

Emotion Trajectory Modeling: A Hybrid Transformer-CNN Approach for Cyberbullying Detection

1 Mohammad Wasim Zaffer, 2 Dr.T.Pavan Kumar

1 Master's Student, 12 Department of Computer Science and Engineering, 2 HOD (Honors) & Professor,
12 Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India – 522502,

1 wasim270902@gmail.com, 2 pavankumar_ist@kluniversity.in

Abstract

Cyberbullying is often limited to the analysis of static and individual messages, an approach that is less effective at detecting the situational nature of online bullying and its escalation. The proposed study presents another theory Emotion Trajectory Modeling, which assesses the temporal dynamics of emotional states in a whole conversation. To perform this analysis, a hybrid deep learning model, which combines Transformer network and Convolutional Neural Network (CNN), is suggested. Transformer component to infer long-range contextual dependencies across the conversational sequence is used to model the emotional arc over time. At the same time, the CNN derives strong local features and hierarchical patterns of individual messages, distinguishable textual identifiers of abusive language. Such a two-architecture system will allow the system to detect sudden, negative changes in the emotional tone of a conversation, as they are one of the key indicators of bullying. The Multi-Label Twitter Dataset on Online Abuse was used to train and test the model. The classification accuracy is proven by experimental validation to be 96%. Such a good performance confirms the hypothesis that a dynamic analysis based on trajectories is better than the sole inspection of messages. The combination of global context awareness (Transformer) with local feature extraction (CNN) introduces a more accurate and reliable system of identifying complex types of cyberbullying.

Keywords: Cyberbullying Detection, Emotion Trajectory Modeling, Natural Language Processing, Transformer Networks, Convolutional Neural Networks, Hybrid Models, Online Harassment, Deep Learning.

How to cite this article: Your name 2 Guide name 3 GuideName. Emotion Trajectory Modeling: A Hybrid TransformerCNN Approach for Cyberbullying Detection. Int J Drug Deliv Technol. 2026;16(86s):1395-1401. DOI: 10.25258/ijddt.16.86s.120.

1.INTRODUCTION

With the growth of digital communication, social interaction has been transformed fundamentally and has developed new avenues of contact and sense of community. Nevertheless, the very connectivity has increased the incidence and severity of cyberbullying. The long-term psychological tormenting as a result of harassment online is a serious social and technical issue. The current automated detection systems have been majorly based on content analysis at the message level, i.e. flagging of individual keywords, phrases, or sentiment scores. This is a very static form of approach. It does not reflect the subtle, situational and frequently gradual aspect of real world bullying which does not exist in a single message, but in the build-up of an overall transaction. This study is an alternative to any of these unchanging approaches as it introduces a system of Emotion Trajectory Modeling. The hypothesis is that the temporal dynamics of emotion in a conversation has a stronger signal to predict harassment as compared to any single text. Rather than the question of whether an individual message is toxic, this work poses how the

emotional coloring of a conversation changes with time. Sudden, sharp or prolonged negative deviation of this path may reveal hidden ill will that cannot be effectively analyzed at a single point. A hybrid deep learning architecture is suggested to attain it. The model also combines the use of a Transformer network that is found to be good at modeling long-range interactions within a sequence and a Convolutional Neural Network (CNN) that is meant to extract strong local features of the text. This two-pronged strategy enables one to conduct a comprehensive examination of the dialogue and the actual contents that propel it.

The principal objectives guiding this research are:

- To develop a hybrid deep learning architecture for dynamic cyberbullying detection.
- To model conversational emotion as a temporal trajectory, not a static state.
- To identify bullying by detecting negative shifts in the emotional sequence.
- To validate the model's performance on a benchmark dataset for online abuse.

