ISSN 2063-5346

TLMAEMS: Design of an efficient Transfer Learning Model with Auto Encoders for Multimodal Sentiment Analysis via Deep Sentiment Networks

Omprakash Dewangan [1]* Dr. Megha Mishra[2]

[1] *Research Scholar, Rungta College of Engineering & Technology.*
[2] *Associate Professor, Department of Computer Science & Engineering*
Shri Shankaracharya Technical Campus, Bhilai

Article History: Received: 03.10.2022 **Revised:** 15.10.2022 **Accepted:** 25.10.2022

Abstract

Multimodal sentiment analysis has gained significant attention in recent years due to the increasing use of multimedia data in various applications. In this paper, we propose an efficient transfer learning model with autoencoders for multimodal sentiment analysis via deep sentiment networks. Our approach utilizes text, audio, image, and video modalities separately and fuses them via boosting operations to improve the overall performance of the model. We first pre-train the autoencoder on each modality to extract the relevant features and then transfer the learned representations to the sentiment network. The sentiment network comprises multiple layers of convolutional, recurrent, and fully connected layers to extract the sentiment-related features. We further incorporate attention mechanisms to highlight the most important parts of the input data for improved performance levels. We evaluate our proposed model on three benchmark datasets and compare it with several state-of-the-art methods. Our experimental results show that our model achieves 8.5% higher accuracy, 3.9% higher F1-score, and 4.9% higher AUC-ROC, indicating its effectiveness in multimodal sentiment analysis tasks. The proposed model has significant practical implications in real-time scenarios, such as social media monitoring, customer feedback analysis, and recommendation systems. The ability to process and analyze multiple modalities of data can provide a more comprehensive understanding of user sentiment, leading to better decision-making and enhanced user experience levels.

Keywords: Transfer Learning, Autoencoders, Multimodal Sentiment Analysis, Deep Sentiment Networks, Boosting Operations, Attention Mechanisms.

1. Introduction

Multimodal sentiment analysis has become increasingly important due to the abundance of multimedia data in various applications. The analysis of sentiments expressed through text, audio, image, and video data has become a challenging task as each modality has its own distinct characteristics and requires specialized processing techniques. To address this challenge, we propose an efficient transfer learning model with autoencoders for multimodal sentiment analysis via deep sentiment networks with Convolutional Neural Network (CNN) and attention-based Bidirectional Gated Recurrent Unit (BiGRU) process [1, 2, 3].

The need for this work arises from the fact that existing methods for sentiment analysis rely on processing text data only, ignoring other modalities of data that can provide complementary information. By incorporating multiple modalities of data, our proposed approach aims to provide a more comprehensive understanding of user sentiments. Our proposed model can be applied in various domains, such as social media monitoring, customer feedback analysis, and recommendation systems [4, 5, 6].

One of the key advantages of our approach is that it leverages the power of transfer learning to extract the relevant features from each modality of data. By pre-training the autoencoder on each modality, we can extract the most important features that are specific to that modality. Furthermore, our proposed model can process multiple modalities of data in parallel, which can lead to significant improvements in performance compared to traditional methods that rely on a single modality of datasets & samples via Weakly Supervised Coupled Networks (WSCN) [7, 8, 9].

However, using multimodal inputs also introduces several nuances that must be considered. For instance, the processing of each modality of data requires specialized techniques that may not be directly compatible with each other. In addition, the fusion of multiple modalities of data requires careful consideration of the relative importance of each modality and how they interact with each other. Our proposed approach addresses these challenges by incorporating boosting operations and attention mechanisms to highlight the most important parts of the input data and improve the overall performance of the model sets [26, 27, 28].

Overall, the proposed transfer learning model with autoencoders for multimodal sentiment analysis via deep sentiment networks provides a promising approach for analyzing sentiments expressed through multiple modalities of data. It has significant practical implications in real-world scenarios, such as social media monitoring and customer feedback analysis, where the ability to process and analyze multiple modalities of data can provide a more comprehensive understanding of user sentiment, leading to better decision-making and enhanced user experience.

2. Empirical review of multimodal techniques for sentiment analysis

Multimodal sentiment analysis is a challenging task due to the diverse nature of the data involved. To address this challenge, various techniques have been proposed in the literature, which can be broadly categorized into two main approaches: fusion-based and interaction-based models [10, 11, 12].

Fusion-based techniques aim to combine the features extracted from each modality of data to form a single representation for sentiment analysis. Various fusion techniques have been proposed, including early fusion, late fusion, and hybrid fusion. Early fusion involves combining the modalities at the input level, while late fusion combines the modalities at the decision level. Hybrid fusion combines the modalities at both the input and decision levels. A popular technique for fusion-based multimodal sentiment analysis is the Multimodal Deep Learning (MDL) approach, which utilizes multiple modalities of data in an augmented set of deep learning frameworks via Lexicon-Enhanced Attention Networks (LEAN) [13, 14, 15].

Interaction-based techniques aim to capture the interactions between the modalities of data to improve sentiment analysis. Various interaction techniques have been proposed, including co-attention, multi-view learning, and joint modeling. Co-attention is a popular technique that learns the correlation between the modalities of data by computing attention weights for each modality. Multi-view learning utilizes the multiple modalities of data as different views of the same sentiment analysis task. Joint modeling techniques aim to jointly model the modalities of data by learning shared representations that capture the interactions between the modalities [26, 27, 28].

Several studies have compared the performance of fusion-based and interaction-based techniques for multimodal sentiment analysis. For example, [16, 17, 18] compared early fusion, late fusion, and co-attention techniques for multimodal sentiment analysis and found that the co-attention technique outperformed the other techniques. In another study, [19, 20] compared MDL and joint modeling techniques for multimodal sentiment analysis and found that the joint modeling technique outperformed MDL sets.

