International Journal of Communication Networks and Information Security

ISSN: 2073-607X, 2076-0930

*Volume **14** Issue **02 Year 2022** Page **15:32***

Cognitive Computing for Multimodal Sentiment Sensing and Emotion Recognition Fusion Based on Machine Learning Techniques Implemented by Computer Interface System

[1] Dr. Arun Kumar Marandi, [2] Prof. Gordhan Jethava, [3] A Rajesh, [4] Mr. Sachin Gupta, [5] Shrddha Sagar, [6] Mrs. Sonia Sharma

[1] Associate Professor, Department of Computer Science, ARKA JAIN University, Jamshedpur, Jharkhand, India.

[2] Assistant Professor, Department of Computer Science and Engineering, Parul Institute of Technology, Parul University, Vadodara, Gujarat, India.

[3]Professor, Department of Computer Science and Engineering, Jain (Deemed-to-be University), Bangalore, India.

[4]Chancellor, Department of Management, Sanskriti University, Mathura, Uttar Pradesh, India.

[5]Professor, School of Computing Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh, India.

[6]Assistant Professor, Department of Computer Science Engineering, Chandigarh Engineering College, Jhanjeri, India.

[1] dr.arun@arkajainuniversity.ac.in, [2] gordhan.jethava@paruluniversity.ac.in, [3]a.rajesh@jainuniversity.ac.in, [4]chancellor@sanskriti.edu.in, [5]shrddha.sagar@galgotiasuniversity.edu.in, [6]sonia.j1220@cgc.ac.in

Article History

Received: 12 Feb 2022
Revised: 27 May 2022
Accepted: 20 June 2022

Abstract

In the last two decades, human emotion recognition and analysis have inspired a lot of interest, garnering a lot of research in neuroscience, psychology, cognitive science, and computer science. Multimodal emotion identification has gotten a lot of interest in field of affective computing, thanks to continued development of portable non-invasive human sensor technologies like BCI (brain–computer interfaces). This paper presents the cognitive computing based multimodal sentimental sensing and emotion recognition using machine learning architectures. Here the data has been collected based on the online review of the social media and twitter datasets. The collected dataset has been pre-processed and segmented to remove noise and unwanted numerical data. Then features have been extracted based on FFT (fast Fourier transform) based convolutional neural network (FFT-CNN) which extracts both numerical data and image data extraction. The extracted features have been classified based on transfer learning based SVM classification (TL-SVM). The experimental analysis has been carried out based on various datasets in terms of accuracy of 98%, precision of 88%, recall of 80%, F-1 score of 78% and mean square error of 45% for proposed FFT-CNN_TL-SVM.

Keywords: BCI, multimodal emotion recognition, sentimental sensing, FFT-CNN, TL-SVM.

1. Introduction

Professor Picard of Institute of Technology first established concept of "emotional computing" in his 1997 book Affective Computing. "Affective computing," she defined, is the computation of factors related to human emotion, triggered by human emotion, or capable of affecting emotion [1]. Affective computing research aims to create a more harmonic and significant HCI so that computers can have a higher and more comprehensive intelligence [2]. Voice, facial expression, posture, and other external expressions of human emotion are the most common. Human speech comprises both linguistic and non-linguistic data like emotional state of speaker. Because of the speaker's various emotional states, the same speech might often feel different to the listener. Because human speech contains dimensions that might reflect emotional traits, it can represent emotion. External expressions of emotion, such as facial expressions, are also essential since they contain emotional data. Study of facial expression recognition can help to advance emotion recognition research as well as the study of autonomous computer picture interpretation [3]. Face expressions are easily exaggerated by difficulties such as bad lighting, changing angles and obstructed locations, as voice emotion recognition is easily disrupted by noise in surrounding environment. As a result, single-modal emotion perception is limited. Researchers suggest a multimodal emotion recognition method based on the merging of speech and facial expression to increase overall recognition performance and learn from each other in diverse emotional aspects, which has substantial research implications for emotion detection in practise. Based on how different modal signals are processed at different levels, it can be characterised as signal-level fusion, feature-level fusion, decision-level fusion and hybrid fusion. Subjectivity detection as well as sentiment analysis are processes of automatically detecting the private states of the human mind, such as views, feelings, moods, behaviours, and beliefs [4]. The former is concerned with identifying sentiment information as objective or subjective, whereas later is concerned with inferring a positive or negative polarity. As a result, both tasks are typically classified as binary classification issues. The majority of sentiment analysis research has so far been done on text data. People are now using videos, photographs, and audio files to voice their thoughts on social media platforms, thanks to the proliferation of video cameras in every pocket and advent of social media. As a result, new ways for extracting opinions and sentiments from these many modalities have become crucial. In subject of audio-visual emotion recognition, a lot of study has been done. Some research has also been done on detecting emotion from videos by combining auditory, visual, and textual modalities. However, for both objectives, a unified framework is still lacking. There are also few literatures [5] that combine textual clues with audio as well as visual cues.

The contribution of this paper is as follows:

- To collect cognitive computed network-based data of social media, Facebook, and twitter
- To process the dataset to reduce noise and remove the unwanted numerical data
- To extract features using fast Fourier transform (FFT) based convolutional neural network (FFT-CNN)
- Then to classify extracted features using transfer learning based SVM (TL-SVM)
- The experimental results have been carried out based on accuracy, precision, recall, F-1 score

2. Related Works

Multimodal emotion recognition has advanced rapidly as fusion algorithm research has progressed. Multimodal fusion can boost recognition rates while also increasing robustness [6]. Physiological signal + emotional behaviour combination and combination of several emotional modalities are now the most frequent multimodal emotion detection approaches. Feature-level fusion, decision-level fusion (late stage), and hybrid fusion are examples of multimodal fusion methods. EF-LSTM [7] is a common early fusion model that combines the feature representations of text, speech, and image to create a multimodal representation, which is subsequently fed into LSTM for encoding. After decoding, late fusion [8] happens; it is a fusion at decision level that can extract interaction data within modalities but not between modalities. The first two fusion processes are combined in hybrid fusion. In [9], the author proposed a paradigm for affective experiences, in which four pleasure-

arousal combinations [10] are grouped into eight "affective families" based on dominance. The framework [e.g. 3] of eight affective families and twenty-four affective experiences: Zhao et al. [26] showed a social media-based interactive visual analytic tool for determining personal emotion type. They ran a series of research to evaluate the tool. To begin, they enlisted the help of ten employees from a well-known IT firm to identify their own tweets using the following criteria: valence, arousal, and dominance levels on a 9-point scale [11], emotional category from Plutchik and Robert's categorical model for primary emotion pairs [12], such as anger-fear, anticipation-surprise, joy-sadness, and trust-disgust [13], and emotional styles [14]. They obtained 308 tagged tweets from 60 mood segments in all. Second, they conducted two user tests to analyse PEARL's effectiveness in measuring others' emotional styles, as well as a Mechanical Turk study to assess PEARL's effectiveness in assessing others' emotional styles. With NVIE (Natural Visible and Infrared Facial Expression) DB, Wang et al. exploited IR pictures for emotion recognition using DBM (Deep Boltzmann Machine) and reported an accuracy of 62.9 percent [15]. After integrating more unlabelled data from other databases to increase feature learning performance, the authors reported an accuracy of 68.2 percent. DCT (Discrete Cosine Transform) with LE (Local Entropy) and LSD (Local Standard Deviation) features were used by the author in [16] for AER using the Imaging, Robotics, and IRIS (Intelligent Systems) DB. Another study utilised kNN and SVM to create a facial emotion detection model in thermal pictures. Using LE features with kNN, greatest accuracy of 90% was achieved. Another example of use of thermal images for AER was the use of DBM to accurately classify emotions with a 51.3 percent accuracy [17-19]. In addition, Basu et al. achieved an accuracy of 87.50 percent by using thermal facial photos as well as employing moment invariant, histogram statistics and multi-class SVM. A clever thermal system with an accuracy of 89.89 percent was also developed to identify emotions using infrared images. Researchers in [20] and [22] used visual and thermal face pictures as well as speech to perform AER and reported an accuracy of 85 percent. In this work, the fusion was simple and focused on the greatest output of the three modalities.

3. System Model

This section discusses the proposed system model for cognitive computing based multimodal sentimental sensing and emotion recognition. Here the machine learning techniques has been used with FFT integrated neural network in feature extraction and transfer learning integrated in classification. The proposed architecture is given in below figure-1.

