5G Network in Content Based Emotion Detection by Sentimental Analysis Integrated with Opinion Mining and Deep Learning Architectures

Nouby M. Ghazaly
Professor, Faculty of Engineering,
South Valley University, Qena 83523, Egypt.
https://orcid.org/0000-0001-6320-1916

Abstract:
The rapid growth of social networking sites in the Internet era has made them a necessary tool for sharing emotions with the entire world. To extract emotions from text, a variety of tools and approaches are available in fields of opinion mining as well as sentiment analysis. These researches propose novel technique opinion mining based emotion detection from the input social content using deep learning architectures. Here the input has been obtained as social media content based on opinion mining by 5G networks. The input has been processed for noise removal, smoothing and normalization. This processed input has been segmented using Markov model based convolutional neural networks (MCCNN). The segmented data has been classified using Canonical Correlation Analysis/Bayesian neural network. An opinion mining method that analyzes statements regarding computer programming and predicts or recognizes their polarity was implemented, along with an earlier module that was integrated into an intelligent learning environment. These three steps made up the creation of the module. We assessed the corpus, text polarity precision, and emotion recognition. Experimental analysis has been carried out for various social media content collected by opinion mining in terms of accuracy, precision, recall, F-1 score, AUC. Proposed technique attained accuracy of 99%, precision of 96%, recall of 96%, F-1 score of 95%, AUC of 89%.

Keywords: opinion mining, emotion detection, social media content, deep learning, segmentation, classification

1. Introduction:
One approach to convey your ideas or messages is through language. One effective medium for communicating your thoughts, feelings, and emotions is written text. Languages are utilized for more than just communication; they also convey the emotions that go along with it. Writing is a simple way to communicate emotions. Humans are capable of experiencing a wide range of emotions since they encounter them frequently in their daily lives [1]. A person experiences a variety of emotional states on a daily basis, including happiness, fear, rage, and sadness. Sentiment analysis/emotion detection is the term for the classification of text in certain emotional states using a computer. Sentiment classification is the process of categorizing a text based on the sentimental information it contains. Opinion mining (OM) is method of identifying the sentiments or feelings included in text. OM is method of removing views from text. Mining Text-based opinions can be helpful for learning about a user's perceptions about a location, an occasion, or a product [2]. Any type of text can be mined for opinions. Sentiment Analysis and emotion detection have a small distinction. Emotion Detection uses a greater set of emotions to separate text, whereas Sentiment Analysis divides text into two binary states. Numerous emotional states, like as happiness, fear, anger, brevity, surprise, or contempt, are faced every day. The fields of study known as "Sentiment Analysis" and "Opinion Mining" are same. Process of finding, examining, and extracting attitudes, sentiments, and views represented in text is what is originally referred to as sentiment analysis [3]. Due to widespread use of social media platforms with cameras, such as Facebook, Twitter, and Weibo, multimedia content, such as photographs and videos, play a significant role in communicating data about people's feelings as well as ideas. Many fresh concepts have surfaced in this promising field in recent years, particularly for visual sentiment analysis (SA). Deep learning, for instance, has started to be utilized for SA of many forms of social media data and has seen considerable success in field of AI [4]. In light of this, an in-depth analysis of social multimedia SA is highly necessary since it will be important, particularly for those who are just learning about visual SAas well as multimodal SA [5]. We
summarize literature on the subject and provide an overview of current methodologies from a macroscopic point of view in light of the dearth of comprehensive surveys on social multimedia SA. Extraction of sentiment from social media material has generally received a lot of attention. Multimodal sentiment analysis is still in its infancy, despite some advancements in visual sentiment analysis. Due to fact that audio data is typically found in videos rather than independently in social networks, audio-only sentiment analysis for social media is uncommon. Therefore, multimodal sentiment prediction makes use of the combination of audio-visual information [6]. The contribution of this paper is as follows:

1. To propose novel technique opinion mining based emotion detection from the input social content using deep learning architectures
2. To segment input data using Markov model based convolutional neural networks (MMCNN)
3. To classify segmented data using Canonical Correlation Analysis Bayesian neural network