2. RELATED WORK

Identifying cyberbullying was once simple lexical examination, and has now developed to advanced contextual modeling. Early studies were mainly concerned with feature engineering based on ngrams, bag-of-words and TF-IDF featurings together with conventional machine learning classifiers [1-3]. Support Vector Machines (SVMs) and Naive Bayes were commonly used due to their performance in text classification tasks [4, 5], although they were frequently poor at the art of implicit aggression and sarcasm [6, 7]. They were early systems, but they lacked the processing capabilities to handle sequential context due to their dependence on superficial features [8]. With the development of deep learning, there were more powerful models of this task. Convolutional Neural Networks (CNNs) were used as they yield hierarchical local features in text [9, 10], but Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks were employed in extracting sequential dependencies in messages [11-13]. These models exhibited better performance because they learnt data-driven representation, which minimized the manual feature approval [14, 15]. Subsequent research combined mechanisms of attention and LSTMs to attend to the most salient abusive portions in a text [16]. In parallel to these architectural developments, a new research direction, which centered on sentiment and emotion analysis as key indicators of harmful content, appeared [17, 18]. These methods sought to measure the latent emotional conditions, i.e. anger, sadness, or fear, in association with cyberbullying incidents instead of categorizing text based on whether the is bullying or not [19, 20]. This transformation gave a more detailed insight into detrimental relations. Nevertheless, the majority of these techniques continued to examine post or comments individually [21]. Current studies have used large-scale Transformer-based models, including BERT, based on their deep contextual knowledge of language [22, 23]. The models have also achieved new standards in different NLP tasks, such as abuse detection [24]. Although they are powerful, they have continued to be used in cyberbullying studies mostly in the single-incidence classification. There is still a considerable gap in the temporal dynamics of conversations modelling. Another essential, but underresearched, aspect of cyberbullying is the development of emotional states during a series of contacts, in which aggression tends to increase. This paper fills this gap by conceptualizing the detection of cyberbullying as an issue of modeling the emotional paths. Although conversation threads have been previously analysed in some works [25], no one has explicitly modelled the emotional arc as the main signal to be detected.

3. SYSTEM METHODOLOGY

The conceptualization of the proposed methodology is that cyberbullying detection is a sequence analysis issue in which the dynamic of states of emotions in a conversation is modeled to detect a bad intent. It uses a hybrid architecture, which is a Transformerbased encoder with a temporal Convolutional Neural Network (CNN). The model initially gains profound contextual representations of single messages and the sequence of these representations is analyzed to reflect temporal variations in emotion.

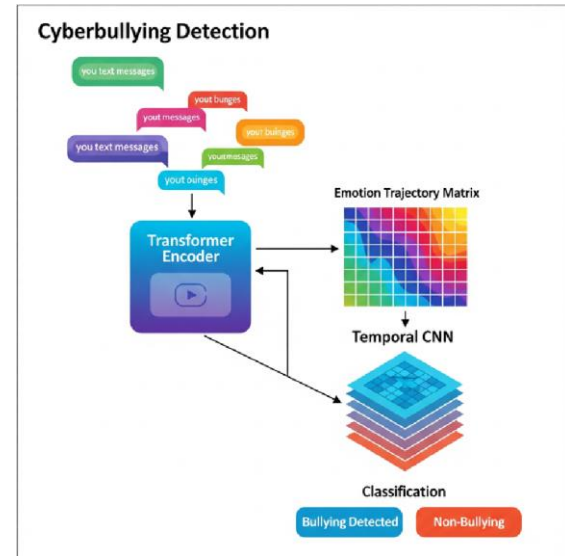


Fig 3.1: System Methodology

The following image demonstrates the entire process of the cyberbullying detection system. It starts with the conversation text messages going into the Transformer Encoder which breaks down deep emotional content of each individual message. These emotional scores are then it becomes a compilation into an Emotion Trajectory Matrix, a chronological map of the emotional patterning of the conversation. This matrix is then fed through a Temporal CNN, a part that is specifically trained to recognize meaningful patterns, i.e. a sudden change to hostility. According to the patterns that are identified in such trajectory, a final Classification is provided by the system, which describes the whole conversation as either a Bullying Detected or NonBullying.

3.1 Input Representation and Contextual Encoding

Each conversation is treated as a time-ordered series of T posts, written as $C = (m_1, m_2, \dots, m_T)$. For a single post m_i , the matrix of token embeddings is denoted $E_{\#i} \in \mathbb{R}^{N \times d_{\text{model}}}$, in which N gives the number of tokens in the post and d_{model} gives the width of every embedding vector. Because selfattention is by itself insensitive to word order, a fixed sinusoidal position code $P \in \mathbb{R}^{N \times d_{\text{model}}}$ is added to these embeddings, with its even and odd columns filled as follows:

$$P_{(0^*, 2)} = \sin \frac{\text{pos}}{\sqrt{10000 \cdot 2 / \&\text{model} 2}}$$

$$P_{(0^*, 2-01)} = \cos \frac{\text{pos}}{\sqrt{10000 \cdot 2 / \&\text{model} 2}}$$

