consequence of the second objective of binding function, as discussed in section 'Binding Function'. The amplitudes are uniformly distributed over the sequence range, a consequence of first objective of binding function, thus allowing a later sequence learning model to efficiently learn them (Multi-head attention in this case).

Chapter 5

Conclusion

In this dissertation, I proposed a novel data-driven fine-grained cyberbully text classification approach, by reinterpreting the problem of text classification into a 1D signal classification problem. More specifically, I extend the well-known Skip-Gram Word2Vec model word embeddings, via a Deep Autoencoder, Multi-layer perceptron based fine-tuning, and Sinc-interpolation to transform the text into signal features, that are dense in the time domain, but sparse in the frequency domain. Leveraging these signal features, I further proposed two 1D models and one 2D-model as means to fit upon these features. The 1D models comprise of Graph Convolution Network, which functions to combine signals over channel. Later the Attention system or Residual system learns the temporal features of the fused signal and maps them towards classification probability. For 1D-model, I also proposed an additional layer called "binding function" whose purpose is to denoise and adjust uneven amplitudes as the result of signal fusion. For 2D approach, I utilized STFT spectrograms of the signal features, which are mapped to classification probabilities via PatchConvNet model. These approaches were validated by four benchmark models and two datasets from cyberbullying literature, for two types of multi-class classification tasks. While proposed models have shown great potential in showing competitive performance to text embedding-based approaches proposed in the literature, where 1D approaches perform better as compared to 2D approach on some conditions like lower sized datasets and vice versa in other conditions like larger sized datasets and sufficient sparsity (measured by N_{max} and RSM). Needless to say that the total number of parameters of the benchmark model is 100 to 1000 times larger than the proposed approaches leading to much lesser inference computations as compared to benchmark approaches.

Chapter 6

Future Works

I consider two limitations of this research dissertation. One that the dataset utilized in this dissertation is not very large. In this dissertation, I utilized different sized subsets of two cyberbullying datasets researched in cyberbullying literature, to draw important conclusions. Second that there are many oppertunities of improvement to the proposed approaches, which I consider for my future dissertation in this regard.
In consideration of these limitations, I consider following directions to improve upon this research work:

- By forecasting the signal features by a forecasting model would allow to extrapolate the data. This would be useful to increase the frequency resolution of the FFT algorithm, thus is expected to improve the 2D signal features via STFT spectrogram.
- Reducing the number of rows in STFT matrix by reducing the length of the window, thus allowing the spectrum to cover the maximum pixels of the image dimension, but this would at the cost of poor resolution in the frequency domain.
  Therefore, finding the non-linear transformation such that the frequency resolution is preserved while reducing the window length might prove beneficial in improving the STFT representation.
- By regularizing the training loss of proposed 1D-models via adding a constraint on the sparsity of Fourier spectrum of the output of binding function might further improve the performance of 1D-models.
- Data augmentation to increase the size of available cyberbullying datasets via text generative models would allow measuring the generalizability over real-world data.

Bibliography

[1] Neetu Rani, Prasenjit Das, and Amit Kumar Bhardwaj. Rumor, misinformation among web: a contemporary review of rumor detection techniques during different web waves. *Concurrency and Computation: Practice and Experience*, 34(1):e6479, 2022.

[2] Meaghan C McHugh, Sandra L Saperstein, and Robert S Gold. Omg u# cyberbully! an exploration of public discourse about cyberbullying on twitter. *Health Education & Behavior*, 46(1): 97–105, 2019.

[3] Cynthia Van Hee, Gilles Jacobs, Chris Emmery, Bart Desmet, Els Lefever, Ben Verhoeven, Guy De Pauw, Walter Daelemans, and Véronique Hoste. Automatic detection of cyberbullying in social media text. *PloS one*, 13(10):e0203794, 2018.

[4] Spiros V Georgakopoulos, Sotiris K Tasoulis, Aristidis G Vrahatis, and Vassilis P Plagianakos. Convolutional neural networks for toxic comment classification. In *Proceedings of the 10th hellenic conference on artificial intelligence*, pages 1–6, 2018.

[5] Semiu Salawu, Yulan He, and Joanna Lumsden. Approaches to automated detection of cyberbullying: A survey. *IEEE Transactions on Affective Computing*, 11(1):3–24, 2017.

[6] SV Drishya, S Saranya, JI Sheeba, and S Pradeep Devaneyan. Cyberbully image and text detection using convolutional neural networks. *CiiT Int. J. Fuzzy Syst*, 11(2):25–30, 2019.