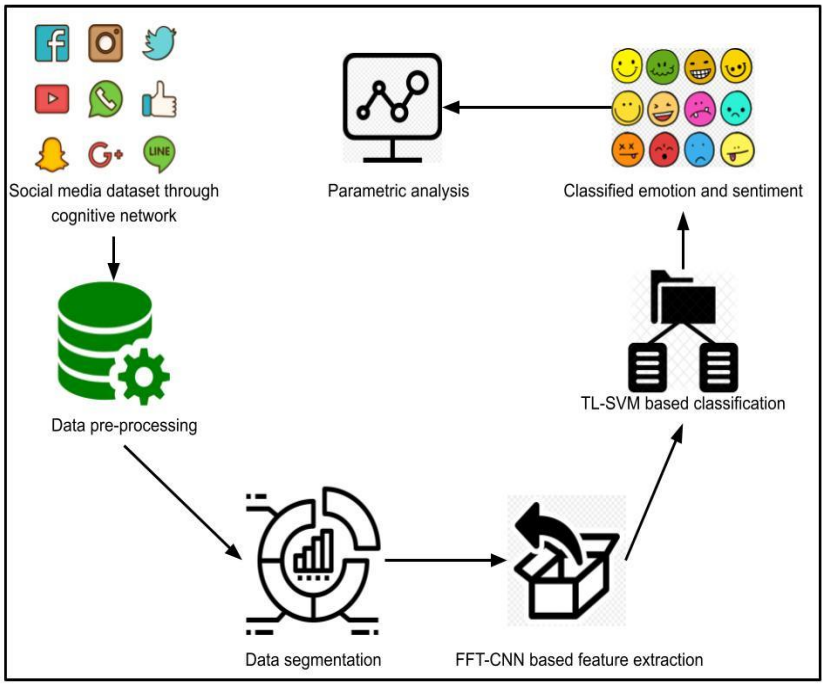


Figure-1 *Overall proposed architecture*

The linear predictor that was used to predict the behaviour of social media reviewers. For j=1, 2,, n, the fundamental form of the linear predictor is j data point is given by eq. (1)

$$f(j) = \alpha_0 + \alpha_1 x_{j1} + \alpha_2 x_{j2} + \cdots + \alpha_m x_{jm} \quad (1)$$

where x_{jq} is q-th instructive variable value for data point j, and α0,α1, α2 , …, αm are coefficients representative of relation result of a certain data variable on output and coefficients which is gathered into a single vector of size m+1. Resulting data variables xj0 (=1), xj1....., xjm are then condensed into a single vector xj with a size of m+1. By applying dot product of two vectors to linear predictor function $f(j) = \alpha x_j$in vector notation, linear predictor function $f(j) = \alpha x_j$is inscribed.$f(j) = \alpha^T x_j + x_j^T \alpha$ is matrix representation, where and xj are (m+1) by -1 column vectors. RBF's which is used to calculatedesignated changed version of distance to designated fixed point:$\emptyset(x; c) = \emptyset(\| x - c||) = \emptyset\big(\emptyset\big(\sqrt{(x1 - c1)^2 + \cdots \ldots \ldots \ldots + (xk - ck)^2}$. The RBF Gaussian has same functional form as normal distribution. $\emptyset(x; c) = e^{-b(\|x-c||)^2}$An error in the data is represented by the symbol ∈from eq. (2)

$$y_{ut} = a_{utb} + a_{utfb} x_{fb} + \epsilon \quad (2)$$

YouTube and Facebook intercept are given in Eq. (3)- Eq. (5)

$$a_{utbb} = \frac{(\sum y_{ut})(\sum x_{fb}{}^2) - \big((\sum x_{fb})(\sum x_{fb} y_{ut})}{n\big(\sum x_{fb}{}^2 - (\sum x_{fb})^2} \quad (3)$$

$$b_{utfb} = \frac{(n\sum x_{fb} y_{ut}) - (\sum x_{fb})(\sum x_{fb})}{n\big(\sum x_{fb}^2 - \big(\sum x_{fb}^2\big)} \quad (4)$$

$$y_{ut} = \left[\frac{(\sum y_{ut})(\sum x_{fb}{}^2) - \big((\sum x_{fb})(\sum x_{fb} y_{ut})\big]}{n\big(\sum x_{fb}{}^2 - (\sum x_{fb})^2}\right] + \left[\frac{(n\sum x_{fb} y_{ut}) - (\sum x_{fb})(\sum x_{fb})}{n(\sum x_{fb}{}^2 - (\sum x_{fb}{}^2)}\right] + \epsilon \quad (5)$$

The aLiTw and bLiTw are the intercept and coefficient for Likes/Followers/Visits/Downloads of product by people on LinkedIn and Twitter, respectively. yLi show like/download of product on LinkedIn and xTw give like/download on Twitter in from Eq. (6)- Eq. (12)

$$a_{LiTw} = \frac{(\sum y_{Li})(\sum x_{Tw}{}^2) - ((\sum x_{Tw})(\sum x_{Tw} y_{Li})}{n(\sum x_{Tw}{}^2 - (\sum x_{Tw})^2} \quad (6)$$

$$b_{LiTw} = \frac{(n\sum x_{Tw} y_{Li}) - (\sum x_{Tw})(\sum x_{Tw})}{n(\sum x_{Tw}^2 - (\sum x_{Tw}^2)} \quad (7)$$

$$y_{Li} = a_{LiTw} + b_{LiTw} x_{Tw} \quad (8)$$

$$y_{Li} = \left[\frac{(\sum y_{Li})(\sum x_{Tw}{}^2) - ((\sum x_{Tw})(\sum x_{Tw} y_{Li})}{n(\sum x_{Tw}{}^2 - (\sum x_{Tw})^2}\right] + \left[\frac{(n\sum x_{Tw} y_{Li}) - (\sum x_{Tw})(\sum x_{Tw})}{n(\sum x_{Tw}{}^2 - (\sum x_{Tw}{}^2)}\right] + \epsilon \quad (9)$$

$$a_{\lg Pi} = \frac{(\sum y_{Ig})(\sum x_{Pi}{}^2) - \big((\sum x_{Pi})(\sum x_{Pi} y_{lg})}{n(\sum x_{Pi}{}^2 - (\sum x_{Pi})^2} \quad (10)$$

$$b_{lgPi} = \frac{(n\sum x_{Pi} y_{Ig}) - (\sum x_{Pi})(\sum x_{Pi})}{n(\sum x_{Pi}{}^2 - (\sum x_{PV}{}^2)} \quad (11)$$

$$y_{Ig} = \frac{(\sum y_{lg})(\sum x_{Pi}{}^2) - \big((\sum x_{Pi})(\sum x_{Pi} y_{Ig})}{n(\sum x_{Pi}{}^2 - (\sum x_{Pi})^2} + \frac{(n\sum x_{Pi} y_{Ig}) - (\sum x_{Pi})(\sum x_{Pi})}{n(\sum x_{Pi}{}^2 - (\sum x_{Pi}{}^2)} + \epsilon \quad (12)$$

Ignore missing rows in dataset to eliminate missing data. With use of regression, noisy data is detached. A python formula is used to delete duplicate records from the dataset. To train and test our model, we used pre-processing techniques to create high-quality data.

FFT-CNN based feature extraction

Using FFT, this function does an n-dimensional DFT across any number of axes in an m-dimensional array. DFT in multi-dimensions was applied is defined by Eq. (13):

$$A_{kl} = \sum_{\ell_1=0}^{m-1} \sum_{\ell_2=0}^{n-1} a_{\ell_1 \ell_2} e^{-2\pi i \left(\frac{\epsilon_1}{m} + \frac{\zeta_2}{n}\right)} \tag{13}$$

where image is m n pixels in size. Spatial convolution and max-pooling comparison algorithms utilised in this paper are based on Keras and Theano implementations. Network designs presented in the supplemental to demonstrate the FCNN's capacity to build all of the fundamental CNN layers in Fourier domain.

Hadamard product of image patch with kernel is used to compute the convolution by Eq. (14):

$$\mathbf{z}_{k_1,k_2}^{i,j} = \sum_{\ell_1=[-m_n/2]}^{[m_N/2]} \sum_{\ell_2=[-n_N/2]}^{\lfloor n_N/2 \rfloor} \kappa_{\ell_1,\ell_2}^{i} \odot \mathbf{u}_{k_1-\ell_1,k_2-\ell_2}^{j} \tag{14}$$

which results in an $(m_{\mathbf{u}} - m_{\kappa}) \times (n_{\mathbf{u}} - n_{\kappa})$ mage z since image is usually re-sized to evade including boundary artefacts in evaluation.