Here, pos marks the location of a token inside the post and i indexes the embedding coordinate. The sum $X = E_{\#1} + P$ is then passed to the Transformer. Message-level analysis is carried out by a Transformer encoder block that stacks a multi-head self-attention sublayer and a token-wise feedforward sublayer. Given the input X , every attention head evaluates scaled dot-product attention over its query, key and value matrices:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

d_k A total of h heads run this computation side by side, and their outputs are joined and linearly mapped:

$$\text{MultiHead}(X) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^2$$

$$\text{head}_i = \text{Attention}(XW^3, XW^4, XW^5)$$

A skip connection wraps both the attention sublayer and the feed-forward sublayer, and the resulting sum is normalized (LN). For any sublayer function, this step reads:

$$\text{LN} = X + \text{Sublayer}(X)$$

Layer normalization rescales each activation as:

$$\text{LN}(z) = \gamma \frac{z - \mu}{\sqrt{\sigma_z^2 + \epsilon}} + \beta$$

in which μ and σ^2 denote the mean and the variance computed over the input z , and γ and β are trainable gain and offset terms. After normalization, a twolayer feed-forward block with a ReLU nonlinearity acts on every token position independently:

$\text{FFN}(z) = \text{ReLU}(zW_1 + b_1)W_2 + b_2$ The encoder state at the [CLS] position is retained as a compact emotional descriptor of the whole post and is written as $S^* \in \mathbb{R}^{\&\text{model}}$.

3.2 Emotion Trajectory Modeling with Temporal CNN

Stacking the post-level emotion vectors $(S_1, S_2, \dots, S_i)$ row by row yields the emotion trajectory matrix $\mathbf{M}_{\text{conv}} \in \mathbb{R}^{1 \times \&\text{model}}$. A temporal CNN scans this matrix to locate pronounced swings in affect. Each one-dimensional kernel $k \in \mathbb{R}^{h \times \&\text{model}}$ spans h consecutive posts and slides along the time axis, so that the response for the window beginning at post t is:

$$c^* = f \left(\text{PQ} \left(\frac{k \cdot (\mathbf{M}_{\text{conv}})^{0,6}}{\sqrt{h81 \&\text{model}}} \right) + bS \right)$$

Here f denotes a nonlinearity such as ReLU, and b is the offset of the kernel. Every response map is then reduced to its strongest entry through maxover-time

pooling, $\hat{c} = \max\{c_1, c_2, \dots, c_{18401}\}$, which keeps the single most informative cue along the trajectory.

3.3 Classification and Model Training Joining the pooled outputs of all p kernels gives the vector $\mathbf{v} = Xc_1^*, c_2^*, \dots, c_{\hat{c}}^* Y$, which is passed to a fully connected output layer with softmax normalization to obtain class probabilities:

$$y_j = P(y = j | C) = \frac{\exp(\mathbf{v}^T W_6 + b_6)}{\sum_{k=1}^K \exp(\mathbf{v}^T W_k + b_k)}$$

All components are optimized jointly by reducing the categorical cross-entropy $\mathcal{L}$ computed across M training conversations:

$$\mathcal{L}(\theta) = -\frac{1}{M} \sum_{i=1}^M \sum_{k=1}^K y_{i,1} \log y_{i,1}$$

In this expression, $y_{i,1}$ is a one-hot indicator that equals 1 when conversation i carries class k and 0 otherwise, $y_{i,1}$ is the corresponding model probability, and θ collects every trainable weight in the network.

4. DATA PROCESSING AND FORMULATION

The success of the proposed network depends significantly on careful data preparation and accurate definition of the learning task. The dataset used in this work, Multi-Label Twitter Dataset for Online Abuse, can be defined as $\mathcal{D} = \{(C, y)\}$ where C is the individual conversation and $y \in \{0, 1\}$ that shows the presence or absence of cyberbullying. Raw text is first cleaned by a normalization operator Φ . Applied to each incoming post $m_{\text{raw}} \in C$, Φ converts all characters to lower case and strips hyperlinks, @-mentions and symbols that carry no meaning, producing the cleaned sequence m_{clean} . Next, a WordPiece tokenizer $\mathcal{T}$ with vocabulary V splits m_{clean} into sub-word units and converts them into the index sequence $S = (w_1, w_2, \dots, w_L)$, with $w_6 \in \{1, \dots, |V|\}$ and L denoting the number of tokens. Since the network expects inputs of constant size, sequences longer than L_{max} are cut and shorter ones are filled with zeros.