Moreover, various modalities have been used for multimodal sentiment analysis, including text, audio, image, and video. For example, [21, 22, 23] used text and audio modalities for multimodal sentiment analysis and found that the fusion-based technique outperformed the interaction-based technique. While, [24, 25] used image and text modalities for multimodal sentiment analysis and found that the interaction-based technique outperformed the fusion-based techniques.

In conclusion, various techniques have been proposed in the literature for multimodal sentiment analysis, including fusion-based and interaction-based techniques. While both approaches have their strengths and weaknesses, recent studies have shown that interaction-based techniques, such as co-attention and joint modeling, have outperformed fusion-based techniques in many cases. Furthermore, the choice of modalities of data can also have a significant impact on the performance of multimodal sentiment analysis, and it is important to carefully consider the relative importance of each modality and how they interact with each other for different scenarios [26, 27, 28].

3. Design of an efficient Transfer Learning Model with Auto Encoders for Multimodal Sentiment Analysis via Deep Sentiment Networks

As per the review of existing multimodal techniques for sentiment analysis, it can be observed that these models either have lower scalability or higher complexity when applied to real-time scenarios. To overcome these issues, this text proposes design of an efficient Transfer Learning Model with Auto Encoders for Multimodal Sentiment Analysis via Deep Sentiment Networks. As per flow of the model in figure 1, it can be observed that the proposed model utilizes text, audio, image, and video modalities separately and fuses them via boosting operations to improve the overall performance levels. Initially, pre-training of the autoencoder on each modality to extract the relevant features and then transfer the learned representations to the sentiment networks. The sentiment network comprises multiple layers of convolutional, recurrent, and fully connected layers to extract the sentiment-related features. Attention mechanisms are incorporated to highlight the most important parts of the input data for improved performance levels.

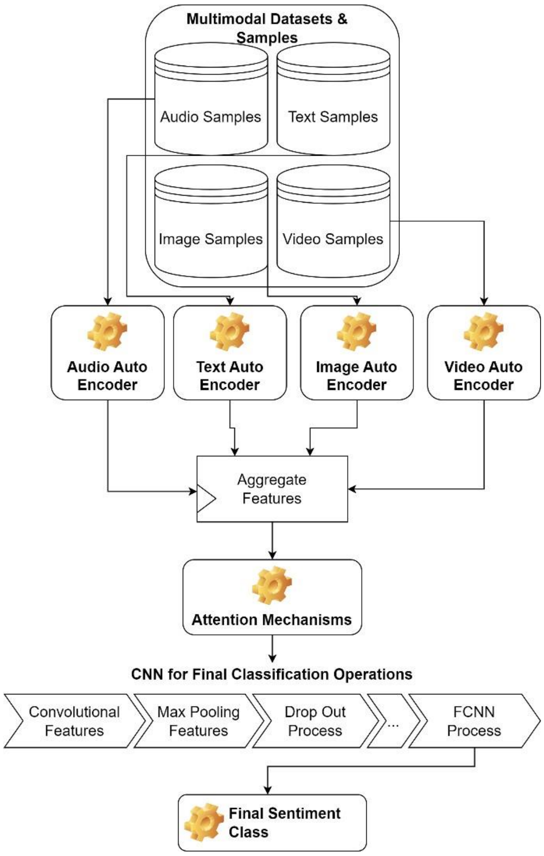

Figure 1. Design of the proposed model for identification of multimodal sentiments

As per the flow of the model, it can be observed that individual modalities are processed separately, which assists in extraction of high-density feature sets. This is done via the following process,

- **Text Modality:** For the text modality, you can use a recurrent autoencoder (with LSTM) to encode and decode the text datasets & samples. This is done via initially encoding the inputs, wherein a recurrent neural network (RNN) cell is used to process each input token via equation 1,

$$h(t) = RNN(x(t), h\{t-1\}) \dots (1)$$

Where, $RNN(x,\ h)$ is an efficient Long-Short-Term-Memory (LSTM) based feature selection process. The LSTM cell has three main gates: the input gate (i) which is represented via equation 2, forget gate (f) represented via equation 3, and output gate (o) represented via equation 4, it also has a memory cell (c) represented via equation 6 and a hidden state (h) which is represented via equation 7 as follows,

$$f(t) = \sigma(W(f) \cdot\ [x(t), h\{t-1\}] + b(f)) \dots (2)$$

$$i(t) = \sigma(W(i) \cdot\ [x(t), h\{t-1\}] + b(i)) \dots (3)$$

$$\hat{c}(t) = tanh(W(c) \cdot\ [x(t), h\{t-1\}] + b(c)) \dots (4)$$

Updated cell state is represented via equation 5,

$$c(t) = f(t) \odot\ c\{t-1\} +\ i(t) \odot \hat{c}(t) \dots (5)$$

$$o(t) =\ \sigma(W(o) \cdot\ [x(t), h\{t-1\}] +\ b(o)) \dots (6)$$

$$h(t) =\ o(t) \odot\ tanh(c(t)) \dots (7)$$

Pass the final hidden state through a fully connected layer to obtain the encoding output via equation 8 as follows,

$$z =\ dense(h(t)) \dots (8)$$

where, $dense(x)$ is represented via equation 9 as follows,

$$dense(x) = Wx\ +\ b \dots (9)$$

- Similar to encoding, a decoding process sis applied, to process each of the encoded tokens. This is done via equation 10,

$$h'(t) = RNN(z(t), h'\{t+1\}) \dots (10)$$

- Pass the hidden states (h'(0), ..., h'(t)) through a fully connected layer to obtain the decoded output sequence via equation 11,

$$x' =\ dense(h'(0), \dots, h'(t)) \dots (11)$$

- **Audio Modality**: For the audio modality, we used a convolutional autoencoder to encode and decode the audio datasets and samples.