2. Review of literature:
There is no universally applicable solution because the majority of the research in this field focuses on sentence-level text analysis, treating the work as a general issue without taking into account various sorts of sentences [7]. Deep learning has become popular in recent years and is utilized in numerous applications that carry out complex tasks, including speech recognition methods, facial recognition systems, and object detection, to name a few examples [8]. For many years, the hardware restrictions hindered the widespread adoption of DL techniques. However, as noted in Ref. [9], the maturity of software frameworks and the rise in processing power in recent years have produced encouraging results, which has greatly increased interest in working with DL. As an illustration, the authors of Ref. [10] employed an LSTM (Long Short-Term Memory) method to predict sentiment on Twitter. They compare many models, including SVM, Bernoulli Naive Bayes, and RNN (Recurrent Neural Networks), in various configurations, during their trials. Stanford Twitter Sentiment Corpus was corpus that was used. They outperformed the other models in their testing, with an accuracy rate of 87.2%. To predict assessments of texts with a valence scale, authors in Ref. [11] present a CNN-LSTM method made up of two components: regional CNN (Convolutional Neural Network) and LSTM. We used open-source tools to develop the OM module, including the Python programming language [12] and sci-kit-learn package [13], which provides a more comprehensive collection of OM-specific features. We utilized snowball method of NLTK library to carry out NLP operations like breaking words down to their base [14]. We used the TASS corpus to train the OM classifier [15]. According to Reference [16], the authors created a technology to automatically analyze the contributions made by students and teachers in community forums with the intention of methodically mining the thoughts they express. In order to assess the level of positivity in comparison phrases, other OM methods focus on the task of recognizing them [17]. Additionally, research has been done on classification tasks involving conditional sentences [18]. The authors of [19] conduct named entity recognition (NER) studies on Turkish tweets and present their findings. The impacts of pre-processing layers on the sentiment categorization of Turkish social media messages are reported in work [20]. While using microblogging, particularly on Twitter and Facebook, has several advantages, such as the variety and velocity of data [21], some microblogging texts may contain sarcasm, which makes it difficult to deal with such a large amount of data by only looking at some of its properties [22].

3. Proposed methodology:
This section discusses novel technique opinion mining based emotion detection from the input social content using deep learning architectures. Here the input has been obtained as social media content based on opinion mining by 5G networks. The input has been processed for noise removal, smoothening and normalization. This processed input has been segmented using Markov model based convolutional neural networks (MMCNN). The segmented data has been classified using Canonical Correlation Analysis Bayesian neural network. The proposed architecture is represented in figure-1.
Pre-processing of train data makes use of NLP, a method used to understand computer information and manage human interactions. Additionally, the text comments that are provided to the model are being pre-processed. Both pieces of information are transmitted to the sentiment library, where the pre-processed data's features are extracted. We frequently obtain the trained model for the knowledge sets from this. The classifier receives the input and categorizes the emotions as either positive or negative. The second classifier receives this data and further classifies the positive and negative feelings. The favourable traits are then divided into categories such as zeal, fun, love, happiness, neutral, relief, and surprise. The unfavorable traits are broken down into furious, bored, hateful, void-filled, depressed, and worried. The findings of the model are then provided by the classifier's prediction of the test input's output. Since they have a URL id, text comments on tweets are pre-processed. We delete any URLs and steer clear of any undesirable places because we have a tendency to not think about any addresses. The pre-processing stage is where these processes are carried out.

Markov model based convolutional neural networks (MMCNN) based segmentation:
Let S be a finite state space. S's constituent parts might be vectors. A state is referred to as each i ∈ S. The probability distribution \( \pi = \{\pi_i \mid i \in S\} \) is described in few probability space \( (\Omega, F, P) \) moving forward. As a result, the total mass satisfies the condition that \( 0 \leq \pi_i \leq 1 \) for all \( i \in S \). For each \( \sum_{i \in S} \pi_i = 1 \), define \( X(t) \) as a random variable (vector). Time is now denoted by the symbol t, which might be discrete or continuous. Idea put forth here seeks to simulate a production line step by step using a stochastic process in which likelihood that \( X(t + \delta t) \) depends only on value of X at time \( t \) is considered. \( X(t) \) is a Markov process, in other words. Let \( X(s) \) denote history of \( X \) values prior to time \( s \leq t \), and let \( z \) denote a potential value for \( X(t + \delta t) \). Then, eq.(1) fulfills the condition.

\[
\mathbb{P}[X(t + \delta t) = z \mid X(s) = x(s), s \leq t] = \text{Eq. 1}
\]

When a process is specified as having discrete times, these will typically be numbered by 0, 1, 2,..., or in another suitable manner, depending on the situation. The notation 0, \( \delta \), 2\( \delta \), . . . . , or simply \( \delta t \), is utilized in continuous time scenario. In a similar manner, eq. (2) will stand for the probability of a transition from I to j occurring within a unit of time.

\[
P_{ij} = \mathbb{P}[X(t + 1) = j \mid X(t) = i] = \text{Eq. 2}
\]

\( X(t) = I \) indicates that random quantity \( X(t) \) is in state I for a Markov chain. Additionally, the normalization eqn established by \( \sum \pi_i(t) = 1 \) is satisfied by \( \pi_i(t) = P[X(t) = i] \). For all \( t > u > s \), it also satisfies the Chapman-Kolmogorov equation in sense that \( P_k(s, t) = \sum_i P_{ji}(s, u) P_k(u, t) \), where \( P_k(s, t) = P[X(t) = k \mid X(s) = j] \).