Complex conjugate of z ∈ C is represented by z. Re(z) for real and Im(z) for imaginary part of z ∈ C. Euclidean inner product of $x, y \in c^d$is $x, y\rangle := \sum_{i=1}^{d} x_i \bar{y}_i$ with related norm$||x| := \sqrt{\langle x, x\rangle}$Denote identity matrix by $E \in \mathbb{R}^{d\times d}$ For matrix$M \in \mathbb{R}^{d\times d}, M_{i,j}$Supremum norm of matrix$M \in \mathbb{R}^{d\times d}$ is demarcated as $|M|_{\infty} := \sup_{i,j} \left|M_{i,j}\right|$ and supremum norm of tensor$T \in \mathbb{R}^{d\times d\times d}$ is $f: \mathbb{R}^d \to \mathbb{C}$ given by Eq. (15)

$$\| f \|_{\infty} := \inf\{\alpha > 0 || f(x) | \leq \alpha \text{ for a.e. } {}^3x \in \mathbb{R}^d\} < \infty \tag{15}$$

For $f, g \in L^2\left(\mathbb{R}^d\right)$ we set $\langle f, g\rangle := \int_{\mathbb{R}^d} f(x)\overline{g(x)}\mathrm{d}x$ The tensor product of functions

$f, g: \mathbb{R}^d \to \mathbb{C}$is $(f \otimes g)(x, y) := f(x)g(y), (x, y) \in \mathbb{R}^d \times \mathbb{R}^d$Id $: L^p\left(\mathbb{R}^d\right) \to L^p\left(\mathbb{R}^d\right)$

$A: L^p\left(\mathbb{R}^d\right) \to L^q\left(\mathbb{R}^d\right)$ is $\| A \|_{p,q} :=$ $\sup_{\|f\|_F=1} \| Af \|_q$

$f \in L^1\left(\mathbb{R}^d\right)$ by $\hat{f}(\omega) := \int_{\mathbb{R}^d} f(x) e^{-2\pi i\langle x,\omega\rangle}\mathrm{d}x$

$g \in L^1\left(\mathbb{R}^d\right)$ is $(f * g)(y) := \int_{\mathbb{R}^d} f(x)g(y - x)\mathrm{d}x$
$(T_t f)(x) := f(x - t), t \in \mathbb{R}^d$
$(M_{\omega} f)(x) := e^{2\pi i\langle x,\omega\rangle} f(x), \omega \in \mathbb{R}^d$
$(If)(x) := \overline{f(-x)}$
$L_R^2\left(\mathbb{R}^d\right) := \left\{f \in L^2\left(\mathbb{R}^d\right) \mid \text{supp } (\hat{f}) \subseteq B_R(0)\right\}$
$\mathcal{Q}, \left(L^2\left(\mathbb{R}^d\right)\right)^{\mathcal{Q}}$ denotes the space of sets$s := \left\{s_q\right\}_{q\in\mathcal{Q}}, s_q \in L^2\left(\mathbb{R}^d\right)$, for all $q \in \mathcal{Q}$, satisfying $\| |s| | := \left(\sum_{q\in\mathcal{Q}} \|s_q\|_2^2\right)^{1/2} < \infty$

At each point (k_1, k_2), there are $m_k n_k$operations essential and so $(m_{\mathbf{u}} - m_{\kappa+1})(n_{\mathbf{u}} - n_{\kappa+1}) m_k n_k$operations are required for a single convolution. To substitute sliding window method with FTby utilizing discrete analogue of convolution method from Eq. (16)- Eq. (21)

$$\mathcal{F}(\kappa * \mathbf{u}) = \mathcal{F}(\kappa) \odot \mathcal{F}(\mathbf{u}) \tag{16}$$

$$\mathcal{F}(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx \quad (x) \tag{17}$$

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega x} d\omega \tag{18}$$

$$F(\omega) = \begin{cases} 1 & |\omega| \le \omega_c \\ 0 & |\omega| > \omega_c \end{cases} \tag{19}$$

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega x} d\omega \tag{20}$$

$$\frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{i\omega x} d\omega = \frac{1}{2\pi} \frac{e^{i\omega x}}{ix} \Big|_{\omega=-\omega_c}^{\omega_c} = \frac{1}{\pi x} \frac{e^{i\omega_c x} - e^{-i\omega_c x}}{2i} = \frac{\sin\ \omega_c x}{\pi x} = \frac{\omega_c}{\pi} \sin\ c\left(\frac{\omega_c}{\pi} x\right) \tag{21}$$

where F represents two-dimensional discrete FT given by Eq. (22) and Eq. (23)

$$\tilde{\mathbf{u}}_{i_1,i_2} = \sum_{j_1=1}^{m_u} \sum_{j_2=1}^{n_\mathbf{u}} e^{-2i\pi \left(\frac{i_1 j_1 n_\mathbf{u} + i_2 j_2 m_\mathbf{u}}{m_\mathbf{u} n_\mathbf{u}} \right)} \mathbf{u}_{j_1,j_2} \tag{22}$$

$$\Phi_W(f) := \cup_{n=0}^{\infty} \Phi_W^n(f) \tag{23}$$

where$\Phi_W^0(f) := \{f * \psi_{(-J,0)}\}$ is shown by Eq. (24)

$$\Phi_W^n(f) := \{| \cdots | |f * \psi_{\lambda^{(1)}}| * \psi_{\lambda^{(2)}}| \cdots * \psi_{\lambda^{(n)}}| * \psi_{(-J,0)}\}_{\lambda^{(1)},\ldots,\lambda^{(n)} \in \Lambda_{\mathrm{Dw}} \setminus \{(-J,0)\}} \tag{24}$$

for all n ∈ N. Here, index set$\Lambda_{\mathrm{DW}} := \{(-J,0)\} \cup \{(j,k) \mid j \in \mathbb{Z}$ with $j > -J, k \in \{0, \ldots, K-1\}\}$comprises pairs of scales j and directions k and from Eq. (25)

$$\psi_\lambda(x) := 2^{dj} \psi\left(2^j r_k^{-1} x\right), \quad \lambda = (j,k) \in \Lambda_{\mathrm{DW}} \setminus \{(-J,0)\} \tag{25}$$

$\{\psi_\lambda\}_{\lambda \in \Lambda_{\mathrm{Dw}}}$ is taken to form a semi-discrete Parseval frame$\Psi_{\Lambda_{\mathrm{Dw}}} := \{T_b I \psi_\lambda\}_{b \in \mathbb{R}^d, \lambda \in \Lambda_{\mathrm{DW}}}$ for $L^2\left(\mathbb{R}^d\right)$ in Eq. (26)

$$\sum_{\lambda \in \Lambda_{\mathrm{Dw}}} \int_{\mathbb{R}^d} |\langle f, T_b I \psi_\lambda \rangle|^2 \mathrm{d}b = \sum_{\lambda \in \Lambda_{\mathrm{Dw}}} \| f * \psi_\lambda \|_2^2 = \| f \|_2^2, \quad \forall f \in L^2\left(\mathbb{R}^d\right) \tag{26}$$

$\langle f, T_b I \psi_\lambda \rangle = (f * \psi_\lambda)(b), (\lambda, b) \in \Lambda_{DW} \times \mathbb{R}^d$, are the underlying frame coefficients.
The function$|f * \psi_\lambda|, \lambda \in \Lambda_{DW} \setminus \{(-J,0)\}$are thought of as representing locations of singularities of$f f \in L^2\left(\mathbb{R}^d\right)$ given by Eq. (27)

$$|f * \psi_\lambda| = \left|f * \psi_{(j,k)}\right|, \lambda = (j,k) \in \Lambda_{DW} \setminus \{(-J,0)\} \tag{27}$$

the feature extractorΦ_W is translation-invariant in sense by Eq. (28)

$$\lim_{J \to \infty} \| \Phi_W(T_t f) - \Phi_W(f) \| = 0, \quad \forall f \in L^2\left(\mathbb{R}^d\right), \forall t \in \mathbb{R}^d$$
$$(F_\tau f)(x) := f(x - \tau(x))$$
$$\| \Phi_W(F_\tau f) - \Phi_W(f) \| \le C\left(2^{-J} \| \tau \|_\infty + J \| D\tau \|_\infty + \| D^2 \tau \|_\infty\right) \| f \|_{H_W} \tag{28}$$

the Lipschitz property $\|M_n f - M_n h\|_2 \le L_n \| f - h \|_2$ for all$f, h \in L^2\left(\mathbb{R}^d\right)$with $M_n f = 0\ M_n\left(f * g_{\lambda_n}\right)$, i , is then pooled according to Eq. (29)

$$f \mapsto S_n^{d/2} P_n(f)(S_n \cdot) \tag{29}$$

Where$S_n \geq 1$is the pooling factor and$P_n: L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$ satisfies the Lipschitz property$\|P_n f - P_n h\|_2 \leq R_n \| f - h \|_2$, for all $f, h \in L^2(\mathbb{R}^d)$, with $P_n f = 0$ for $f = 0$in discrete-time is demarcated by Eq. (30) and Eq. (31)

$$f_{\mathrm{d}} \mapsto h_{\mathrm{d}} := f_{\mathrm{d}}[S\,\cdot] \tag{30}$$

$$\hat{h}_{\mathrm{d}}(\theta) := \sum_{k\in\mathbb{Z}} h_{\mathrm{d}}[k] e^{-2\pi i k\theta} = \frac{1}{S}\sum_{k=0}^{S-1} \hat{f}_{\mathrm{d}}\left(\frac{\theta-k}{S}\right) \tag{31}$$

As a result, we use the dilation operation to simulate the discrete-time subsampling operation in continuous time given by Eq. (32)

$$f \mapsto h := S^{d/2} f(S\,\cdot), \quad f \in L^2(\mathbb{R}^d) \tag{32}$$

$$\begin{gathered} f_{\mathrm{d}} \mapsto h_{\mathrm{d}} := (f_{\mathrm{d}} * \phi_{\mathrm{d}})[S\,\cdot] \\ \tilde{\kappa}^i = \mathcal{F}(\kappa^i), i = 1, \ldots, N^{\kappa} \\ \tilde{\mathbf{u}}^i = \mathcal{F}(\mathbf{u}^i), i = 1, \ldots, N^{\mathbf{u}} \\ \tilde{\mathbf{z}}^{i,j} = \tilde{\kappa}^i \odot \tilde{\mathbf{u}}^j, i = 1, \ldots, N^{\kappa}, j = 1, \ldots, m^{\mathbf{u}} \\ \mathbf{z}^{i,j} = \mathcal{F}^{-1}(\tilde{\mathbf{z}}^{i,j}), i = 1, \ldots, N^{\kappa}, j = 1, \ldots, N^{\mathbf{u}} \end{gathered} \tag{33}$$