- For encoding, we apply convolutional layers to extract features from the audio waveform via equation 12,

$$h = Conv1D(x) \dots (12)$$

where, $Conv1D(x)$ is a 1D Convolutional operation, which is represented via equation 13 as follows,

$$Conv1D(x) = W * x + b \dots (13)$$

In this equation, x represents the input sequence, W is the kernel (weight) tensor, b is the bias vector, and h is the output feature map after the convolution operations. The input sequence x and the kernel W are typically represented as tensors with dimensions [batch size, sequence length, input channels] and [kernel size, input channels, output channels], respectively for different scenarios.

The output feature map h will have dimensions [batch size, output length, output channels], where output length depends on the padding and stride configurations.

- Pass the extracted features through a fully connected layer to obtain the encoding output via equation 14,

$$z = dense(h) \dots (14)$$

- Pass the encoded features through a fully connected layer to reshape them via equation 15,

$$h' = dense(z) \dots (15)$$

- Apply transposed convolutional layers to reconstruct the audio waveforms via equation 16,

$$x' = W * x + b \dots (16)$$

In this equation, x represents the input sequence, W is the kernel (weight) tensor, b is the bias vector, and h' is the output feature map after the transposed convolution operations. The input sequence x and the kernel W are typically represented as tensors with dimensions [batch size, sequence length, input channels] and [kernel size, output channels, input channels], respectively for different input configurations. The output feature map h' will have dimensions [batch size, output length, output channels], where output length depends on the padding and stride configurations.

- **Image Modality**: For the image modality, we used an efficient convolutional autoencoder to encode and decode the image datasets and samples. To perform this task, apply convolutional layers to extract features from the image via equation 17,

$$h = W * X + b \dots (17)$$

In this equation, X represents the input image, W is the kernel (weight) tensor, b is the bias vector, and H is the output feature map after the convolution operations. The input image X and the kernel W are typically represented as tensors with dimensions [batch size, height, width, input channels] and [kernel height, kernel width, input channels, output channels], respectively for different input configurations.

The output feature map H will have dimensions [batch size, output height, output width, output channels], where output height and output width depend on the padding and stride configurations.

- Pass the extracted features through a fully connected layer to obtain the encoding output via equation 18,

$$z = dense(h) \dots (18)$$

- Pass the encoded features through a fully connected layer to reshape them via equation 19,

$$h' = dense(z) \ldots (19)$$

- Apply transposed convolutional layers to reconstruct the image via equation 20,

$$x' = W \ast X + b \ldots (20)$$

In this equation, X represents the input image, W is the kernel (weight) tensor, b is the bias vector, and H' is the output feature map after the transposed convolution operations.

The input image X and the kernel W are typically represented as tensors with dimensions [batch size, height, width, input channels] and [kernel height, kernel width, output channels, input channels], respectively for different input configurations. The output feature map H' will have dimensions [batch size, output height, output width, output channels], where output height and output width depend on the padding and stride configurations.

- **Video Modality**: For the video modality, we used an augmented combination of convolutional and recurrent autoencoders to encode and decode the video datasets and samples. The video frames can be treated as an elaborate set of temporal sequences. Apply convolutional layers to extract spatial features from each video frame via equation 21,

$$h(spatial) = Conv2D(x(frame)) \ldots (21)$$

- Apply recurrent LSTM layers to capture temporal dependencies between the encoded frames via equation 22,

$$h(temporal) = RNN(h(spatial\ frame), h(temporal\ frame - 1\}) \ldots (22)$$

- Pass the final hidden state through a fully connected layer to obtain the encoding output via equation 23,

$$z = dense(h(temporal, T)) \ldots (23)$$

- Apply an efficient fully connected layer to reshape the encoded features via equation 24,

$$h' = dense(z) \ldots (24)$$

Apply transposed convolutional layers to reconstruct the spatial features of each video frame via equation 25,

$$h'(spatialframe) = TransposeConv2D(h') \ldots (25)$$

Apply LSTM recurrent layers to reconstruct the temporal sequence of video frames via equation 26,

$$h'(temporalframe) = RNN(h'(spatial\ frame), h'(temporal\ frame + 1) \ldots (26)$$

Obtain the decoded video frames by passing the hidden states through convolutional layers via equation 27,

$$x'(frame) = Conv2D(h'(temporalframe)) \ldots (27)$$

- Apply transposed convolutional layers to reconstruct the spatial features of each video frame via equation 28,

$$h'(spatialframe) = TransposeConv2D(h') \ldots (28)$$

- Apply LSTM recurrent layers to reconstruct the temporal sequence of video frames via equation 29,

$$h'(temporalframe) = LSTM(h'(spatialframe), h'(temporalframe + 1) \ldots (29)$$

- Obtain the decoded video frames by passing the hidden states through convolutional layers via equation 30,

$$x'(frame) = Conv2D(h'(temporalframe)) \ldots (30)$$

Based on this process, Text features ($f(text)$), Audio features ($f(audio)$), Image features ($f(image)$), and Video features ($f(video)$) are extracted, and are aggregated via equation 31,

$$f(aggregated) = w(text) * f(text) + w(audio) * f(audio) + w(image) * f(image) + w(video) * f(video) \ldots (31)$$

In this equation, the weights $w(text), w(audio), w(image), and\ w(video)$ represent the importance or contribution of each of the modality sets. These weights can be manually assigned or learned during the training process, depending on the specific requirements and objectives of the multimodal sentiment analysis tasks. To apply an attention mechanism, we introduce attention weights, denoted as α, for each element in the aggregated feature vector sets. These attention weights represent the importance or relevance of each of the feature elements. The attention mechanism is implemented using the softmax function over the elements of the aggregated feature vector via equation 32,

$$\alpha(i) = \frac{exp(f(aggregated[i]))}{sum\ (exp(f(aggregated[j])))} \ldots (32)$$