![Figure 1: Overall proposed architecture](image-url)
Consequently, it follows from (3) and the Law of Total Probability that it is likewise true.
\[ \pi_j(t + 1) = \sum_i P_{ij} \pi_j(t) \] (3)
with \( P = \text{P}(X(1) = j \mid X(0) = i) \). Equivalently, we may write eq. (4)
\[ \pi^t = \left( \pi_1^t, \pi_2^t, \ldots, \pi_n^t \right), \quad P = \left( \begin{array}{cccc}
p_{11} & \cdots & p_{1n} \\
\vdots & \ddots & \vdots \\
p_{n1} & \cdots & p_{nn}
\end{array} \right) \text{ and } \theta = \left( \begin{array}{c} 1 \\
1 \\
1 \\
1
\end{array} \right) \]
\[ \pi^{t+1} = \pi^t \theta, \theta^T \pi^t = 1 \] (4)

Probability graph method of the HMM is shown in Figure 3, where \( X_1:T = (X_1, X_2, \ldots, X_T) \) denotes an observable variable and \( Y_1:T = (Y_1, Y_2, \ldots, Y_T) \) denotes a hidden variable. Latent variables form a Markov chain in eqn (5).
\[ p(x, y; \theta) = \prod_{t=0}^T p(y_t \mid y_{t-1}, \theta) p(x_t \mid y_t, \theta) \] (5)

Observables and hidden variables are represented by \( x \) and \( y \), respectively. The output probability is denoted by \( p(x_t \mid y_t, \theta) \), the transition probability is denoted by \( p(y_t \mid y_{t-1}, \theta) \), \( s \), and the parameters for both are denoted by \( t \) and \( s \).

Figure 2. HMM. \( X_{1:T} = (X_1, X_2, \ldots, X_T) \) represents an observable variable, and \( Y_{1:T} = (Y_1, Y_2, \ldots, Y_T) \) represents a hidden variable.

The joint probability density of generative method is eq (6), according to the Graphical Model.
\[ p(x, z, z; \theta) = p(z_t) \prod_{i=1}^n p(z_{t} \mid z_{t-1}, z_{i}; \phi_{zi}^t) p(x_{t} \mid z_{t}; \phi_{xi}^t) \] (6)

The parameters are \( \theta = (\theta_1, \theta_2) \). The generative model’s parameter is \( \theta_2 \), but the prior probability distribution function of \( z_i \)’s parameter is \( \theta_1 \).
\[ q(z_{t}, z \mid x; \phi) = q(z_{t} \mid x; \phi_{t}) \prod_{i=1}^n q(z_{t} \mid x; \phi_{xi}^t) \phi_{xi}^t \] (7)

\( \phi = (\phi_1, \phi_2) \) are specifications. Given that preceding local latent variable \( z_i \) and the global latent variable \( z_t \) determine local latent variable \( z_i \) in the generative network, we obtain eq (8)
\[ p(x_t, z; \theta_t) = p(z_t) \prod_{i=1}^n p(z_{t} \mid z_{t-1}, z_{i}; \phi_{zi}^t) \] (8)

The Evidence Lower Bound (ELBO) results in following under the assumptions of (9) and (10).
\[ \log p(x) = \log \int_{z_t} \int_{z} p(x, z, z_t) \] (9)
\[ = \log E_{q(z_t|x)} \prod_{i=1}^n p(z_{t} \mid z_{t-1}, z_{i}) \prod_{i=1}^n q(z_{t} \mid x_t) \] (10)

If we create the following eqn (11):
\[ I = E_{q(z_t|x)} \log \frac{q(z_{t} \mid x_t)}{p(z_{t} \mid z_{t-1}, z_t)} \] (11)
\[ \begin{align*}
&= \int_{z_t} \int_{z} q(z_{t} \mid x_t) \prod_{i=1}^n q(z_{t} \mid x_t) \\
&\quad \log \frac{q(z_{t} \mid x_t)}{p(z_{t} \mid z_{t-1}, z_t)} \\
&= \int_{z_t} \int_{z} q(z_{t} \mid x_t) \prod_{i=1}^n q(z_{t} \mid x_t) \\
&\quad \log \frac{q(z_{t} \mid x_t)}{p(z_{t} \mid z_{t-1}, z_t)} \\
&= E_{q(z_t|x)} E_{q(z_{t-1}|x_{t-1})} q(z_{t} \mid x_t) \frac{q(z_{t} \mid x_t)}{p(z_{t} \mid z_{t-1}, z_t)} \\
&= E_{q(z_t|x)} E_{q(z_{t-1}|x_{t-1})} D_{KL}(q(z_{t} \mid x_t) \parallel p(z_{t} \mid z_{t-1}, z_t))
\end{align*} \]

we, thus, obtain following eq. (12).
\[ ELBO = \sum_{i=1}^n E_{q(z_t|x)} \log p(z_{i} \mid z_{i}; \theta_{zi}) \\
- D_{KL}(q(z_{i} \mid x; \phi_{zi}) \parallel p(z_{i})) \] (12)

Two KL divergences can be found for closedform solutions if consider that prior distribution as well as posterior distribution are both Gaussian distributions. ELBO becomes the following eq. (17).
\[ ELBO = \sum_{i=1}^n E_{q(z_t|x)} \log p(z_{i} \mid z_{i}; \theta_{zi}) \\
- D_{KL}(q(z_{i} \mid x; \phi_{zi}) \parallel p(z_{i})) \] (13)

\[ H^w = \text{Transformer}_{\text{word}}(x_t), 1 \leq i \leq n \]
\[ H^s = \text{Transformer}_{\text{sen}}(H^w_i), 1 \leq i \leq n \] (14)