Given a complex three-dimensional tensor with X, Y, and Z dimensions, as well as an arbitrary pool size variable indicating how much data want to save. For x ∈ X, given by Eq. (34):

$$\begin{gathered} x_{y\min} = \left(0.5 - \frac{\text{pool size}}{2}\right) \times Y, \quad x_{y\max} = \left(0.5 + \frac{\text{pool size}}{2}\right) \times Y \\ x_{z\min} = \left(0.5 - \frac{\text{pool size}}{2}\right) \times Z, \quad x_{z\max} = \left(0.5 + \frac{\text{pool size}}{2}\right) \times Z \\ f \mapsto S_n^{d/2}(f * \phi)(S_n\,\cdot), \quad f \in L^2(\mathbb{R}^d) \end{gathered} \tag{34}$$

with the averaging window $\phi \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$.For $n \in \mathbb{N}$, let $\Psi_n = \{T_b I g_{\lambda_n}\}_{b\in\mathbb{R}^d, \lambda_n \in \Lambda_n}$ be a semi-discrete frame for$L^2(\mathbb{R}^d)$ and let$M_n: L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$and $P_n: L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$. Then, triplets' sequence is given by Eq. (35)- Eq. (36)

$$\Omega := \left((\Psi_n, M_n, P_n)\right)_{n\in\mathbb{N}} \tag{35}$$

$$U_n: \Lambda_n \times L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$$

$$U_n(\lambda_n, f) := U_n[\lambda_n] f := S_n^{d/2} P_n\left(M_n(f * g_{\lambda_n})\right)(S_n). \tag{36}$$

For $1 \leq n < \infty$, describe set $\Lambda_1^n := \Lambda_1 \times \Lambda_2 \times \cdots \times \Lambda_n$. An ordered sequence $q = (\lambda_1, \lambda_2, \ldots, \lambda_n) \in \Lambda_1^n$ is known as path. For empty path $e := \emptyset$ we set $\Lambda_1^0 := \{e\}$ and $U_0[e] f := f$. for all $f \in L^2(\mathbb{R}^d)$.The operator U_n is well-defined, i.e., $U_n[\lambda_n] f \in L^2(\mathbb{R}^d)$, for all $(\lambda_n, f) \in \Lambda_n \times L^2(\mathbb{R}^d)$,as shown by Eq. (37) - Eq. (40)

$$\begin{aligned} \|U_n[\lambda_n] f\|_2^2 &= S_n^d \int_{\mathbb{R}^d} \left|P_n\left(M_n(f * g_{\lambda_n})\right)(S_n x)\right|^2 \mathrm{d}x = \int_{\mathbb{R}^d} \left|P_n\left(M_n(f * g_{\lambda_n})\right)(y)\right|^2 \mathrm{d}y \\ &= \left\|P_n\left(M_n(f * g_{\lambda_n})\right)\right\|_2^2 \leq R_n^2 \|M_n(f * g_{\lambda_n})\|_2^2 \\ &\leq L_n^2 R_n^2 \|f * g_{\lambda_n}\|_2^2 \leq B_n L_n^2 R_n^2 \| f \|_2^2 \end{aligned} \tag{37}$$

$$\|f * g_{\lambda_n}\|_2^2 \le \sum_{\lambda_n \in \Lambda_n} \|f * g_{\lambda_n}\|_2^2 \le B_n \| f \|_2^2,$$
$$U[q]f = U[(\lambda_1, \lambda_2, \dots, \lambda_n)]f := U_n[\lambda_n] \cdots U_2[\lambda_2] U_1[\lambda_1] f$$
$$\| U[q]f \|_2^2 \le \left(\prod_{k=1}^{n} B_k L_k^2 R_k^2\right) \| f \|_2^2, \quad \forall q \in \Lambda_1^n, \forall f \in L^2(\mathbb{R}^d) \tag{38}$$

Let $\left|\Omega = \left((\Psi_n, M_n, P_n)\right)_{n\in\mathbb{N}}\right|$ be a module-sequence

$$\Phi_\Omega(f) := \bigcup_{n=0}^{\infty} \Phi_\Omega^n(f) \tag{39}$$

$\Phi_\Omega^n(f) := \{(U[q]f) * \chi_n\}_{q\in\Lambda_i}$, for all$n \in \mathbb{N}$. $\Phi_\Omega: L^2(\mathbb{R}^d) \to \left(L^2(\mathbb{R}^d)\right)^{\mathcal{Q}}$, with $\mathcal{Q} :=$
$\bigcup_{n=0}^{\infty} \Lambda_1^n$,$\Phi_\Omega(f) \in \left(L^2(\mathbb{R}^d)\right)^{\mathcal{Q}}$,for all $f \in L^2(\mathbb{R}^d)$.
Let $\Omega = \left((\Psi_n, M_n, P_n)\right)_{n\in\mathbb{N}}$ Ψ_n by $B_n > 0$ M_n and P_n by $L_n > 0$ and $R_n > 0$,
$\max\{B_n, B_n L_n^2 R_n^2\} \le 1$, $\forall n \in \mathbb{N}$,
$\Phi_\Omega: L^2(\mathbb{R}^d) \to \left(L^2(\mathbb{R}^d)\right)^{\mathcal{Q}}$
$\Phi_\Omega(f) \in \left(L^2(\mathbb{R}^d)\right)^{\mathcal{Q}}$, for all $f \in L^2(\mathbb{R}^d)$ (40)

Algorithm of FFT-CNN:

Input: w, x, m, n
Output: a
$sa \leftarrow max(m, n)$;
$si \leftarrow min(m, n)$;
$k \leftarrow \lceil sa/si \rceil$;
partition w *into* k *vectors,* $w_1, \dots, w_k$;
if $m > n$ *then*
  for $i \leftarrow 0$ *until* k *do*
  $a \leftarrow a + ifft\,(fft(\boldsymbol{w}_i) \circ fft(x))$;
  end
else
  partition x *into* k *vectors,* $x_1, \dots, x_k$;
  for $i \leftarrow 0$ *until* k *do*
  end
end
return a;
$sa \leftarrow max(m, n)$;
$si \leftarrow min(m, n)$;
$k \leftarrow \lceil sa/si \rceil$;
partition w *into* k *vectors,* $w_1, \dots, w_k$;
partition $\frac{\partial L}{\partial w}$ *into* k *vectors,* $\frac{\partial L}{\partial w_1}, \dots, \frac{\partial L}{\partial w_k}$;
if $m > n$ *then*
  partition $\frac{\partial L}{\partial a}$ *into* k *vectors,* $\frac{\partial L}{\partial a_1}, \dots, \frac{\partial L}{\partial a_k}$;
  for $i \leftarrow 0$ *until* k *do*
  $\frac{\partial L}{\partial w_i} \leftarrow ifft\left(fft\left(\frac{\partial L}{\partial a}\right) \circ fft(\boldsymbol{x}')\right) \cdot \mathbf{1}$;

$\frac{\partial L}{\partial x} \leftarrow \frac{\partial L}{\partial x} + ifft\left(fft\left(\frac{\partial L}{\partial a}\right) \circ fft(\boldsymbol{w}_1')\right)$;

end

Else

partition x *into* k *vectors,* $x_1, \ldots, x_k$;

partition $\frac{\partial L}{\partial x}$ *into* k *vectors,* $\frac{\partial L}{\partial x_1}, \ldots, \frac{\partial L}{\partial x_k}$;

for $i \leftarrow 0$ *until* k *do*

$\frac{\partial L}{\partial w_i} \leftarrow ifft\left(fft\left(\frac{\partial L}{\partial a}\right) \circ fft(\boldsymbol{x}_i')\right) \cdot \mathbf{1}$;

$\frac{\partial L}{\partial x_i} \leftarrow ifft\left(fft\left(\frac{\partial L}{\partial a}\right) \circ fft(\boldsymbol{w}_i')\right)$;

end

end

return $\frac{\partial L}{\partial w}$ *and* $\frac{\partial L}{\partial x}$

TL-SVM based classification:

The standard machine learning paradigm is as follows: suppose you're given a collection of iid pairs (training data).
$(x_1, y_1), \ldots, (x_\ell, y_\ell), x_i \in X, y_i \in \{-1, +1\}, P(x, y) = P(y \mid x)P(x)$, function y = f (x, α∗) that minimises likelihood of inaccurate classifications in a given collection of indicator functions f (x, α). Each vector xi ∈ X in this model describes an example produced by an unknown producer P (x) of random vectors xi, and yi∈ {−1, +1} is its categorization described by unknown conditional probability P (y | x). To put it another way, goal is to discover function that minimises risk given by Eq. (41)

$$R(\alpha) = \frac{1}{2}\int |y - f(x, \alpha)| dP(x, y) \tag{41}$$

Given a set of iid triplets, LUPI paradigm defines a more complicated model as shown in Eq. (42)

$$(x_1, x_1^*, y_1), \ldots, (x_\ell, x_\ell^*, y_\ell), x_i \in X, x_i^* \in X^*, y_i \in \{-1, +1\}$$

$$l(\mathbf{x}; \mathbf{w}, b) = \Psi\left(\sum_{i=1}^{N} w_i \phi_i(\mathbf{x}) + b\right) = \Psi(\Phi(\mathbf{x})^\top \mathbf{w} + b) \tag{42}$$

$$\Psi(\mathbf{x}) = \int_{-\infty}^{x} \mathcal{N}(t \mid 0,1) dt \tag{43}$$

Where Ψ(x) is normal CD N (0, 1). Let D be a domain made up of samples xi with xi ∈ X D, i∈ [1, . . ., N], where N is sample number and X D ⊆ F is a D-dimensional feature space. Feature space X D sampled from F is referred to by the abbreviation FX. Using a data matrix X selected from X D, we also investigate a marginal probability distribution P(X). D = F, P(X) is formal definition. In general, if two domains X and Z differ, they either have distinct feature spaces FZ or they have different feature spaces FZ. Heterogeneous Transfer is defined as 6= FX, whereas Homogeneous Transfer is defined as various marginal distributions P(X) 6= P(Z). To obtain optimal prediction models in each of these scenarios, transfer learning is required. K is a kernel matrix that is approximated is achieved by Eq. (44):

$$\tilde{\mathbf{K}} = \mathbf{K}_{N,M}\left(\mathbf{K}_{M,M}\right)^{-1}\mathbf{K}_{M,N} \tag{44}$$

KN,M is a sub-matrix of K that contains all N rows and all M landmark columns. Corresponding kernel matrices $\mathbf{K}_Z$ by taking $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ training data sampled from p(Z) in training domain Z and $= \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ M test data taken from p(X) in test domain X. Mathematical depiction of SVM is expressed as below. Equation of line is outlined as given by Eq. (45),

$$a_1 = a_2 x + b \tag{45}$$

In Eq. (46) 'x' stands for the line's slobe and 'b' stands for intersect,

$$a_1 - a_2 x + b = 0 \tag{46}$$

Let $a' = (a_1, a_2)^T$ and $z' = (x, -1)$ Thus, the above equation can be written by Eq. (47)

$$z'.a' += 0 \tag{47}$$

Eq. (48) is obtained from two-dimensional vectors. Above equation is also relevant for any number of dimensions. Eq. (48) is also known as hyper lane equation. Vector direction $a' = (a_1, a_2)^T$ is mentioned in the form of z' and explained as

$$z' = \frac{a_1}{\|a\|} + \frac{a_2}{\|a\|} \tag{48}$$

Where from Eq. (49)

$$\| a \| = \sqrt{a_1^2 + a_2^2 + a_3^2 + \cdots . a_n^2} \tag{49}$$

as we know that from Eq. (50)

$$\cos(\theta_1) = \frac{a_1}{\|a\|} \; and \; \cos(\theta_2) = \frac{a_2}{\|a\|} \tag{50}$$

So, Eq. (51) expressed as

$$z' = (\cos(\theta_1), \cos(\theta_2))$$
$$z'.a = \| z \| \| a \| \cos(\theta)$$
$$\theta = \theta_1 - \theta_2$$
$$\cos(\theta) = \cos(\theta_1 - \theta_2) = \cos(\theta_1)\cos(\theta_2) + \sin(\theta_1)\sin(\theta_2)$$
$$= \frac{z'_1}{\| z' \|}\frac{a_1}{\| a \|} + \frac{z'_2}{\| z' \|}\frac{a_2}{\| a \|}$$
$$\frac{z'_1 a_1 + z'_2 a_2}{\| z' \| \| a \|}$$
$$z'.a' = \sum_{i=1}^{n} z'_i a_i \tag{51}$$

The dot product of atop equation is calculated for n-dimensional vectors as - consider f = y(z · a + b) when sign (f) > 0 refers to categorization that is correct and sign (f) < 0 refers to categorization that is incorrect. If D is the provided dataset, then f is calculated on a training dataset as shown in Eq. (52)

$$f_i = y_i(z'.a + b) \tag{52}$$

Calculate functional margin (F) of a dataset as given in Eq. (53),

$$F = min_{i=1\ldots m} f_i \tag{53}$$

Through the contrast between the hyperplanes, hyperplane that has largest F are chosen in which F is called dataset geometric mean. Optimal values of z and b for selection of optimal hyperplane should be found. Lagrangian function is given below in Eq. (54) - Eq. (55),

$$L(z', b, \alpha) = \frac{1}{2} z'.z' - \sum_{i=1}^{m} \alpha_i [y: (z'.a + b) - 1] \quad (54)$$

$$\nabla_b L(z', b, \alpha) = -\sum_{i=1}^{m} \alpha_i y_i = 0 \quad (55)$$

By using Eqs. (56) and (57) we

$$z' = \sum_{i=1}^{m} \alpha_i y_i a_i \ \ and \ \sum_{i=1}^{m} \alpha_i y_i = 0 \quad (56)$$

After replacement of Lagrangian function L is given by eq (57)

$$z'(\alpha, b) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j a_i a_j \quad (57)$$

Thus by Eq. (58),

$$max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j a_i a_j \quad (58)$$

If point is over hyperplane is categorized as +1 class mean HD found, and if point is under hyperplane, it is categorized as −1 class mean HD does not found.
the function ˆf(x) are taken as a PDF by Eq. (59):

$$y = f(x) \sim \mathcal{P}\big(\mu(x), k(x, x')\big) \quad (59)$$

where μ : X → R is covariance function that specifies relationship between response values, y, and distance between input values, x, x 0 as given by Eq. (60)- Eq. (62)

$$\mu_t(x) = \mu(x) + k(x)^{\mathrm{T}}(K + \sigma^2 I)^{-1}(y - \mu)$$
$$\sigma_t^2(x) = k(x, x) + \sigma^2 I - k(x)^{\mathrm{T}}(K + \sigma^2 I)^{-1} k(x) \quad (60)$$

where $\boldsymbol{k}(\boldsymbol{x})^{\mathrm{T}} = [k(\boldsymbol{x}, \boldsymbol{x}_1) \quad k(\boldsymbol{x}, \boldsymbol{x}_2) \quad \dots \quad k(\boldsymbol{x}, \boldsymbol{x}_t)]$

$$K: = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_t) \\ \vdots & \ddots & \vdots \\ k(x_t, x_1) & \dots & k(x_t, x_t) \end{bmatrix} \quad (61)$$

$$\mathcal{C}(\mathcal{D}_n, \mathcal{D}_t) = c_x \cdot |\mathcal{D}_s| + c_t \cdot |\mathcal{D}_t| + \mathcal{C}_{Tr}(|\mathcal{D}_s| \cdot |\mathcal{D}_t|),$$
$$\mathcal{C}(\mathcal{D}_s, \mathcal{D}_t) \leq \mathcal{C}_{\max}, \quad (62)$$

To capture relationship, define following kernel function as shown by Eq. (63):

$$k(f, g, \boldsymbol{x}, \boldsymbol{x}') = k_i(f, g) \times k_{xx}(\boldsymbol{x}, \boldsymbol{x}') \quad (63)$$

Matched features from first course to comparable features from second course one by one to produce a common collection of features for each pair of courses. When a matching feature not located, a new feature with zero values was fashioned for each occurrence. The C1 course, for example, included features connected to resources with seven pages, but the C2 course had features linked to resources with six pages. Finally, because a new empty feature was constructed and added in C2 course dataset, the new {C1, C2} pair of datasets comprised seven features relating page resources, matching the seventh feature of C1 course. Training process for two supporting deep networks is referred known as the second phase. The first was trained on new source course Ci to extract its updated weights, while second was trained on new target course Cj to compute the baseline evaluation. The accuracy measure, which corresponds to the percentage of properly categorised

examples, was calculated in both situations while models were trained for 150 epochs. In addition, the 10-fold cross validation resampling technique was used to assess DNN overall performance. The transfer learning approach was implemented in third phase, which was most important. Target course's deep model was built from scratch, but network weights were initialised using source course's previously determined weights. Pre-trained model was fine-tuned by running it for a specific number of epochs (hence referred to as Ci,j) each time: zero (starting point) 10, 20, 30, 40, 50, 100, and 150.

Algorithm 2 provides pseudocode of proposed transfer learning with SVM method.