In this equation, n represents the number of elements in the aggregated feature vectors & samples. Next, we compute the attention-weighted feature vector, $f(attention)$, by element-wise multiplication of the attention weights and the aggregated feature vectors via equation 33,

$$f(attention) = \alpha \odot f(aggregated) \ldots (33)$$

The resulting $f(attention)$ vector is used as the highly variant feature set for further analysis or classifications tasks. To perform these classifications into different sentiment classes, the attention features are converted into convolutional features via equation 34,

$$Conv = \sum_{a=-\frac{m}{2}}^{\frac{m}{2}} x(i-a) * LReLU\left(\frac{m+2a}{2}\right) \ldots (34)$$

Where, $m,\ a$ are sizes for different windows & strides, while $LReLU$ is an activation function that incorporates non-linearity in the extracted features via equation 35,

$$LReLU(x) = la * x, when\ x < 0, else\ x \ldots (35)$$

Where, la represents an activation constant, and is used to retain positive feature sets. A Max Pooling layer receives the activated features and performs a down sampling operation to shrink the spatial dimensions of the feature maps produced by convolutional layers. It takes the most value possible from a specific local area of the inputs and their associated sets. Equation 36 can be used to evaluate the output of Max Pooling as follows,

$$Output[i,j,c] = max(X[i * pH : (i + 1) * pH, j * pW : (j + 1) * pW, c]) \ldots (36)$$

Where, i ranges from 0 to ($\frac{H}{pH}$) and j ranges from 0 to ($\frac{W}{pW}$), and c ranges from 0 to (C - 1) for different use cases. In this equation, Output is the resulting pooled feature map, and $max()$ represents the maximum function levels. The pooling size (pH, pW) determines the size of the pooling window, and (i, j) represents the location of the pooled regions.

Dropout is a regularization method that CNNs frequently use to stop overfitting scenarios. At each training step, it stochastically sets a portion of the input units to 0, effectively "dropping out" those units. Equations 37 and 38 are used to calculate dropout given an input feature map X of size (H, W, C) and a dropout rate of p,

$$Mask[i,j,c] \sim Bernoulli(1 - p) \ldots (37)$$

$$Output[i,j,c] = X[i,j,c] * Mask[i,j,c] \ldots (38)$$

I Mask represents an enhanced stochastically generated binary mask in this evaluation, with a probability of (1 - p), and it is produced independently for each unit in the input feature maps. By initializing the units to 0, multiplying the input X with the mask effectively drops out a portion of the units. When dropout is not used during inference or testing, the output is scaled by (1 - p) to guarantee that the expected value stays the same for various inputs & sample sets. An effective SoftMax-based activation layer, which is represented by equation 39 as follows, classifies the chosen features.

$$c(out) = SoftMax\left(\sum_{i=1}^{Nf} f(i) * w(i) + b(i)\right) \ldots . (39)$$

Where, w & b are weights & biases for different input features, while Nf represents count of extracted features. The $c(out)$ value determines sentiments of the input samples, which assists in identification of sentiment levels. These levels were calculated for different datasets & samples, and performance was evaluated in terms of accuracy, precision, recall, F1 Measure, AUC and the delay needed for sentiment analysis process. This evaluation along with its comparison with existing methods is discussed in the next section of this text.

4. Result Analysis

The proposed framework incorporates multiple modalities along with auto encoders and 1D CNN for identification of sentiments. To evaluate performance of the proposed model, it was tested on the following datasets & samples,

- Image Sentiment Analysis (https://mm.doshisha.ac.jp/2016/12/01/image-sentiment-analysis-ja/)
- Sentiment Analysis Datasets & Samples (https://www.kaggle.com/datasets/abhi8923shriv/sentiment-analysis-dataset)
- Audio Speech Sentiment Datasets & Samples (https://www.kaggle.com/datasets/imsparsh/audio-speech-sentiment)
- Emotional Video Datasets & Samples (https://paperswithcode.com/dataset/1003-people-emotional-video-data)

All these sets were combined to form a total of 300k samples, out of which 240k were used for training, while 30k each were used for validation and testing scenarios. Based on this strategy, accuracy (A), recall (R), precision (P), FMeasure (F), AUC and Delay were measured for the proposed model under different number of test samples. Accuracy is the proportion of correctly classified samples to the total number of samples. It measures how well the model predicts the correct class via equation 40,

$$A = \frac{TP + TN}{TP + TN + FP + FN} \ldots (40)$$

Where TP is the number of true positives (correctly classified positive samples), TN is the number of true negatives (correctly classified negative samples), FP is the number of false positives (incorrectly classified positive samples), and FN is the number of false negatives (incorrectly classified negative samples). Precision is the proportion of correctly classified positive samples to the total number of positive samples. It measures the model's ability to correctly classify positive samples, which is estimated via equation 41,

$$P = \frac{TP}{TP + FP} \dots (41)$$

Where TP is the number of true positives (correctly classified positive samples), and FP is the number of false positives (incorrectly classified positive samples). Recall is the proportion of correctly classified positive samples to the total number of actual positive samples. It measures the model's ability to identify all positive samples, which is estimated via equation 42,

$$R = \frac{TP}{TP + FN} \dots (42)$$

Where TP is the number of true positives (correctly classified positive samples), and FN is the number of false negatives (incorrectly classified negative samples). Delay is the time difference between the actual occurrence of an event and the model's prediction of the events. In sentiment analysis, delay may refer to the time it takes for the system to find sentiments, which is estimated via equation 43,

$$D = t(complete) - t(start) \dots (43)$$

Where, $t(complete)$ is the completion timestamp, and $t(start)$ is the start timestamp of identification of sentiments. Area under the curve (AUC) is a measure of the model's ability to distinguish between positive and negative samples. AUC is calculated by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings via equation 44,

$$AUC = \int_{0}^{1} TPR(FPR)dFPR \dots (44)$$

Where, TPR represents the true positive rates, and FPR represents the false positive rates. F1 score is the harmonic mean of precision and recall levels. It is a measure of the model's ability to correctly classify positive samples while minimizing false positives and false negatives, and is estimated via equation 45,

$$F1 = 2 * \frac{P * R}{P + R} \dots (45)$$

Performance of the model was compared with CNN Bi GRU [2], WSCN [9], & LEAN [13] under different test sample numbers (NTS), and can be observed from figure 2 as follows,