[7] JI Sheeba and S Pradeep Devaneyan. Impulsive intermodal cyber bullying recognition from public nets. *International Journal of Advanced Research in Computer Science*, 9(3), 2018.

[8] Jason Wang, Kaiqun Fu, and Chang-Tien Lu. Sosnet: A graph convolutional network approach to fine-grained cyberbullying detection. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1699–1708. IEEE, 2020.

[9] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33:17766–17778, 2020.

[10] Lei Yang and Hongdong Zhao. Sound classification based on multihead attention and support vector machine. *Mathematical Problems in Engineering*, 2021, 2021.

[11] Sandeep Kumar Pandey, Hanumant Singh Shekhawat, and SRM Prasanna. Emotion recognition from raw speech using wavenet. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pages 1292–1297. IEEE, 2019.

[12] Rui Zhao, Anna Zhou, and Kezhi Mao. Automatic detection of cyberbullying on social networks based on bullying features. In *Proceedings of the 17th international conference on distributed computing and networking*, pages 1–6, 2016.

[13] Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Piotr Bojanowski, Armand Joulin, Gabriel Synnaeve, and Hervé Jégou. Augmenting convolutional networks with attention-based aggregation. *arXiv preprint arXiv:2112.13692*, 2021.

[14] Salim Alami and Omar Elbeqqali. Cybercrime profiling: Text mining techniques to detect and predict criminal activities in microblog posts. In *2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pages 1–5. IEEE, 2015.

[15] April Kontostathis, Kelly Reynolds, Andy Garron, and Lynne Edwards. Detecting cyberbullying: query terms and techniques. In *Proceedings of the 5th annual acm web science conference*, pages 195–204, 2013.

[16] Pete Burnap and Matthew L Williams. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & internet*, 7(2): 223–242, 2015.

[17] Kun Wang, Yanpeng Cui, Jianwei Hu, Yu Zhang, Wei Zhao, and Luming Feng. Cyberbullying detection, based on the fasttext and word similarity schemes. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 20(1):1–15, 2020.

[18] Steven Zimmerman, Udo Kruschwitz, and Chris Fox. Improving hate speech detection with deep learning ensembles. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*, 2018.

[19] Gretel Liz De la Pena Sarracén, Reynaldo Gil Pons, Carlos Enrique Muniz Cuza, and Paolo Rosso. Hate speech detection using attention-based lstm. *EVALITA evaluation of NLP and speech tools for Italian*, 12:235, 2018.

[20] Nijia Lu, Guohua Wu, Zhen Zhang, Yitao Zheng, Yizhi Ren, and Kim-Kwang Raymond Choo. Cyberbullying detection in social media text based on character-level convolutional neural network with shortcuts. *Concurrency and Computation: Practice and Experience*, 32(23):e5627, 2020.

[21] Raunak Joshi and Abhishek Gupta. Performance comparison of simple transformer and res-cnn-bilstm for cyberbullying classification. *arXiv preprint arXiv:2206.02206*, 2022.

[22] Jianwei Zhang, Taiga Otomo, Lin Li, and Shinsuke Nakajima. Cyberbullying detection on twitter using multiple textual features. In *2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST)*, pages 1–6. IEEE, 2019.

[23] Rui Zhao and Kezhi Mao. Cyberbullying detection based on semantic-enhanced marginalized denoising auto-encoder. *IEEE Transactions on Affective Computing*, 8(3):328–339, 2016.

[24] Lu Cheng, Ruocheng Guo, Yasin N Silva, Deborah Hall, and Huan Liu. Modeling temporal patterns of cyberbullying detection with hierarchical attention networks. *ACM/IMS Transactions on Data Science*, 2(2):1–23, 2021.

[25] Suyu Ge, Lu Cheng, and Huan Liu. Improving cyberbullying detection with user interaction. In *Proceedings of the Web Conference 2021*, pages 496–506, 2021.

[26] Sudhanshu Mishra, Shivangi Prasad, and Shubhanshu Mishra. Multilingual joint fine-tuning of transformer models for identifying trolling, aggression and cyberbullying at trac 2020. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 120–125, 2020.

[27] Tasnim Ahmed, Shahriar Ivan, Mohsinul Kabir, Hasan Mahmud, and Kamrul Hasan. Performance analysis of transformer-based architectures and their ensembles to detect trait-based cyberbullying. *Social Network Analysis and Mining*, 12(1):1–17, 2022.