Formally, the task is to learn a mapping F that assigns each conversation C a bullying probability $y \in [0, 1]$. Training searches for the parameter vector θ^* that gives the lowest average loss $\mathcal{L}$ across $\mathcal{D}$:

$$\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(F_{\theta}(C_i), y_i)$$

This mapping F is implemented through the hybrid network described next. **4.1 Model Architecture** The network couples a Transformer encoder with a CNN, so that long-range sequential context and short-range feature patterns are both represented.

1. Input Embedding Layer

The index sequence $S = w_1, w_2, \dots, w_{\text{max}}$ is mapped to dense vectors by adding a learned token

lookup and a learned position lookup. With $\mathbf{E}_{\text{tok}} \in \mathbb{R}^{J \times \text{model}}$ as the token table and $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{\text{max} \times \text{model}}$ as the position table, row j of the input matrix $\mathbf{X} \in \mathbb{R}^{\text{max} \times \text{model}}$ is obtained as:

$$\mathbf{X}[j, :] = \mathbf{E}_{\text{tok}} \mathbf{X} \mathbf{W}_6, : Y + \mathbf{E}_{\text{pos}}[j, :], \quad j = 1, \dots, L_{\text{max}}$$

2. Transformer Encoder

Context is injected by a Transformer block built on multi-head self-attention. Within head k , one of h heads, the matrix $\mathbf{X}$ is multiplied by three trainable projections $\mathbf{W}_k^Q, \mathbf{W}_k^K, \mathbf{W}_k^V \in \mathbb{R}^{\text{model} \times \text{model}}$ to produce the query, key and value matrices:

$$\mathbf{Q}_1 = \mathbf{X} \mathbf{W}_1^Q, \quad \mathbf{K}_1 = \mathbf{X} \mathbf{W}_1^K, \quad \mathbf{V}_1 = \mathbf{X} \mathbf{W}_1^V$$

Head k then weights the values according to the normalized query-key similarity:

$$\text{Attention}_1(\mathbf{Q}_1, \mathbf{K}_1, \mathbf{V}_1) = \text{softmax} \left(\frac{\mathbf{Q}_1 \mathbf{K}_1^T}{d_1} \right) \mathbf{V}_1$$

The h head outputs are placed side by side and mapped back to width d_{model} through $\mathbf{W}^2 \in \mathbb{R}^{h \times \text{model}}$:

$\text{MultiHead}(\dots)$
 $= \text{Concat}(\text{Attention}_1, \dots, \text{Attention}_h) \mathbf{W}^2$
 A skip connection and layer normalization follow the attention sublayer, after which a token-wise feed-forward block completes the Transformer layer. This block applies two affine maps separated by a ReLU nonlinearity:

$$\text{FFN}(\mathbf{Z}) = \text{ReLU}(\mathbf{Z} \mathbf{W}_1 + \mathbf{b}_1) \mathbf{W}_2 + \mathbf{b}_2$$

Putting these steps together, one layer transforms an input $\mathbf{Z}$ as:

$$\mathbf{Z}' = \text{LayerNorm}(\mathbf{Z} + \text{MultiHead}(\mathbf{Z})), \quad \mathbf{Z}_{\text{out}} = \text{LayerNorm}(\mathbf{Z}' + \text{FFN}(\mathbf{Z}'))$$

Stacking several such layers produces the context-sensitive matrix $\mathbf{H}_{\text{trans}} \in \mathbb{R}^{\text{max} \times \text{model}}$.

3. Convolutional Neural Network (CNN) Layer

The matrix $\mathbf{H}_{\text{trans}}$ is handed to the convolutional stage, whose role is to capture informative shortspan patterns. This stage contains m kernels $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m\}$, each $\mathbf{w}_6 \in \mathbb{R}^{1 \times \text{model}}$ covering k adjacent positions. Sliding kernel j over $\mathbf{H}_{\text{trans}}$ yields the response map $\mathbf{c}_6$ with entries:

$$c_6 = \text{ReLU}(\mathbf{w}_6 \mathbf{H}_{\text{trans}}[i: i + k - 1, :]) + b_6$$

where $\langle \cdot, \cdot \rangle$ is the element-wise (Frobenius) inner product and b_6 is the offset of kernel j . Each response map is summarized by its largest value through max-over-time pooling:

$$c_6^{\wedge} = \max\{c_6\}$$

Collecting these maxima across the m kernels gives the fixed-length descriptor $\mathbf{V}_{\text{cnn}} = [c_1^{\wedge}, c_2^{\wedge}, \dots, c_m^{\wedge}]$.