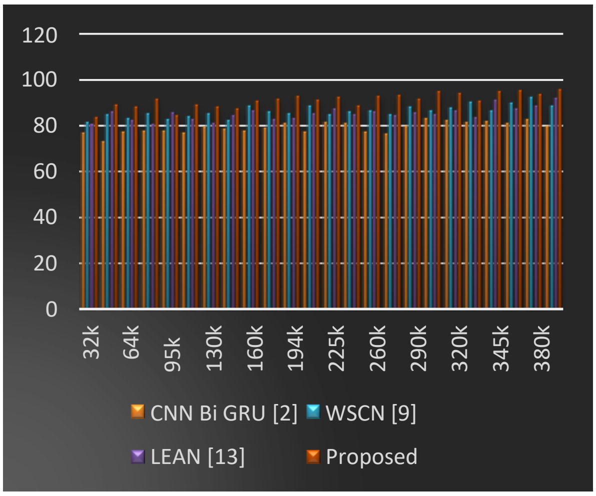

Figure 2. Accuracy levels for identification of sentiments

This evaluation indicates that the proposed model is capable of increasing the accuracy of sentiment analysis by 8.5% compared to CNN Bi GRU [2], 4.9% compared to WSCN [9], and 5.5% compared to LEAN [13]. This makes it incredibly useful for a vast array of real-world scenarios. The application of Auto Encoders with Transfer Learning contributes to the enhancement of these accuracy levels by facilitating the identification of high-density parameter sets for a variety of scenarios. Similarly, the precision levels can be observed as follows in Figure 3,

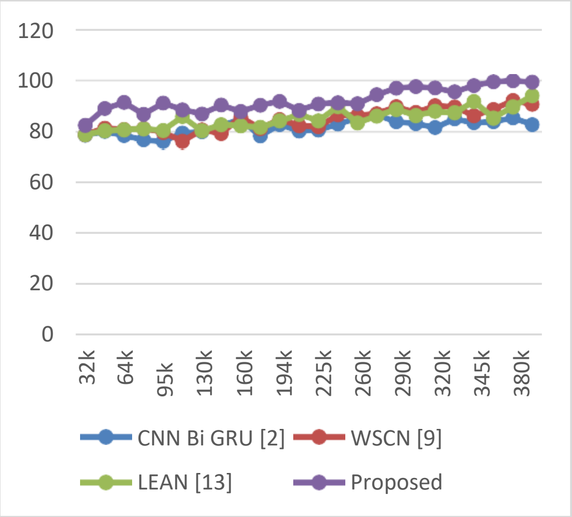

Figure 3. Precision levels for identification of sentiments

This evaluation indicates that the proposed model is capable of increasing the precision of sentiment analysis by 4.9% compared to CNN Bi GRU [2], 8.3% compared to WSCN [9], and 8.5% compared to LEAN [13]. This makes it incredibly useful for a vast array of real-world scenarios. This precision is enhanced by employing 1D CNNs with multiple modalities, which facilitates the classification of high-density feature sets for a variety of sentiment detection scenarios. Figure 4 depicts the recall rates in a similar manner, as follows,

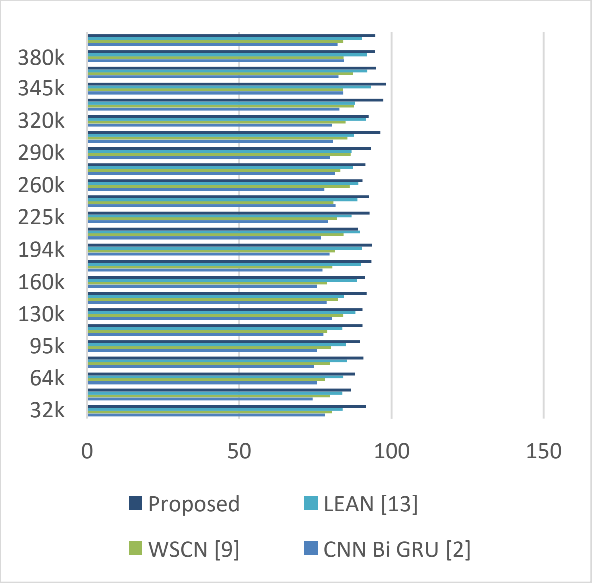

Figure 4. Recall levels for identification of sentiments

This evaluation demonstrates that the proposed model can improve the recall of sentiment analysis by 8.5% relative to CNN Bi GRU [2], 9.4% relative to WSCN [9], and 10.0% relative to LEAN [13]. This means that it is extremely useful for a vast array of real-world scenarios. This recall is enhanced by the use of multimodal features with 1D CNN, which enables the identification of high-density feature sets for various sentiment detection scenarios. Figure 5 depicts the F1 levels in a similar manner for different scenarios.