Input: $c1, c2$, *scores* $=$ [[]#$c1$ *is source course dataset and* $c2$ *is target*

Output: *scores*

1. *(c1', c2')* ← *commonRepresentation (c1, c2)*
2. *model1* ← *createAndCompileModel ()*
3. *weights* ← *fitModel (model1,* $c1'$*, epochs=150)*
4. *model2* ← *createAndCompileModel(weights)*
5. *for each* e *in* [0,10,20,30,40,50,100,150] *do*
6. *model2'* ← *fitModel (model2,* $c2'$*, epochs* $= e$ *)*
7. *score* ← *evaluate (model2',* $c2'$*, folds=10)*
8. *add (score, scores)*
9. *end for each*
10.
11. *# Hypothesis a common feature depiction for two sequences*
12. *function common Representation (dataset1, dataset2)*
13. *dataset1'* $=$ [[]], *dataset2'* $=$ []
14. *# Match features of dataset1 with features of dataset2*
15. *# Create new features when necessary*
16. *for each* t *in ['forum', 'page', 'recourse', 'folder', 'url', 'assign views', 'assign'] do*
17. *features1* ← *getFeaturesOf Type (dataset* $1, t$)
18. *features2* ← *getFeaturesOf Type (dataset 2,* t)
19. *size* ← *min (features1.size, features2.size)*
20. *diff* ← *absoluteDifference (features1.size, features2.size)*
21. *for i=0 to size-1 do*
22. *add (features1 [i] , dataset1')*
23. *add (features2[i], dataset2')*
24. *end for*
25. *for j=0 to diff-1 do*
26. *if f1*← *get Features At (features*$1, features1.size + j)do$*# if f1 exists*
27. *add (f1,dataset1`)*
28. *add (createEmptyFeatures(),dataset2`)*
29. *else f2*← *get Features At (features*$1, features1.size + j)do$
30. *add (f2,dataset2`)*

```
31. add (create Empty Feature(),dataset1')
```

4. Performance Analysis:

The entire implementation of the proposed FFT-CNN_TL-SVM is done in Python tool and configurations considered for simulation are: PC with Ubuntu, 4GB RAM, and Intel i3 processor.

4.1 Database description

Tweets Set: Tweepy [13], a Python module for accessing the Twitter API, is used to gather tweets. To compile a list of tweets from all over the country as a result of this. It takes many parameters as input, such as coordinates, radius, and so on, and records the tweet-ids, text and location of most recent ones in database after removing duplicates, links, hashtags, and terms in other languages from these tweets. For example, the Delhi Tweets Set we produced comprises around 10,000 entries. Another method we used Tweepy was to pass it a twitter-username (of a user) as an input and have it saved all of that user's tweets in our database (to date).

B. Emotion-Words Set:(EWS) One of the most important and necessary parts of emotional quantification of a sentence is the right selection of relevant and commonly-used Emotion words. We built the Emotion-Words Set (EWS), a high-quality, accurate bag of words with roughly 1500 words. After that, each of these words was manually assigned to one of three Intensity-Categories such as Strong, Medium, Light. It was created by recursively searching (Depth First Search) in a thesaurus up to two levels for synonyms of 6 main Emotion-Categories such as Happiness, Sadness, Anger, Surprise, Fear, Disgust.

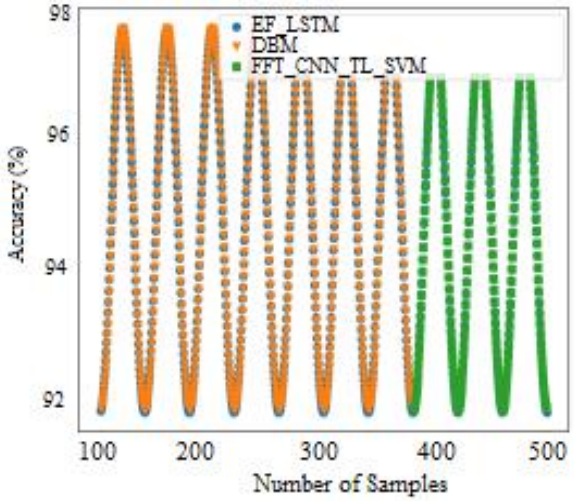
C. Degree-Words Set: Degree-Words set is a group of roughly 50 words utilized to increase or decrease intensity of emotions in a statement. For example, the terms "too cheerful" and "hardly happy" have nearly opposite connotations. Each word in this collection has a Degree-Intensity connected with it, with H denoting High, N denoting Negation and L denoting Low. Examples include words like, “too”, “more” (H); “hardly” (N); and “nearly” (L), etc

Table-1 *Comparative analysis of multimodal sentiment and emotion classification-based parameters*

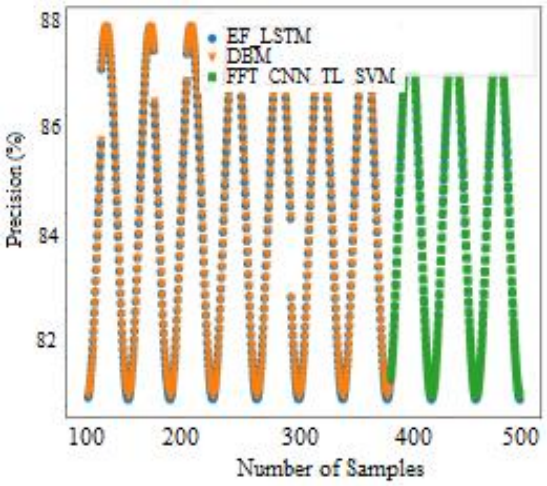
Dataset	Techniques	Accuracy	Precision	Recall	F-1 score	MSE
Tweets Dataset	EF-LSTM	96.8	85.6	78.5	76.5	48
	DBM	97.2	87	79.6	77.8	46
	FFT-CNN_TL-SVM	98	88	80	78	45
Emotion-Words Dataset	EF-LSTM	95.8	85.9	76.8	76	45.2
	DBM	96.5	86.5	77.4	76.4	45.7
	FFT-CNN_TL-SVM	97	87.2	78.5	77.2	46
Degree-Words Dataset	EF-LSTM	94.8	84.2	75	75	43
	DBM	95.2	84.6	77	76	44
	FFT-CNN_TL-SVM	96	85	81	78	45

The above table-1represents comparative analysis between existing and proposed methods in terms of accuracy, precision, recall, F-1 score and mean square error in classification of online sentimental and emotion analysis for multimodal datasets. Here the datasets compared are Tweets dataset,

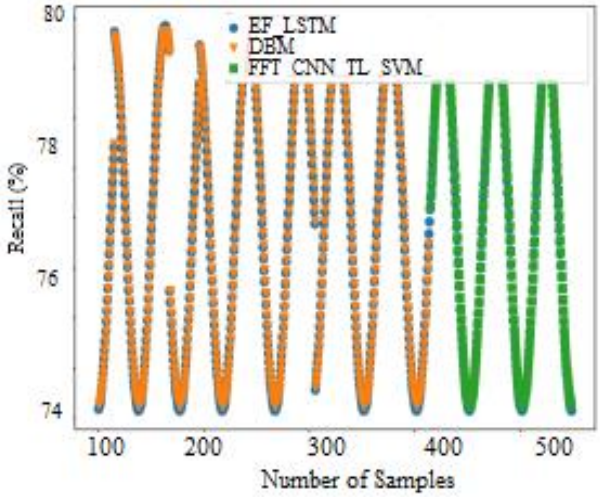
emotion-words dataset and degree words dataset and the existing techniques compared are EF-LSTM *and DBM.*