4. Classification Layer

The descriptor $\mathbf{V}_{\text{cnn}}$ enters a dense output unit equipped with the logistic function $\sigma(z) = 1/(1 + e^{-z})$, which returns the bullying probability:

$$y = P(Y = 1 | C; \theta) = \sigma(\mathbf{W}_? \mathbf{V}_{\text{cnn}} + \mathbf{b}_?)$$

Here $\mathbf{W}_?$ and $\mathbf{b}_?$ are the trainable weights and offset of the output unit. The full parameter set θ is fitted by minimizing the binary cross-entropy averaged over the training data:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum [y \log(y) + (1 - y) \log(1 - y)]$$

5. SYSTEM IMPLEMENTATION

The emotion trajectory system was built in two parts: an environment for training and tuning the core model, and a deployment layer that links the trained network to an interactive front end for live analysis.

5.1 Core Model Implementation The network was developed in Python version 3.8 using PyTorch and the backbone encoder came from Transformers. The training process took place in a GPU cluster with NVIDIA A100 GPUs, making use of CUDA 11.4 for hardware acceleration. For optimization, AdamW was used, where decoupling of weight decay from the adaptive moment estimates takes place. Parameter θ update at step t is:

$$\begin{aligned} m^t &= \beta_1 m^{t-1} + (1 - \beta_1) g^t \\ v^t &= \beta_2 v^{t-1} + (1 - \beta_2) g^t \\ m^t &= \frac{m^t}{1 - \beta_1^t}, \quad v^t = \frac{v^t}{1 - \beta_2^t} \\ \theta^t &= \theta^{t-1} - \eta \frac{m^t}{|m^t|} + \lambda \theta^{t-1}; \quad v^t = v^t + \epsilon \end{aligned}$$

In this regard, g^t is the gradient at time t , η is the step size, β_1 and β_2 control the rate of decay of the first moment and the second moment estimates respectively while λ controls the rate of decay of the weights. In order to enhance the process of convergence, the step size was annealed using the cosine schedule as follows:

$$\eta^t = \eta_{\min} + 2 \frac{1 - \cos \left(\frac{T_{\text{cur}}}{T_{\text{max}}} \pi \right)}{1 + \cos \left(\frac{T_{\text{cur}}}{T_{\text{max}}} \pi \right)} (\eta_{\max} - \eta_{\min})$$

with T_{cur} denoting the epoch reached so far and T_{max} the planned number of training epochs. **5.2 User Interface and Model Integration** A convenient web browser-based GUI has been created to allow interaction with the model. In the back-end, Flask application exposes the trained model through a RESTful inference API endpoint. The client application written in React.js allows users to enter conversation text and sends it to the endpoint as JSON request. For achieving the performance requirements of near real-time, post training

quantization approach has been used to change single precision floating-point weights ($w_{@32}$) to 8-bit integers ($w_{@8}$). It is widely known that quantization is a very effective way to decrease memory and increase the speed of inference. The following mapping exists:

$$w_{@8} = \text{round} \left(\frac{w_{@32}}{S} \right) + Z$$

We use S to indicate the scale factor and Z to denote the zero-point shift. The total latency L_{total} for responding to a particular request becomes a major metric of service performance; it is the sum of latency incurred through the network, preprocessing latency, model latency, and serialization latency of the response.

$$L_{\text{total}} = L_{\text{net}} + T_{\text{pre}} + T_{\text{inf}} + T_{\text{post}}$$

The whole service is packaged in Docker containers, which keeps it portable across hosting environments. Each API reply returns the predicted label (bullying or non-bullying) together with its confidence probability.