We utilized a 3-layer MLP network to learn local posterior distribution as well as global posterior distribution by eq.
The technique aims to estimate the supplied dataset using function eq. (16):

\[
f(\mathbf{x}) = \sum_{j=1}^{M} c_j \phi(\|\mathbf{x} - \xi_j\|)
\]

where approximation function f(x) is sum of M RBFs, every weighted by the appropriate coefficient c_j and connected with a unique reference point \(\xi_j\). As a result, the vector of weights \(c = (c_1, ..., c_M)^T\) must be determined, which results in the minimization of the quadratic form by equation (17):

\[
\frac{1}{2} \mathbf{c}^T \mathbf{Q} \mathbf{c} = \frac{1}{2} \mathbf{c}^T \mathbf{Q} \mathbf{c} - \mathbf{A}^T (\mathbf{A} \mathbf{c} - \mathbf{h}) = 0
\]

where \(\mathbf{Q} = (Q_{i,j})\) is a symmetric matrix. Equation (20) is then solved.

It is possible to determine the approximate value in a manner similar to interpolation.

\[
f(\mathbf{x}) = \sum_{j=1}^{M} c_j \phi(\|\mathbf{x} - \xi_j\|)
\]

where approximation function f(x) is sum of M RBFs, every weighted by the appropriate coefficient c_j and connected with a unique reference point \(\xi_j\). It is evident from eq. (21) that we obtain an overdetermined LSE for provided dataset:

\[
h_i = f(\mathbf{x}_i) = \sum_{j=1}^{M} c_j \phi(\|\mathbf{x}_i - \xi_j\|) = \sum_{j=1}^{M} c_j \phi(\|\mathbf{x}_i - \xi_j\|)
\]

The matrix equation can be used to represent the linear system of equations (22):

\[
\mathbf{A} \mathbf{c} = \mathbf{h}
\]

where N M is total number of rows and M is total number of reference points (c1, c2, ..., cM) T. Now, the approximation can be represented in the following manner using equation (23):

\[
f(\mathbf{x}) = \sum_{j=1}^{M} c_j \phi(\|\mathbf{x} - \xi_j\|) + P_h(\mathbf{x})
\]

where \(\xi_j\) are user-specified reference points. This leads to the LSE being solved by equation (28):

\[
h_i = f(\mathbf{x}_i) = \sum_{j=1}^{M} c_j \phi(\|\mathbf{x}_i - \xi_j\|) + P_h(\mathbf{x}_i)
\]

\[
= \sum_{j=1}^{M} c_j \phi(\|\mathbf{x}_i - \xi_j\|) + P_h(\mathbf{x}_i)
\]

For example, for d = 2, the vectors \(\mathbf{x}_i\) and an are given as \(\mathbf{x}_i = (x_i, y_i)^T\) and \(a = (a_x, a_y)^T\). We can express E by equation (25) using the matrix notation.

\[
\begin{pmatrix}
\phi_{1,1} & \cdots & \phi_{1,M} & x_1 & y_1 & 1 \\
\vdots & & \vdots & \vdots & \vdots & \vdots \\
\phi_{N,1} & \cdots & \phi_{N,M} & x_N & y_N & 1 \\
\end{pmatrix}
\begin{pmatrix}
c_1 \\
\vdots \\
c_M \\
ax \\
a_y \\
a_0 \\
\end{pmatrix}
= \begin{pmatrix}
h_1 \\
\vdots \\
h_N \\
\end{pmatrix}
\]

(25)

Without using eq. (26), it expresses approximant (sk (x)) in terms of radial basis function:

\[
s_k(\mathbf{x}) = \sum a_j \varphi(r(\mathbf{x}, \mathbf{x}_j), \varepsilon)
\]

where the interpolation requirements are applied to the approximant, and aj stands for the coefficients related to each basis. By equating approximant at points \(\mathbf{x}_i\) to given data at same position (f(\(\mathbf{x}_i\))), this can be accomplished.

\[
s_k(\mathbf{x}_i) = \sum a_j \varphi(r(\mathbf{x}_i, \mathbf{x}_j), \varepsilon) = f(\mathbf{x}_i)
\]

(27)

A linear set of eqns that are written as a matrix is equation (28):

\[
[f_{|XX}] a = [f_{|X}]
\]

(28)

If no repeated point is used, the matrix \(\varphi\) \(\chi\) is symmetric and invertible. Because it lowers the computational cost, this characteristic is useful when solving PDEs. To solve the PDE by eq. (29) the differential operator is used on approximant at interior locations.

\[
\left[ L \varphi \right]_{\chi} [a] = \left[ L f \right]_{\chi}
\]

(29)

Equation (30) substitutes coefficients in order to approximate the differential operator. Despite having a greater accuracy, this method is not recommended for solving PDEs due to its high computational cost. In second method, coefficients \(j\) are determined analytically by inverting matrix \(\varphi\) \(\chi\) from Equation (34). Remedy is used in place of

\[
\left[ L \varphi \right]_{\chi} [f_{|XX}]^{-1} [f_{|X}] = \left[ L f \right]_{\chi}
\]