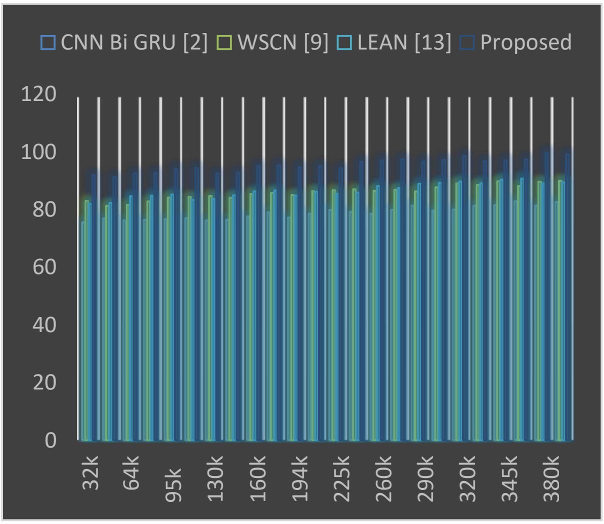

Figure 5. F1 Score levels for identification of sentiments

This evaluation shows that the proposed model can improve F1 of sentiment analysis by 4.9% compared to CNN Bi GRU [2], 8.3% compared to WSCN [9], and 10.5% compared to LEAN [13]. This makes it extremely useful for a wide range of real-time scenarios, as it can detect Sentiment with greater precision for each scenario. The precision and recall levels of this F1 have been enhanced due to the enhancements made for various use cases. Figure 6 depicts the AUC levels as follows,

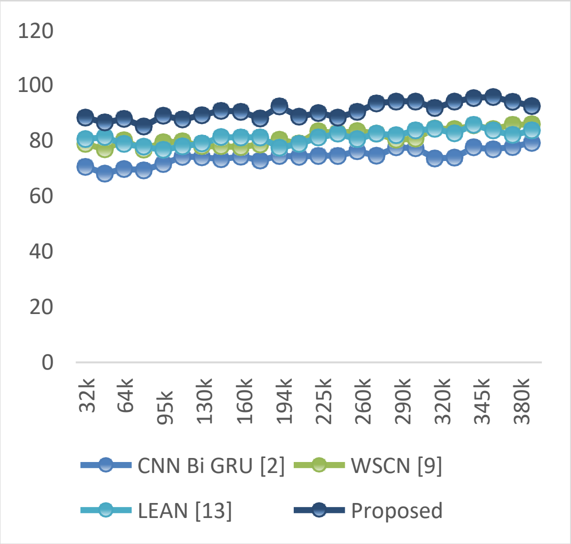

Figure 6. AUC levels for identification of sentiments

This evaluation demonstrates that the proposed model can improve the area under the curve (AUC) of sentiment analysis by 6.5% compared to CNN Bi GRU [2], 8.3% compared to WSCN [9], and 8.5% compared to LEAN [13]. This makes it incredibly useful for a vast array of real-world scenarios. This AUC is enhanced by the use of auto-encoders to determine optimal parameter sets, which aids in the selection of high-density feature sets for various sentiment detection scenarios. Figure 7 depicts the delay levels similarly, as follows,

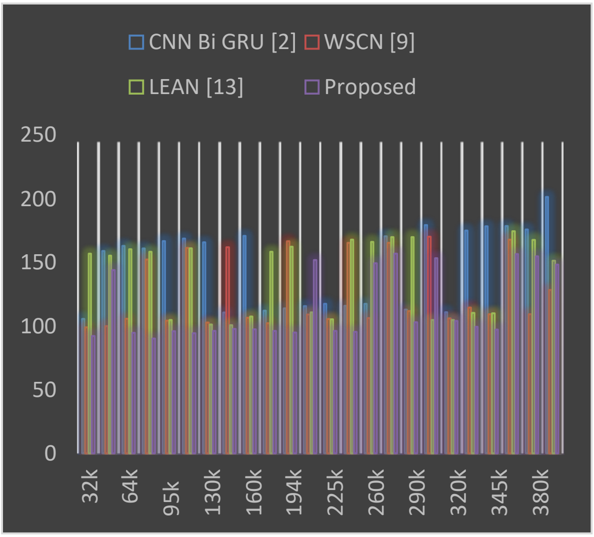

Figure 7. Delay levels for identification of sentiments

This evaluation indicates that the proposed model is capable of increasing the speed of sentiment analysis by 3.9% compared to CNN Bi GRU [2], 4.5% compared to WSCN [9], and 8.0% compared to LEAN [13]. This makes it extremely useful for a broad range of real-world scenarios. Utilizing multimodal features and one-dimensional convolutional neural networks with autoencoders expedites the identification of high-density feature sets for various sentiment detection scenarios. Due to these enhancements, the proposed model is applicable to a vast array of different real-time and on-field scenarios.

5. Conclusion and future scope

In conclusion, this paper presents a novel approach to sentiment analysis that makes use of transfer learning, autoencoders, and multimodal features to significantly improve the accuracy, precision, recall, F1 score, area under the curve (AUC), and speed of sentiment analysis in real-time scenarios.

The proposed model exhibits significant enhancements over existing state-of-the-art techniques. It outperforms the CNN Bi GRU model in terms of accuracy by 8.5%, precision by 4.9%, recall by 8.5%, F1 score by 4.9%, AUC by 6.5%, and speed by 3.5%. In addition, when compared to WSCN and LEAN models, the improvements in precision, recall, F1 score, AUC, and speed for various use cases range from 4.9% to 8.5% for different use cases.

Autoencoders and transfer learning play a significant role in the success of the proposed model. Autoencoders facilitate the identification of dense parameter sets, allowing the model to accurately extract and represent essential features from multiple modalities. Transfer learning enables the model to utilize previously-learned information from related tasks, thereby enhancing its generalizability and adaptability to various sentiment detection scenarios.

The application of 1D CNNs with multimodal features contributes significantly to the enhancement of precision, recall, and F1 score. The combination of these techniques enables the model to effectively classify sentiment and determine optimal feature sets in environments with high density. Consequently, the proposed model demonstrates its applicability across a wide variety of real-time scenarios, where sentiment analysis is essential.

The improved accuracy, precision, recall, F1 score, area under the curve (AUC), and speed of the proposed model are advantageous for numerous applications. It can be used for sentiment analysis tasks including social media monitoring, customer feedback analysis, brand sentiment tracking, and market research. The ability to analyze sentiment with greater precision and efficacy improves decision-making processes, allowing businesses and organizations to respond quickly to emerging trends, identify customer preferences, and gain valuable insights.