[28] Bandeh Ali Talpur and Declan O'Sullivan. Multi-class imbalance in text classification: A feature engineering approach to detect cyberbullying in twitter. In *Informatics*, volume 7, page 52. MDPI, 2020.

[29] Nabi Rezvani, Amin Beheshti, and Alireza Tabebordbar. Linking textual and contextual features for intelligent cyberbullying detection in social media. In *Proceedings of the 18th International Conference on Advances in Mobile Computing & Multimedia*, pages 3–10, 2020.

[30] Joan Plepi and Lucie Flek. Perceived and intended sarcasm detection with graph attention networks. *arXiv preprint arXiv:2110.04001*, 2021.

[31] Devin Soni and Vivek Singh. Time reveals all wounds: Modeling temporal dynamics of cyberbullying sessions. In *12th International AAAI Conference on Web and Social Media, ICWSM 2018*, pages 684–687. AAAI press, 2018.

[32] Vimala Balakrishnan, Shahzaib Khan, and Hamid R Arabnia. Improving cyberbullying detection using twitter users' psychological features and machine learning. *Computers & Security*, 90: 101710, 2020.

[33] Krishanu Maity, Abhishek Kumar, and Sriparna Saha. A multi-task multi-modal framework for sentiment and emotion aided cyberbully detection. *IEEE Internet Computing*, 2022.

[34] Hugo Rosa, David Matos, Ricardo Ribeiro, Luisa Coheur, and João P Carvalho. A "deeper" look at detecting cyberbullying in social networks. In *2018 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2018.

[35] Lihao Ge and Teng-Sheng Moh. Improving text classification with word embedding. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 1796–1805. IEEE, 2017.

[36] Lingfei Wu, Ian EH Yen, Kun Xu, Fangli Xu, Avinash Balakrishnan, Pin-Yu Chen, Pradeep Ravikumar, and Michael J Witbrock. Word mover's embedding: From word2vec to document embedding. *arXiv preprint arXiv:1811.01713*, 2018.

[37] Joseph Lilleberg, Yun Zhu, and Yanqing Zhang. Support vector machines and word2vec for text classification with semantic features. In *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pages 136–140. IEEE, 2015.

[38] Ankit Thakkar, Dhara Mungra, Anjali Agrawal, and Kinjal Chaudhari. Improving the performance of sentiment analysis using enhanced preprocessing technique and artificial neural network. *IEEE transactions on affective computing*, 2022.

[39] Nltk wordnet. https://www.nltk.org/_modules/nltk/stem/wordnet.html.

[40] Jesus Selva. Convolution-based trigonometric interpolation of band-limited signals. *IEEE transactions on signal processing*, 56(11):5465–5477, 2008.

[41] Junaid Iqbal Khan and Usman Zabit. On two fourier transform-based methods for estimation of displacement and parameters of self-mixing interferometry over major optical feedback regimes. *IEEE Sensors Journal*, 21(9):10610–10617, 2021.

[42] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[44] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[45] Jianfeng Zhao, Xia Mao, and Lijiang Chen. Speech emotion recognition using deep 1d & 2d cnn lstm networks. *Biomedical signal processing and control*, 47:312–323, 2019.

[46] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[47] Jingshan Huang, Binqiang Chen, Bin Yao, and Wangpeng He. Ecg arrhythmia classification using stft-based spectrogram and convolutional neural network. *IEEE access*, 7:92871–92880, 2019.

[48] fine-grained balanced cyberbullying dataset. https://ieee-dataport.org/open-access/fine-grained-balanced-cyberbullying-dataset, .

[49] toxic comment classification challenge. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge, .

[50] Sentencetransformer. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2, .

[51] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[52] John Muradeli. ssqueezepy. *GitHub. Note: https://github.com/OverLordGoldDragon/ssqueezepy/*, 2020. doi: 10.5281/zenodo.5080508.

[53] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[54] Jinyin Chen, Yi-tao Yang, Ke-ke Hu, Hai-bin Zheng, and Zhen Wang. Dad-mcnn: Ddos attack detection via multi-channel cnn. In *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*, pages 484–488, 2019.

[55] Nitin Kumar Chauhan and Krishna Singh. Impact of variation in number of channels in cnn classification model for cervical cancer detection. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, pages 1–6. IEEE, 2021.