(30)

It is possible to create discrete differential operator (W) without directly evaluating coefficients. Evaluation must be performed once before time iteration may begin. Partial differential equation’s solution can then be obtained by

\[\text{Equation (30)}\]
applying the discrete differential operator \(W\) at every time-step.

\[
\begin{align*}
    s_h(x) & = \sum_{i}^{x\varepsilon_{a}} a_j \phi(\tau(x, \xi), \epsilon) \bigg|_{\xi=x_j} + \\
    \sum_{i}^{\varepsilon(\varepsilon, 2)} \beta_i \mathcal{L} \phi(\tau(x, \xi), \epsilon) \bigg|_{\xi=x_j}
\end{align*}
\]

By equating approximant at points \(x_i\) to \(f(x_i)\) by eq (32), it is possible to apply interpolation constraints on approximant in order to calculate interpolation coefficients.

1. \( s_h(x_i) = \sum_{i}^{x\varepsilon_{a}} a_j \phi(\tau(x_i, \xi), \epsilon) \bigg|_{\xi=x_j} + \\
2. \sum_{i}^{\varepsilon(\varepsilon, 2)} \beta_i \mathcal{L} \phi(\tau(x_i, \xi), \epsilon) \bigg|_{\xi=x_j} = f(x_i).

(32)

It can be expressed as a system of equations using matrices by eq. (33):

\[
\begin{bmatrix}
    \mathcal{L} \phi_{JI} & \phi_{IB} \\
    \mathcal{L} \phi_{BI} & \phi_{BB}
\end{bmatrix}
\begin{bmatrix}
    \alpha_j \\
    \beta_i
\end{bmatrix} =
\begin{bmatrix}
    f_{Ij} \\
    f_{Bj}
\end{bmatrix}
\]

(33)

Although interpolation matrix is not symmetric, it can be used to derive coefficients \(a_j\) and \(\beta_i\) in a manner similar to Kansa's method. To solve PDE by equation (34) it is essential to first build the PDE matrix by applying differential operator \(L\) to interior points.

\[
\begin{bmatrix}
    \mathcal{L} \phi_{JI} & \phi_{IB} \\
    \mathcal{L} \phi_{BI} & \phi_{BB}
\end{bmatrix}
\begin{bmatrix}
    \alpha_j \\
    \beta_i
\end{bmatrix} =
\begin{bmatrix}
    f_{Ij} \\
    f_{Bj}
\end{bmatrix}
\]

(34)

Equation (34) in the interpolation system of equations can be solved to yield the coefficients. The discrete differential operator \(W\) is then obtained by substituting them in Equation (35).

\[
[A_{x}][A_{w}]^{-1} \begin{bmatrix}
    f_{Ij} \\
    f_{Bj}
\end{bmatrix} =
\begin{bmatrix}
    \mathcal{L} \phi_{JI} & \phi_{IB} \\
    \mathcal{L} \phi_{BI} & \phi_{BB}
\end{bmatrix}
\begin{bmatrix}
    \alpha_j \\
    \beta_i
\end{bmatrix} =
\begin{bmatrix}
    f_{Ij} \\
    f_{Bj}
\end{bmatrix}
\]

(35)

By using eq.(36), it may be found for Gaussian function as follows.

\[
\langle \phi, H_n(\epsilon) \rangle = \begin{cases} 
    \frac{nn^n}{(n/2)!}R^{n+1} \sqrt{\pi}(-1)^{n/2} & \text{even } n \\
    0 & \text{odd } n
\end{cases}
\]

(36)

where \(B_n = (1 \ n \ 2/ \ 2 \ n \ 2/ \ 2)!\) is a variable that changes with \(n\). By evaluating derivative of Equation (37), it is clear that first two derivatives based on space are functions of first 2 derivatives based on \(r\).

\[
\frac{\partial a_n}{\partial r} = B_n \frac{r^{n+1} - r^{n-1}}{R^{n+3}}
\]

(37)

To derive an explicit equation for derivatives based on space, the previous two Equations (38) are substituted in Equations (39).

\[
\begin{align*}
    \frac{\partial a_n}{\partial r} &= B_n \frac{nr^{n-1} - r^{n+1}}{R^{n+3}} \\
    \frac{\partial^2 a_n}{\partial r^2} &= B_n \frac{2r^{n+2} - (5n+1)r^{n-1}r^{n-2}}{R^{n+5}}
\end{align*}
\]

(38)

The study criticized the following artificial neural network model for image operation. Given a set of input vectors \(x\) and their associated transpose as shown in equations (39), the picture input is in a vectors matrix.

\[
\begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_m
\end{bmatrix} = \begin{bmatrix}
    x_{k1}x_{k2}x_{k3} ... x_{km}
\end{bmatrix}
\]

(39)

And associated weight vector, \(w^{****k}\, k\), to input vector \(x^{****k}\, k\), is expressed in (40)

\[
\begin{bmatrix}
    \vec{w}_j \\
    \vec{w}_k
\end{bmatrix} = \begin{bmatrix}
    w_{j1}(k) & w_{j2}(k) & ... & w_{jm}(k)
\end{bmatrix}
\]

(40)

Output \(y^{****k}\) is expressed eq. (42):

\[
\begin{bmatrix}
    y_{k1} \\
    y_{k2} \\
    \vdots \\
    y_{km}
\end{bmatrix} = \sum_{j=1}^{m} \begin{bmatrix}
    w_{j1}(k) & w_{j2}(k) & ... & w_{jm}(k)
\end{bmatrix} \begin{bmatrix}
    x_{k1} \\
    x_{k2} \\
    \vdots \\
    x_{km}
\end{bmatrix}
\]

(43)

FIG. 3 provides an illustration of the neural network model and a graphical depiction.