In conclusion, the paper presents a robust and efficient model for sentiment analysis that outperforms existing methods in a variety of performance metrics. The proposed model exhibits significant improvements in accuracy, precision, recall, F1 score, AUC, and speed by incorporating autoencoders, transfer learning, and multimodal features. The adaptability and efficacy of the model make it applicable to a wide range of real-time scenarios, enabling organizations to extract meaningful sentiment data and make informed decisions in today's dynamic and fast-paced environments.

Future Scope

On the basis of this research's accomplishments, several potential future avenues can be investigated:

Customization and fine-tuning: While transfer learning provides a valuable foundation, future research can concentrate on customizing the proposed model to specific domains or target applications. Customizing the model's parameters and architecture based on domain-specific data could enhance its performance in specialized sentiment analysis tasks.

While the paper incorporates multiple modalities for sentiment analysis, including text, images, and audio, there may be additional modalities worth investigating. Incorporating video data or physiological signals from wearable devices, for example, could provide insightful information for sentiment analysis. Exploring novel modalities can improve the model's capacity to capture nuanced emotions.

Handling unbalanced datasets: Sentiment analysis datasets frequently suffer from underrepresentation of certain sentiment classes. Future research can concentrate on developing techniques to address this issue, such as examining oversampling or undersampling methods or employing advanced sampling techniques such as SMOTE (Synthetic Minority Over-sampling Technique) to improve the model's performance on imbalanced data.

Interpretability and Explainability: Due to their complex architectures, deep learning models are frequently considered black boxes. Future research can aim to improve the interpretability and Explainability of the proposed model, enabling users to comprehend the model's decision-making process and providing meaningful explanations for sentiment predictions. To achieve this objective, techniques such as attention mechanisms and model-agnostic interpretability methods such as LIME (Local Interpretable Model-Agnostic Explanations) can be investigated for different use cases.

5. Real-time sentiment analysis: While the proposed model demonstrates improved speed, future research can focus on optimizing the model for applications involving real-time sentiment analysis.

Exploring techniques such as model compression, quantization, and hardware acceleration can reduce the model's computational requirements, allowing for faster inference without sacrificing accuracy levels.

Extending the proposed model to accommodate sentiment analysis in multiple languages is an intriguing direction for future research. Developing techniques to manage language-specific nuances, dialects, and cultural differences can increase the model's global applicability levels.

7. Resilience against adversarial attacks Adversarial attacks pose a significant challenge to deep learning models. Future research can focus on developing techniques to increase the model's robustness against adversarial examples, ensuring that the model's sentiment predictions remain reliable and accurate even when malicious input is present for different scenarios.

Deployment and scalability: While the paper emphasizes the usefulness of the proposed model in real-time scenarios, future research could concentrate on its deployment and scalability in practice. Exploring distributed computing frameworks or cloud-based solutions can facilitate the efficient utilization of computational resources, thereby making the model scalable and accessible to a wider variety of applications.

The model proposed in this paper provides a solid foundation for future research and exploration in the field of sentiment analysis. By addressing future scopes such as customization, additional modalities, imbalanced dataset handling, interpretability, real-time analysis, multilingual support, adversarial robustness, and deployment scalability, researchers can continue to advance the field and make significant contributions to the application of sentiment analysis in various domains.

6. References

[1] J. Khan, N. Ahmad, S. Khalid, F. Ali and Y. Lee, "Sentiment and Context-Aware Hybrid DNN With Attention for Text Sentiment Classification," in IEEE Access, vol. 11, pp. 28162-28179, 2022, doi: 10.1109/ACCESS.2022.3259107.

[2] L. Yang, Y. Li, J. Wang and R. S. Sherratt, "Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning," in IEEE Access, vol. 8, pp. 23522-23530, 2020, doi: 10.1109/ACCESS.2020.2969854.

[3] Z. Li, R. Li and G. Jin, "Sentiment Analysis of Danmaku Videos Based on Naïve Bayes and Sentiment Dictionary," in IEEE Access, vol. 8, pp. 75073-75084, 2020, doi: 10.1109/ACCESS.2020.2986582.

[4] Y. Wang, G. Huang, J. Li, H. Li, Y. Zhou and H. Jiang, "Refined Global Word Embeddings Based on Sentiment Concept for Sentiment Analysis," in IEEE Access, vol. 9, pp. 37075-37085, 2021, doi: 10.1109/ACCESS.2021.3062654.

[5] L. Wang, J. Niu and S. Yu, "SentiDiff: Combining Textual Information and Sentiment Diffusion Patterns for Twitter Sentiment Analysis," in IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 10, pp. 2026-2039, 1 Oct. 2020, doi: 10.1109/TKDE.2019.2913641.

[6] B. -W. On, J. -Y. Jo, H. Shin, J. Gim, G. S. Choi and S. -M. Jung, "Efficient Sentiment-Aware Web Crawling Methods for Constructing Sentiment Dictionary," in IEEE Access, vol. 9, pp. 161208-161223, 2021, doi: 10.1109/ACCESS.2021.3129187.

[7] F. Alattar and K. Shaalan, "Using Artificial Intelligence to Understand What Causes Sentiment Changes on Social Media," in IEEE Access, vol. 9, pp. 61756-61767, 2021, doi: 10.1109/ACCESS.2021.3073657.

[8] L. Zhu, W. Li, Y. Shi and K. Guo, "SentiVec: Learning Sentiment-Context

Vector via Kernel Optimization Function for Sentiment Analysis," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 6, pp. 2561-2572, June 2021, doi: 10.1109/TNNLS.2020.3006531.

[9] D. She, J. Yang, M. -M. Cheng, Y. -K. Lai, P. L. Rosin and L. Wang, "WSCNet: Weakly Supervised Coupled Networks for Visual Sentiment Classification and Detection," in IEEE Transactions on Multimedia, vol. 22, no. 5, pp. 1358-1371, May 2020, doi: 10.1109/TMM.2019.2939744.