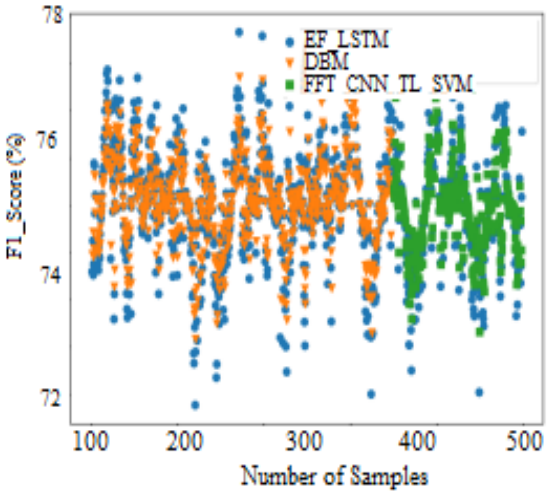
(a) Accuracy



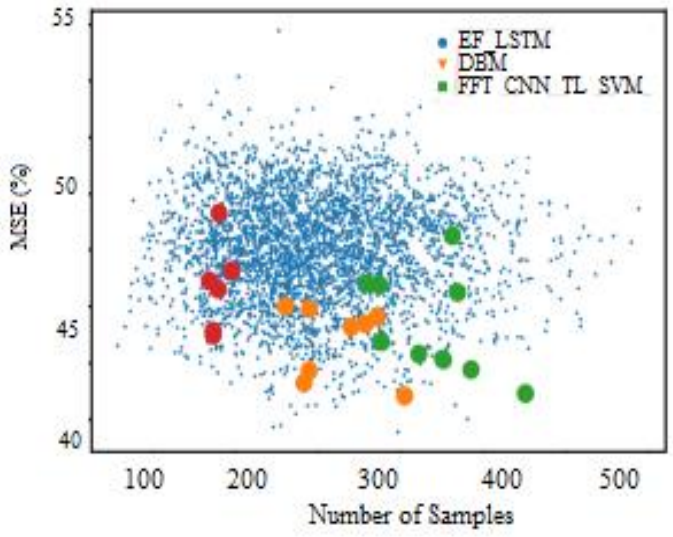
(b) Precision



(c) Recall

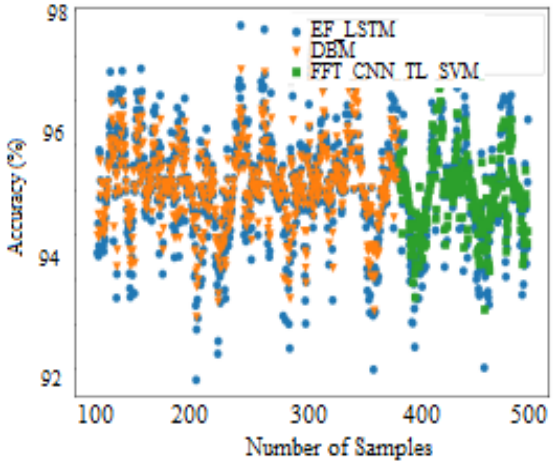


(d) F1- score

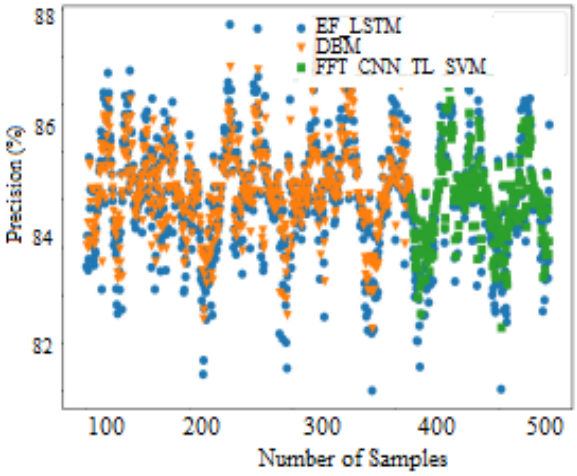


(e) MSE

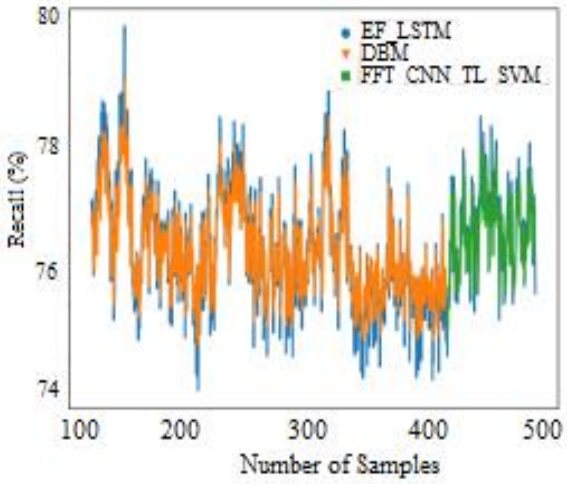
Figure-2 *Comparative analysis of multimodal sentimental and emotion classification for Tweets dataset in terms of (a) Accuracy, (b) Precision, (c) recall, (d) F-1 score, (f) MSE*



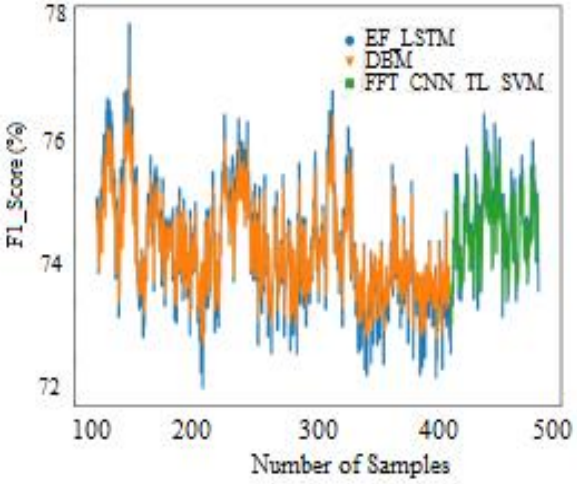
(a) Accuracy



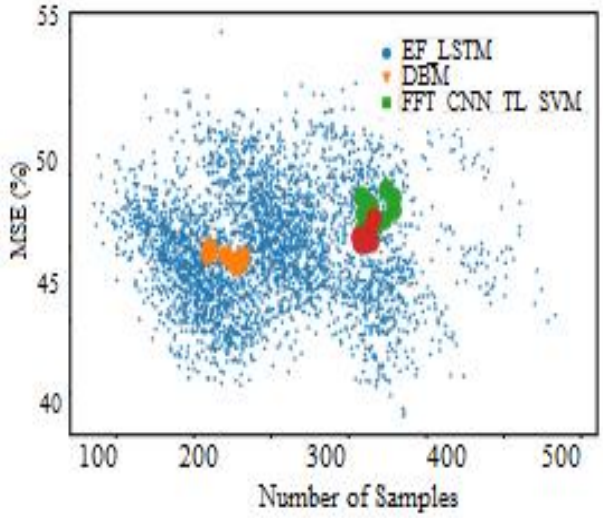
(b) Precision



(c) Recall

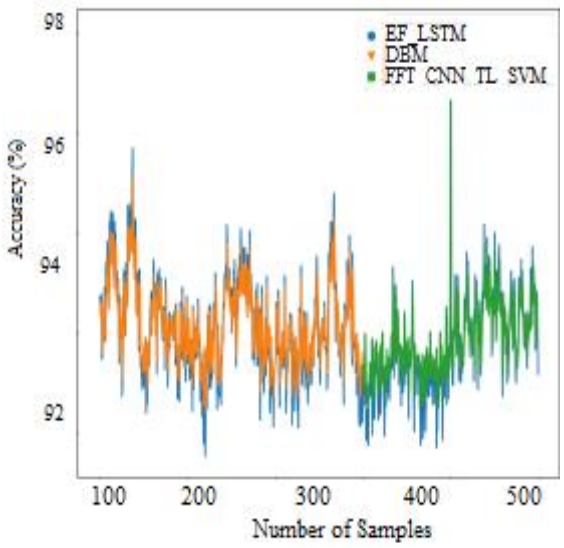


(d) F-1 score

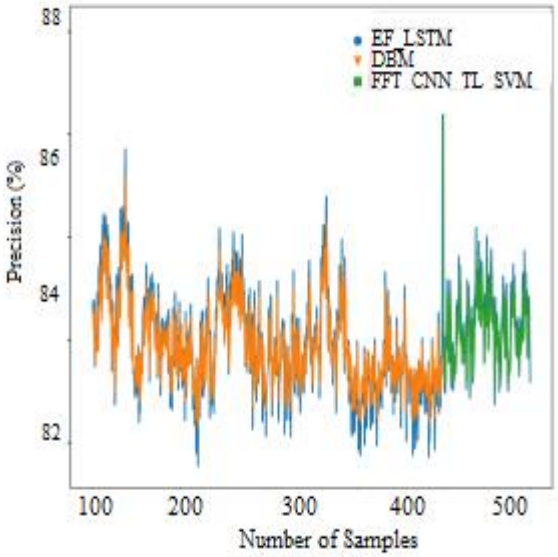


(e) MSE

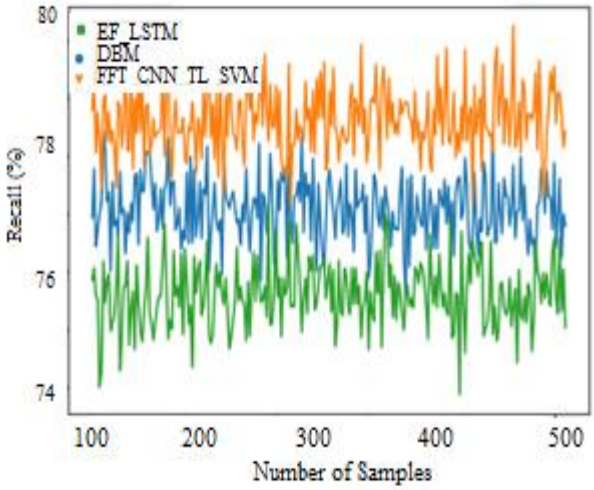
Figure-3 *Comparative analysis of multimodal sentimental and emotion classification for emotion-words dataset in terms of (a) Accuracy, (b) Precision, (c) recall, (d) F-1 score, (f) MSE*