Following network training, the following formula can be used to express the requirement for perfect recall:

\[
\vec{y} = \sum_{k=1}^{m} \begin{bmatrix}
    x_{k1} & \cdot & \cdot & \cdot \\
    x_{k2} & \cdot & \cdot & \cdot \\
    \vdots & \cdot & \cdot & \cdot \\
    x_{km} & \cdot & \cdot & \cdot
\end{bmatrix} \begin{bmatrix}
    y_{k1} \\
    y_{k2} \\
    \vdots \\
    y_{km}
\end{bmatrix}
\]

(44)

As shown, a more condensed form of equation (45) can be used.
\[
\hat{y} = \tilde{y}_{1} + \tilde{v}_{1}
\]

Where \( \tilde{v}_{j} = \sum_{m=j}^{n} (x_{m}^{T}, x_{j}) \tilde{y}_{k} \)  

(45)

Perfect recollection is contingent on Equation (49). The system's intended or anticipated output is denoted by \( y_{j} \), and the noise that results from remembering a pattern from memory is denoted by \( v_{j} \). This model has four levels: an input layer, two (2) hidden layers with 128 units or nodes each, and a top-level output layer. Figure 4 depicts the training block using Keras and TensorFlow library tools, whereas Figure 4 displays the actual layer configuration of the 128x128 CNN-ML model.

The 128x128 CNNML model is slightly modified in this model. Additionally, this model has four levels: an input layer, two (2) hidden layers with 256 nodes each, and a final output layer. Getting superior outcomes for layer parameters that create a strong association between the layers to ensure successful identification is the primary goal of CNN training. As indicated, the gradient descent approach is used in conventional CNN to optimize model parameters such as convolution filters as well as weights of fully linked layers. Due to relevance of last layer on classification outcomes, it is crucial to place the image in the appropriate class, which is accomplished by properly connecting the weights to the
Canonical Correlation Analysis Bayesian neural network based classification:

Canonical correlation (ρ) between two sets of multivariate random variables X and Y is covariance (Cov) between two variables normalized by geometric mean of variances (Var) of Xα and Yβ by eq. (46):

$$\rho = \frac{\text{Cov}(X\alpha, Y\beta)}{\sqrt{\text{Var}(X\alpha)} \sqrt{\text{Var}(Y\beta)}}$$  \hspace{1cm} (46)

where α and β are vectors of appropriate dimensions. Initially, CCA did not receive much attention from practitioners due to little understanding of the concept and absence of computer programs. A multivariate multiple regression representation of CCA allows for finding an evaluate of β, α and D(pk) in following model equation (47):

$$Y\beta = X\alpha D(\rho_\beta) + E$$  \hspace{1cm} (47)

The kth column of which contains the canonical weights for set of kth canonical variate pair, where β is a q x d matrix. E is a n x d matrix of errors. Given the pair of multivariate random variables (X, Y ), kernel CCA maximizes the eq. (48)

$$\rho_K = \max_{\alpha, \beta} \alpha^T K_X K_Y \beta$$  \hspace{1cm} (48)

where α = (α1, α2, ... , αm), β = (β1, β2, ... , βn), and Kn and Ky are Gram matrices [Kn]mnxn = k (X1, Xj) and [Ky]nxn = k (Y1, Yj) calculated from sample. Model is presented as eq. (49)

$$z \sim N(0, I_d), \quad \min(p, q) \geq d \geq 1 \quad X | z \sim N(az, \psi_1) \quad \alpha \in \mathbb{R}^{p \times d}, \quad \psi_1 \geq 0$$

$$Y | z \sim N(\beta z, \psi_2) \quad \beta \in \mathbb{R}^{q \times d}, \quad \psi_2 \geq 0.$$  \hspace{1cm} (49)

Consider N observations on two sets of standardized variables X and Y with eq. (50)

$$X = \{x_{ij}\}; \quad i = 1, 2, \ldots, N, j = 1, 2, \ldots, p \quad Y = \{y_{ij}\}; \quad i = 1, 2, \ldots, N, j = 1, 2, \ldots, q,$$

CCA seeks to maximize the correlation (ρ) between Xα and Yβ, (ap x 1) and (βq x 1) 3 the variance, Var(Xα) = Var(Yβ) = 1 by eq. (51):

$$\rho = \frac{\alpha^T R_{XY} \beta}{\sqrt{(\alpha^T R_{XX} \alpha)(\beta^T R_{YY} \beta)}}$$  \hspace{1cm} (51)