[10] U. Sehar, S. Kanwal, K. Dashtipur, U. Mir, U. Abbasi and F. Khan, "Urdu Sentiment Analysis via Multimodal Data Mining Based on Deep Learning Algorithms," in IEEE Access, vol. 9, pp. 153072-153082, 2021, doi: 10.1109/ACCESS.2021.3122025.

[11] E. Zuo, H. Zhao, B. Chen and Q. Chen, "Context-Specific Heterogeneous Graph Convolutional Network for Implicit Sentiment Analysis," in IEEE Access, vol. 8, pp. 37967-37975, 2020, doi: 10.1109/ACCESS.2020.2975244.

[12] Y. S. Mehanna and M. B. Mahmuddin, "A Semantic Conceptualization Using Tagged Bag-of-Concepts for Sentiment Analysis," in IEEE Access, vol. 9, pp. 118736-118756, 2021, doi: 10.1109/ACCESS.2021.3107237.

[13] Z. Ren, G. Zeng, L. Chen, Q. Zhang, C. Zhang and D. Pan, "A Lexicon-Enhanced Attention Network for Aspect-Level Sentiment Analysis," in IEEE Access, vol. 8, pp. 93464-93471, 2020, doi: 10.1109/ACCESS.2020.2995211.

[14] H. T. Phan, V. C. Tran, N. T. Nguyen and D. Hwang, "Improving the Performance of Sentiment Analysis of Tweets Containing Fuzzy Sentiment Using the Feature Ensemble Model," in IEEE Access, vol. 8, pp. 14630-14641, 2020, doi: 10.1109/ACCESS.2019.2963702.

[15] R. Obiedat, D. Al-Darras, E. Alzaghoul and O. Harfoushi, "Arabic Aspect-Based Sentiment Analysis: A Systematic Literature Review," in IEEE Access, vol. 9, pp. 152628-152645, 2021, doi: 10.1109/ACCESS.2021.3127140.

[16] H. Liu, X. Chen and X. Liu, "A Study of the Application of Weight Distributing Method Combining Sentiment Dictionary and TF-IDF for Text Sentiment Analysis," in IEEE Access, vol. 10, pp. 32280-32289, 2022, doi: 10.1109/ACCESS.2022.3160172.

[17] F. Yin, Y. Wang, J. Liu and L. Lin, "The Construction of Sentiment Lexicon Based on Context-Dependent Part-of-Speech Chunks for Semantic Disambiguation," in IEEE Access, vol. 8, pp. 63359-63367, 2020, doi: 10.1109/ACCESS.2020.2984284.

[18] X. Zhang, J. Xu, Y. Cai, X. Tan and C. Zhu, "Detecting Dependency-Related Sentiment Features for Aspect-Level Sentiment Classification," in IEEE Transactions on Affective Computing, vol. 14, no. 1, pp. 196-210, 1 Jan.-March 2023, doi: 10.1109/TAFFC.2021.3063259.

[19] H. Liang, U. Ganeshbabu and T. Thorne, "A Dynamic Bayesian Network Approach for Analysing Topic-Sentiment Evolution," in IEEE Access, vol. 8, pp. 54164-54174, 2020, doi: 10.1109/ACCESS.2020.2979012.

[20] H. Silva, E. Andrade, D. Araújo and J. Dantas, "Sentiment Analysis of Tweets Related to SUS Before and During COVID-19 pandemic," in IEEE Latin America Transactions, vol. 20, no. 1, pp. 6-13, Jan. 2022, doi: 10.1109/TLA.2022.9662168.

[21] S. Zhang, D. Zhang, H. Zhong and G. Wang, "A Multiclassification Model of Sentiment for E-Commerce Reviews," in IEEE Access, vol. 8, pp. 189513-189526, 2020, doi: 10.1109/ACCESS.2020.3031588.

[22] C. R. Aydin and T. Güngör, "Combination of Recursive and Recurrent Neural Networks for Aspect-Based Sentiment Analysis Using Inter-Aspect Relations," in IEEE Access, vol.

8, pp. 77820-77832, 2020, doi: 10.1109/ACCESS.2020.2990306.

[23] F. Huang, X. Li, C. Yuan, S. Zhang, J. Zhang and S. Qiao, "Attention-Emotion-Enhanced Convolutional LSTM for Sentiment Analysis," in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 9, pp. 4332-4345, . 2022, doi: 10.1109/TNNLS.2021.3056664.

[24] G. Zhai, Y. Yang, H. Wang and S. Du, "Multi-attention fusion modeling for sentiment analysis of educational big data," in Big Data Mining and Analytics, vol. 3, no. 4, pp. 311-319, Dec. 2020, doi: 10.26599/BDMA.2020.9020024.

[25] S. Poria, D. Hazarika, N. Majumder and R. Mihalcea, "Beneath the Tip of the Iceberg: Current Challenges and New Directions in Sentiment Analysis Research," in IEEE Transactions on Affective Computing, vol. 14, no. 1, pp. 108-132, 1 Jan.-March 2022, doi: 10.1109/TAFFC.2020.3038167.

[26] O. Dewangan, M. Mishra, “An Implementation of Multiple Modalities of Sentiments Obtained Using Machine Learning Algorithms” in Journal of Interdisciplinary Cycle Research ,Volume XII, Issue XI, November, 2020 pp. 849-856.

[27] O. Dewangan & M. Mishra. (2021). An Approach of Multimodal Sentiment Analysis using Machine Learning in Webology, vol. 18(6), 8491–8503.

[28] O. Dewangan & M. Mishra. (2022). An Implementation of Sentiment Analysis with Multiple Modalities using a Machine Learning in Harbin Gongye Daxue Xuebao/Journal of Harbin Institute of Technology, vol. 54(8), 378–386, doi: 10.11720/JHIT.54082022.36.