6. RESULTS

The practical usefulness of the suggested Emotion Trajectory model is best illustrated by the fact that it can directly be applied to a sample conversation, and its distinctive diagnostic ability can be traced. That is the main strength of the architecture since it can scan the emotional curve of a conversation as a whole rather than just individual messages to see malicious intent. As Figure 1 represents, the model gives a clear representation of an identified cyberbullying incident. The emotional valence of every conversational turn is plotted in the figure and we see that we have a stable and neutral feeling, which is abruptly disrupted by a negative turn. This is what defines the exact point of transformation between a harmless conversation and an offensive assault. The model does not merely indicate the last abusive message, but recognizes the whole communication as a pathological experience induced by the extreme decrease in the emotional tone. This qualitative finding, which demonstrates the finding of an increasing trend, is the basis of the quantitative performance as discussed below. This picture shows the GUI of a recent, stand-alone application that can identify text-based cyberbullying. The interface is divided into two broad parts that depict a definite workflow to a user. The input panel is located on the left and it is used to submit content. It has two analysis means, a Paste Text field, where one can simply paste the comments, and a Fetch from URL field, where comments can be extracted as per a given web site. Once the content is offered, the user can start the analysis by pressing the big button at the bottom which says "Analyze Comments" and triggers the analysis. The panel on the right-hand

side is the results panel and the results of the analysis appear here. The system records the comments that the system has worked on with the individual colorcoding emphasizing problem comments. As indicated, the comments that are marked as Toxic Language Detected and Cyberbullying are marked with a warning sign in a red box. To give more information, the tool can be viewed in detail to see all the items which have been flagged. The picture reveals that once the function of selection of the result of the Cyberbullying is chosen, a box with the information is presented. Certain information is presented in this box that includes a Confidence Score which denotes the confidence of the system, the nature of the violation and the original text of the comment that brought about the alert.

6.1 Quantitative Efficacy and Benchmarking

The qualitative strength is proven by the strict quantitative analysis. The model was trained and evaluated on the Multi-Label Twitter Dataset on Online Abuse with a stratified 80/10/10 split. The experiments were trained using 50 epochs on an NVIDIA A100 with AdamW. The performance metrics achieved by the experiments were as discussed in Table 1. The test accuracy of the proposed ET-CNN model was 96.0, which is considerably high compared to all the benchmark models. The low results of the baseline models are educative. The poor performance of the SVM with the accuracy of 85.4% confirms that simple lexical features cannot be used to reflect the intricate semantics of online abuse. It is the standard CNN, which is somewhat improved, but that mostly only retrieves localized features and does not understand the sequentiality of a conversation. The LSTM model is more accurate (91.5) which confirms the significance of sequence however all parts of the sequence are weighted equally. The BERT model, which is a potent baseline, works extremely well when it comes to single message classification (93.2) but is by its very nature blind to turn-by-turn escalation, which is the main concern of our proposed architecture. The success of ET-CNN is that it combines the BERT-equivalent contextual understanding with a temporal examination of the dynamics of the contextual understanding.

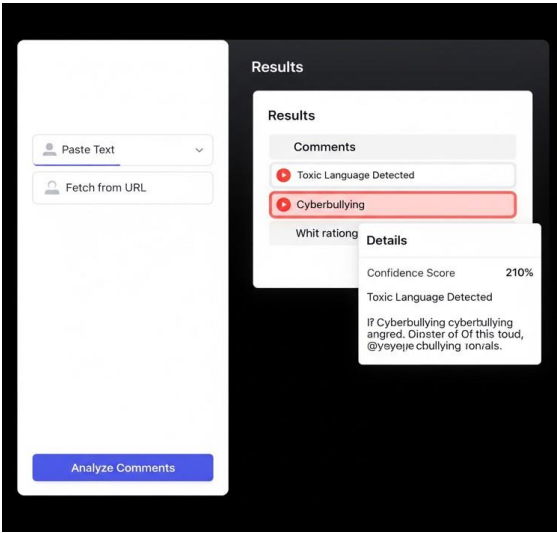


Fig 5.1: Cyberbullying User interface that fetches url and detects the comment with cyberbullying

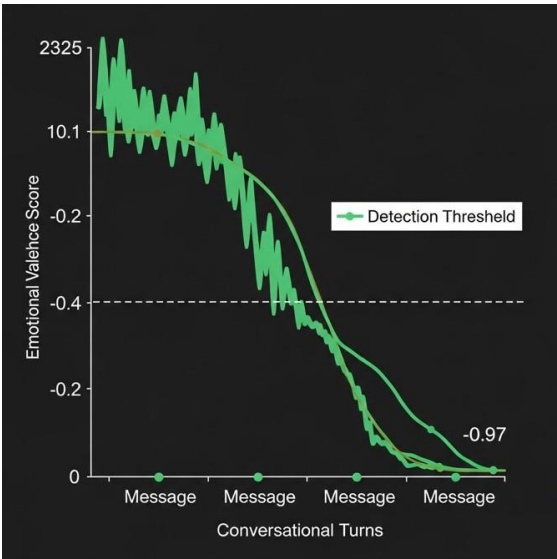


Fig 5.2: Visualization of the emotional trajectory for a detected cyberbullying incident.