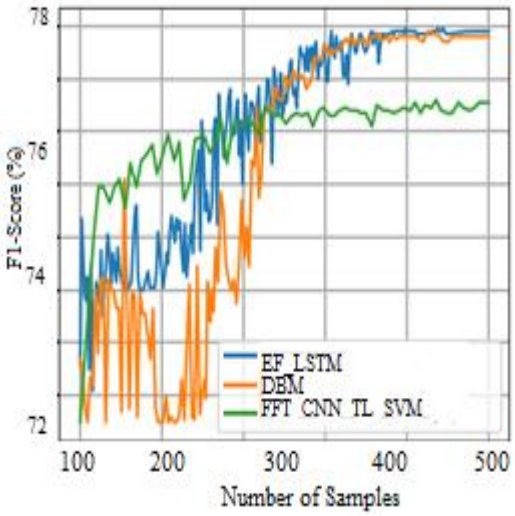
(a) Accuracy



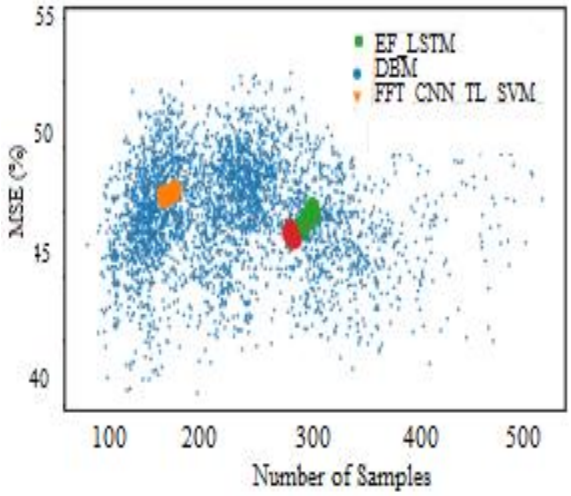
(b) Precision



(c) Recall



(d) F-1 Score



(e) MSE

Figure-4 *Comparative analysis of multimodal sentimental and emotion classification for degree words dataset in terms of (a) Accuracy, (b) Precision, (c) recall, (d) F-1 score, (f) MSE*

The above figure 2-4 shows the comparison of multimodal sentimental and emotion classification for various datasets like Tweets dataset, emotion-words dataset and degree words dataset. The comparative analysis has been carried out in terms of (a) Accuracy, (b) Precision, (c) Recall, (d) F-1 score and (e) MSE. Existing techniques compared based on the classification performance for sentimental and emotional dataset. Based on this comparison, the accuracy obtained for Tweets dataset is 98%, emotion-words dataset is 97% and degree words dataset is 96% by FFT-CNN_TL-SVM technique; precision obtained for Tweets dataset is 88%, emotion-words dataset is 87.2% and degree words dataset is 85% by FFT-CNN_TL-SVM technique; recall achieved by FFT-CNN_TL-SVM for Tweets dataset is 80%, emotion-words dataset is 78.5% and degree words dataset is 81%; for Tweets dataset is 78%, emotion-words dataset is 77.2% and degree words dataset is 78% attained F 1-score value by FFT-CNN_TL-SVM technique; MSE based on classification attained by FFT-CNN_TL-SVM technique for Tweets dataset is 45%, emotion-words dataset is 46% and degree words dataset is 45%. From the above-mentioned results obtained by proposed technique, it has achieved optimal results when compared with proposed technique.

5. Conclusion

This paper proposed the novel technique in cognitive computing based multimodal sentimental and emotion classification using machine learning technique. Here the aim is to collect the cognitive computed network-based data of social media, Facebook, and twitter. The input data has been processed and segmented to reduce noise and remove the unwanted numerical data which includes data cleaning. Then the features of this data have been extracted based on FFT (fast Fourier transform) to change the cognitive data frequency range with convolutional neural network (FFT-CNN). Then to classify extracted features using transfer learning based SVM (TL-SVM). The experimental results have been carried out based on accuracy, precision, recall, F-1 score for various dataset and from this comparative analysis the proposed technique obtained optimal results in multimodal sentimental and emotion classification through cognitive computing.

References:

[1] Jamshidi, P., Velez, M., Kästner, C., Siegmund, N., &Kawthekar, P. (2017, May). Transfer learning for improving model predictions in highly configurable software. In 2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS) (pp. 31-41). IEEE.

[2] Tsiakmaki, M., Kostopoulos, G., Kotsiantis, S., &Ragos, O. (2020). Transfer learning from deep neural networks for predicting student performance. Applied Sciences, 10(6), 2145.

[3] S. Li and W. Deng, “Deep facial expression recognition: a survey,” IEEE Transactions on Affective Computing, 2020.

[4] Y. Li, J. Zeng, S. Shan, and X. Chen, “Occlusion aware facial expression recognition using CNN with attention mechanism,” IEEE Transactions on Image Processing, vol. 28, pp. 2439–2450, 2018.

[5] Raab, C., &Schleif, F. M. (2018, June). Transfer learning for the probabilistic classification vector machine. In Conformal and Probabilistic Prediction and Applications (pp. 187-200). PMLR.

[6] R. Tanabe and H. Ishibuchi, “An easy-to-use real-world multiobjective optimization problem suite,” Applied Soft Computing, vol. 89, article 106078, 2020.

[7] M. J. Mayer, A. Szilágyi, and G. Gróf, “Environmental and economic multi-objective optimization of a household level hybrid renewable energy system by genetic algorithm,” Applied Energy, vol. 269, article 115058, 2020.

[8] J. Zhang, Y. Huang, Y. Wang, and G. Ma, “Multi-objective optimization of concrete mixture proportions using machine learning and metaheuristic algorithms,” Construction and Building Materials, vol. 253, article 119208, 2020.

[9] S, R. D., Shyamala, L. ., & Saraswathi, S. . (2022). Adaptive Learning Based Whale Optimization and Convolutional Neural Network Algorithm for Distributed Denial of

Service Attack Detection in Software Defined Network Environment. International Journal on Recent and Innovation Trends in Computing and Communication, 10(6), 80–93.

[10] M. Abdel-Basset, R. Mohamed, and S. Mirjalili, "A novel whale optimization algorithm integrated with Nelder-Mead simplex for multi-objective optimization problems," Knowledge-Based Systems, vol. 212, p. 106619, 2021

[11] M. D. Ding and L. Li, "CNN and HOG dual-path feature fusion for face expression recognition," Information and Control, vol. 49, no. 1, pp. 47–54, 2020.

[12] L. Q. Lan and X. Zhang, "Facial expression recognition method based on a joint normalization strategy," Journal of Beijing University of Aeronautics and Astronautics, vol. 46, no. 9, pp. 1797–1806, 2020.

[13] Stojanovic, N. . (2022). Smart Grid based Wireless Communication in 5G Network for Monitoring and Control Systems in Renewable Energy Management. International Journal on Future Revolution in Computer Science &Amp; Communication Engineering, 8(3), 23–32.

[14] A. A. Muhammad and J. K. Muhammad, "EEG-based multi-modal emotion recognition using bag of deep features: an optimal feature selection approach," Sensors, vol. 19, no. 23, 2019.

[15] Wiatowski, T., &Bölcskei, H. (2017). A mathematical theory of deep convolutional neural networks for feature extraction. IEEE Transactions on Information Theory, 64(3), 1845-1866.

[16] W. Rahman, M. K. Hasan, and A. Zadeh, "M-BERT: injecting multimodal information in the BERT structure," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 2359–2369, 2020.

[17] Pise, P. . (2022). Mobile Cloud IoT for Resource Allocation with Scheduling in Device-Device Communication and Optimization based on 5G Networks. International Journal on Future Revolution in Computer Science &Amp; Communication Engineering, 8(3), 33–42.

[18] X. Chen, B. Wu, and P. Sheng, "A multiobjective evolutionary algorithm based on surrogate individual selection mechanism," Personal and Ubiquitous Computing, vol. 23, no. 3-4, pp. 421–434, 2019.

[19] Boyapati, B. ., & Kumar, J. . (2022). Parasitic Element Based Frequency Reconfigurable Antenna with Dual Wideband Characteristics for Wireless Applications. International Journal on Recent and Innovation Trends in Computing and Communication, 10(6), 10–23.

[20] Y. Yu and Y. J. Kim, "Attention-LSTM-attention model for speech emotion recognition and analysis of IEMOCAP database," Electronics, vol. 9, no. 5, p. 713, 2020.

[21] Huang, Y.; Yang, J.; Liu, S.; Pan, J. Combining Facial Expressions and Electroencephalography to Enhance Emotion Recognition. Future Internet 2019, 11, 105

[22] Wu, X.; Zheng, W.-L.; Lu, B.-L. Investigating EEG-Based Functional Connectivity Patterns for Multimodal Emotion Recognition. arXiv 2020, arXiv:2004.01973.

[23] Chunawale, A.; Bedekar, D. Human Emotion Recognition using Physiological Signals: A Survey. In Proceedings of the 2nd International Conference on Communication & Information Processing (ICCIP), Tokyo, Japan, 20 April 2020.

[24] Patil, P. ., Waghole, D. D. ., Deshpande, D. V. ., & Karykarte, D. M. . (2022). Sectoring Method for Improving Various QoS Parameters of Wireless Sensor Networks to Improve Lifespan of the Network. International Journal on Recent and Innovation Trends in Computing and Communication, 10(6), 37–43.

[25] Liao, J.; Zhong, Q.; Zhu, Y.; Cai, D. Multimodal Physiological Signal Emotion Recognition Based on Convolutional Recurrent Neural Network. IOP Conf. Ser. Mater. Sci. Eng. 2020, 782, 032005.

[26] Pathak, G. ., & Angurala, M. . (2022). Multipath Routing in Cloud Computing using Fuzzy based Multi-Objective Optimization System in Autonomous Networks. International Journal on Future Revolution in Computer Science & Communication Engineering, 8(3), 43–53.

[27] Rahimi, A.; Kanerva, P.; Benini, L.; Rabaey, J.M. Efficient Biosignal Processing Using Hyperdimensional Computing: Network Templates for Combined Learning and Classification of ExG Signals. Proc. IEEE 2019, 107, 123–143.