Of course, Cov is the covariance between U and V . (5) can be equivalently expressed as eq. (52),

$$\rho = \frac{\alpha^T R_{XY} \beta}{\sqrt{(\alpha^T R_{XX} \alpha)}(\beta^T R_{YY} \beta)}.$$  \hspace{1cm} (52)

Increase quantity a 0 RXY β subject to constraints α 0 RXα = β 0 RY β = 1: Introduce Lagrangian multipliers, then compute matrix derivatives, set them to zero and simplify. The solution leads to the normal eq. (53)

$$|R_{XX}^{-1} R_{XY} R_{YY} R_{XY}^T - K^2 | = 0$$

$$|R_{YY}^{-1} R_{XY} R_{XX} R_{XY}^T - K^2 | = 0.$$  \hspace{1cm} (53)

The values of K are obtained by solving the multivariate eigenvalue problem by eq. (54)

$$|R_{XX}^{-1} R_{XY} R_{YY} R_{XY}^T - K^2 | = 0$$

$$|R_{YY}^{-1} R_{XY} R_{XX} R_{XY}^T - K^2 | = 0.$$  \hspace{1cm} (54)

Further, the values of α and β can be obtained from Equations (55), (56) as follows:

$$\beta = \frac{1}{\rho} R_{XX}^{-1} R_{XY} \alpha$$

$$\alpha = \frac{1}{\rho} R_{YY}^{-1} R_{XY} \beta.$$  \hspace{1cm} (55)

After viewing a set of training data D = fxi, yigN I = 1, one can write joint posterior distribution of network specifications, p(w, b, SjY, X), which includes the covariance matrix, S, considered zero mean error, as eq. (57).

$$p(w, b, S \mid Y, X) = \frac{p(Y \mid w, b, S, X)p(w, b, S \mid X)}{p(Y \mid X)}$$

$$= \frac{p(Y \mid w, b, S, X)p(w, b, S \mid X)\text{d}wd\text{d}S}{\int_{w, b, S} p(Y \mid w, b, S, X)p(w, b, S \mid X)\text{d}wd\text{d}S}$$  \hspace{1cm} (57)

where w and b stand for BNN's weights and biases. Additionally, X and Y are specified as follows:

$$X = \{x_{11}, x_{21}, \ldots, x_{K1}\} \in \mathbb{R}^{N \times \text{dim}(x) \times 1} \quad \text{and} \quad Y = \{y_{11}, y_{21}, \ldots, y_{K1}\} \in \mathbb{R}^{N \times \text{dim}(y) \times 1}$$

Correspondingly. Due to conditional independence, conditional distribution p(w, b, SjY, X) is identical to p(w, b, S). By integrating the typically intractable

$$\int_{w, b, S} p(Y \mid w, b, S, X)p(w, b, S \mid X)\text{d}wd\text{d}S,$$

one can derive the marginal probability density function, p(YjX). The likelihood function p(Yjw, b, S) is represented as eq.(58) because the observation data D are statistically independent of one another.

$$p(Y \mid w, b, S, X) = \prod_{i=1}^{N} p(y_{1i} \mid w, b, S, x_{1i})$$  \hspace{1cm} (58)

Eq.(59) is predicated on the following measurement model-eqution.

$$y_{1i} = NN(x_{i}; w; b) + \varepsilon_{1i}, \quad \varepsilon_{1i} \sim \mathcal{N}(0, \sigma_{\varepsilon_{1i}}), k = \text{dim} (y_{1i})$$  \hspace{1cm} (59)
Just over 10,000 pieces of 62 Tweets concerning each -

opinion mining based emotion detection from the input social content using deep learning and existing technique in. Here the parameters analysed are accuracy, detection from the input social content using deep learning and existing technique in. For sentiment analysis in this paper, we used deep learning BNN prediction likewise regarded as an unknown covariance matrix \( p(X_{\text{train}}, Y_{\text{train}}) \), joint posterior distribution, where, following observation of a set of \( X_{\text{test}} \) for a collection of observed points \( X_{\text{test}} \). The normal distribution \( N() \) is represented by the mean and variance, and \( k = \text{dim.} \) It is assumed that \( w, b, \text{and } S \) are statistically independent for preceding \( p(w, b, s). \)

The above table shows comparative analysis of proposed and existing technique in opinion mining based emotion detection from the input social content using deep learning architectures. Here the parameters analysed are accuracy, precision, recall, F-1 score and AUC. The existing technique compared are CNN_LSTM and NLTK with proposed 5G_CBED_SAI_DL.