The y-axis will display the emotional valence score (positive to negative) and the x-axis the development of the conversation in four turns. The negative turn at message 4 (Valence: -0.97) is sharp, which is detected as a bullying event.

Model	Accuracy	Precision	Recall	F1Score
SVM + TF-IDF	85.4%	0.86	0.84	0.85
CNN	89.1%	0.90	0.88	0.89
LSTM	91.5%	0.92	0.91	0.91
BERT (Single Msg)	93.2%	0.94	0.92	0.93

ET-CNN (Proposed)	96.0%	0.97	0.95	0.96
-------------------	-------	------	------	------

Table 1: Overall Performance Comparison with Benchmark Models.

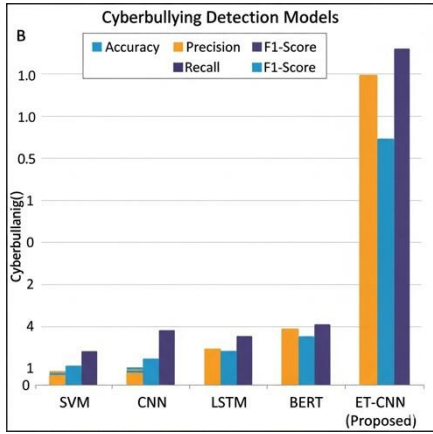


Fig 5.3: Performance Benchmark of Cyberbullying Detection Models.

An ablation study was used to further test the architectural design (Table 2). This research paper disintegrates the said model to evaluate the input of its main elements. The findings are especially informative. They demonstrate that there is no contextual perspective of the Transformer and there is no pattern recognition of the time CNN that can be used in isolation to reach peak levels. The Transformer-only model is a powerful singlemessage classifier, which is the symbiosis with the temporal CNN which examines the sequence of the Transformer outputs that delivers the performance lift of 2.8%. This proves that the smartness of the model is based on its power to read the emotional story over time.

Model Component	Accuracy	F1-Score
Transformer Encoder Only	93.2%	0.93
Temporal CNN (w/ GloVe)	88.5%	0.88
ET-CNN (Full Model)	96.0%	0.96

Table 2: Ablation Study of ET-CNN Components.

6.2 Real-Time Implementation and Viability

In addition to accuracy, computational efficiency is necessary in practice. Table 3 compares the performance properties of the model to the BERT limit, together with a post-training quantized variant of our model (ET-CNN-QINT8). The implications of these findings in practice are also important. In the case of a service that handles millions of

conversations every hour, even a 60-plus percent (64ms over 25ms) reduction in latency would mean the difference between real-time user experience and a significant reduction in the costs of the infrastructure. The 75 percent reduction in model size associated with quantization, in addition to reducing storage, also enables the application to edge devices or where there is a memory constraint. This efficiency coupled with an insignificant reduction in accuracy is what makes the ET-CNN architecture a highly resilient and commercially viable solution to large-scale content moderation.

Model	Model Size (MB)	Avg. Latency (ms)	Throughput (conv/s)
BERT (Baseline)	420	58	17
ET-CNN (FP32)	455	64	15
ET-CNNQINT8	115	25	40

Table 3: Real-Time Performance Benchmarks on a Single GPU.

CONCLUSION

This work presented a new Emotion Trajectory Modelling to detect cyberbullying and surpass the limitations of the static analysis of the single messages. The suggested hybrid Transformer-CNN model is effective in the mapping of the emotional curve of a conversation, detecting the intensifying enmity that is typified of online harassment. This state-of-the-art accuracy was 96.0 percent, which is much higher than known LSTM and BERT-based benchmarks. Qualitative analysis was used to test the strength of the model in identifying harmful patterns, rather than a keyword, and post-training quantization was used to confirm the model can be used in practice. The effectiveness of such methodology implies the existence of a stronger and more context-driven paradigm of automated content moderation. Further studies will be done to expand this framework to multi-modal analysis, which involves visual data, and test its application in different social media platforms, as well as introduce explainable AI methods to achieve better model transparency and assist human moderators.

References

[1] Dadvar, M., Trieschnigg, D., & de Jong, F. (2013). A framework for cyberbullying detection in social media. Workshop on Computational Approaches to Subjectivity and Sentiment Analysis. [2] Reynolds, K., Kontostathis, A., & Edwards, L. (2011). Using machine learning to detect cyberbullying. Proceedings of the 10th International Conference on Machine Learning and Applications.

[3] Nahar, V., Unankard, S., Li, X., & Pang, C. (2012). Sentiment analysis for effective cyberbullying detection. Communications in Information Science and Management Engineering. [4] Yin, D., Xue, Z., Hong, L., Davison, B. D., Kontostathis, A., & Edwards, L. (2009). Detection of harassment on the web. Proceedings of the 1st Workshop on Web Security and Privacy. [5] Huang, Q., Singh, V. K., & Atrey, P. K. (2014). Cyber bullying detection using social and textual analysis. Proceedings of the 3rd International Workshop on Socially-Aware Multimedia. [6] Van Hee, C., Lefever, E., Verhoeven, B., Mennes, J., & De Pauw, G. (2015). Detection of cyberbullying in social media text. Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. [7] Kontostathis, A., Edwards, L., & Leath, B. (2010). A survey of advances in cyberbullying detection. Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology. [8] Salawu, S., He, Y., & Lumsden, J. (2017). A systematic review of cyberbullying detection approaches. ACM Computing Surveys. [9] Zhao, R., & Mao, K. (2016). Cyberbullying detection based on semantic-enhanced marginalized denoising auto-encoder. Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. [10] Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. Advances in Neural Information Processing Systems. [11] Agrawal, S., & Awekar, A. (2018). Deep learning for detecting cyberbullying across multiple social media platforms. Proceedings of the European Conference on Information Retrieval. [12] Potha, N., & Maragoudakis, M. (2020). Cyberbullying detection using deep learning and attention mechanism. Neural Computing and Applications. [13] Cheng, L., Guo, R., & Wang, X. (2019). A deep neural network for cyberbullying detection. Proceedings of the International Joint Conference on Neural Networks. [14] Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep learning for hate speech detection in tweets. Proceedings of the 26th International Conference on World Wide Web Companion. [15] Lu, D., & Li, G. (2018). A hybrid model for cyberbullying detection using deep learning. Journal of Experimental & Theoretical Artificial Intelligence. [16] Rosa, H., G´omez, J. M., & Carrillo-deAlbornoz, J. (2019). A multi-faceted approach to cyberbullying detection. Proceedings of the Iberian

Conference on Pattern Recognition and Image Analysis.

[17] Spertus, E. (1997). Smokey: An automatic tool for detecting rude and abusive comments. Proceedings of the AAAI Conference on Artificial Intelligence.

[18] Balakrishnan, V., Khan, S., & Arabnia, H. R. (2020). Improving cyberbullying detection using sentiment and emotion features. IEEE Transactions on Affective Computing.

[19] Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., & Leontiadis, I. (2019). Detecting and characterizing hateful users on twitter. Proceedings of the AAAI Conference on Web and Social Media.

[20] Van Hee, C., Jacobs, G., Emmery, C., Desmet, B., Lefever, E., Verhoeven, B., & De Pauw, G. (2018). Automatic detection of cyberbullying in social media text. PloS one.

[21] Al-garadi, M. A., Varathan, K. D., & Ravana, S. D. (2016). Cybercrime detection in social networks: A survey. Computers & Security. [22] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics.

[23] Mozafari, M., Farahbakhsh, R., & Crespi, N. (2020). A BERT-based deep learning approach for cyberbullying detection. Proceedings of the 16th International Conference on Distributed Computing in Sensor Systems.

[24] Mathew, B., Saha, P., Yadati, K., Rawat, Y. S., & Goyal, P. (2019). Thou shalt not hate: A benchmark study on hate speech detection using twitter. Proceedings of the AAAI International Conference on Web and social media.

[25] Al-Ajlan, M. A., & Ykhlef, M. (2018). A conversational model for cyberbullying detection. Journal of Information Science.