**Table 1** Comparative analysis between proposed and existing technique for various dataset

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Techniques</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1_Score</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford Sentiment Treebank</td>
<td>CNN_LSTM</td>
<td>91</td>
<td>81</td>
<td>84</td>
<td>81</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>NLTK</td>
<td>93</td>
<td>83</td>
<td>86</td>
<td>82</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>5G_CBED_SAI_DL</td>
<td>98</td>
<td>86</td>
<td>88</td>
<td>85</td>
<td>85</td>
</tr>
<tr>
<td>Twitter US Airline Sentiment</td>
<td>CNN_LSTM</td>
<td>89</td>
<td>85</td>
<td>91</td>
<td>88</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>NLTK</td>
<td>92</td>
<td>88</td>
<td>93</td>
<td>91</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>5G_CBED_SAI_DL</td>
<td>94</td>
<td>91</td>
<td>95</td>
<td>93</td>
<td>88</td>
</tr>
<tr>
<td>IMDB Movie Reviews Dataset</td>
<td>CNN_LSTM</td>
<td>95</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>NLTK</td>
<td>97</td>
<td>94</td>
<td>94</td>
<td>94</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>5G_CBED_SAI_DL</td>
<td>99</td>
<td>96</td>
<td>96</td>
<td>95</td>
<td>89</td>
</tr>
</tbody>
</table>

**Dataset Description:**

**Stanford Sentiment Treebank:** Just over 10,000 pieces of Stanford data from Rotten Tomatoes HTML files are included in this collection. One is the most unfavorable and twenty-five is the most favorable when rating the sentiments. Instead of only awarding points based on positive as well as negative words, Stanford’s deep learning model has been created to represent phrases based on their sentence structure.

**Twitter US Airline Sentiment:** Tweets concerning each major US airline are included in this sentiment analysis dataset, which dates back to February 2015. Each tweet is categorized as either favorable, unfavorable, or neutral. Twitter ID, sentiment confidence score, sentiments, airline name, retweet count, name, tweet coordinates, date and time of tweet, and location of tweet are among the features that are provided.

**IMDB Movie Reviews Dataset:** About 50,000 IMDB movie reviews are included in this sizable movie dataset. Only reviews that are extremely polarized are taken into account in this dataset. The number of both positive and negative reviews is equal, but negative review has a rating of \( \leq 4 \) out of 10, while positive review has a rating of \( \geq 7 \) out of 10.
Proposed technique attained accuracy of 98%, precision of 86%, recall of 88%, F-1 score of 85%, AUC of 82%; while existing CNN_LSTM attained accuracy of 91%, precision of 81%, recall of 84%, F-1 score of 81%, AUC of 79%, NLTK attained accuracy of 93%, precision of 83%, recall of 86%, F-1 score of 82%, AUC of 82% for Stanford Sentiment Treebank dataset as represented in figure 6(a)- (e).
Figure-7 Comparative analysis between propose and existing technique for in terms of Twitter US Airline Sentiment (a) Accuracy, (b) Precision, (c) recall, (d) F-1 score, (e) AUC

For Twitter US Airline Sentiment dataset obtained accuracy of 94%, precision of 91%, recall of 95%, F-1 score of 93%, AUC of 88%; while CNN_LSTM attained accuracy of 89%, precision of 85%, recall of 91%, F-1 score of 88%, AUC of ...
82%, NLTK obtained accuracy of 92%, precision of 88%, recall of 93%, F_1 score of 91%, AUC of 85% as shown in figure 7 (a)-(e).

Figure-8 Comparative analysis between propose and existing technique for in terms of IMDB Movie Reviews Dataset (a) Accuracy, (b) Precision, (c) recall, (d) F-1 score, (e)AUC

Proposed technique attained accuracy of 99%, precision of 96%, recall of 96%, F-1 score of 95%, AUC of 89%; while existing CNN_LSTM attained accuracy of 95%, precision of 92%, recall of 92%, F-1 score of 92%, AUC of 86%, NLTK attained accuracy of 97%, precision of 94%, recall of 94%.
F_1 score of 94%, AUC of 87% for IMDB Movie Reviews Dataset as shown in figure 8 (a)- (e).

We can infer the following conclusions from the analysis. (1) Multimedia sentiment analysis powered by deep learning will continue to be a hot topic. (2) Multimedia sentiment analysis, particularly for visual sentiment analysis, will benefit from the use of existing semantic analysis methods in related fields. (3) We should pay close attention to how current methods for textual SA can be applied to combination visual-textual sentiment analysis. The relationship between textual as well as visual material is considered. (4) For the audio-visual content in social networks, more effort should be put into audio, video, and multimodal sentiment analysis.

5. Conclusion:
Multimedia data has grown in importance as a means of conveying human emotions and viewpoints as a result of the social media revolution. A promising study area is sentiment analysis of multimedia content in social networks. Proposed framework of this research designed for opinion mining based emotion detection from the input social content using deep learning architectures. The input data segmentation is carried out using Markov model based convolutional neural networks (MCMCN). The segmented data has been classified using Canonical Correlation Analysis Bayesian neural network. For experimental analysis the parameters analysed are accuracy, precision, recall, F_1 score and AUC. Proposed technique attained accuracy of 99%, precision of 96%, recall of 96%, F-1 score of 95%, AUC of 89%. We'll concentrate on enhancing emotion recognition precision using various neural network designs in the future. For more efficient model training, we will also take into account utterance-level annotation of the ICT-MMDO dataset.

References:


