Knowledge-based Data Processing for Multilingual Natural Language Analysis

DEEPAK KUMAR JAIN, Key Laboratory of Intelligent Control and Optimization for Industrial Equipment of Ministry of Education, Dalian University of Technology, Dalian, School of Artificial Intelligence, Dalian, China, and Symbiosis Institute of Technology, Symbiosis International University, Pune, India
YAMILA GARCÍA-MARTÍNEZ EYRE, Universidad Internacional de La Rioja (UNIR), Avda. de la Paz, Logroño (La Rioja). España
AKSHI KUMAR, Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, United Kingdom
BRIJ B. GUPTA, International CenterforAI and Cyber Security Research and Innovations, & Department of Computer Science and Information Engineering, Asia University, Taichung 413, Taiwan, Symbiosis Centre for Information Technology, Symbiosis International University, Pune, India, & Lebanese American University, Beirut, 1102, Lebanon, and Center for Interdisciplinary Research at University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India, and Department of Computer science, Dar Alhekma University, Jeddah, Saudi Arabia, & Immersive Virtual Reality Research Group, King Abdulaziz University, Jeddah, 21589, Saudi Arabia
KETAN KOTECHA, Symbiosis Centre for Applied Artificial Intelligence, Symbiosis International University, Pune, India and School of Mathematical Sciences, Sunway University, Malaysia

Natural Language Processing (NLP) aids the empowerment of intelligent machines by enhancing human language understanding for linguistic-based human-computer communication. Recent developments in processing power, as well as the availability of large volumes of linguistic data, have enhanced the demand for data-driven methods for automatic semantic analysis. This paper proposes multilingual data processing using feature extraction with classification using deep learning architectures. Here, the input text data has been collected based on various languages and processed to remove missing values and null values. The processed

BB Gupta is now with International Center for AI and Cyber Security Research and Innovations, & Department of Computer Science and Information Engineering, Asia University, Taichung 413, Taiwan, Symbiosis Centre for Information Technology, Symbiosis International University, Pune, India, & Lebanese American University, Beirut, 1102, Lebanon, and Center for Interdisciplinary Research at University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India, and Department of Computer science, Dar Alhekma University, Jeddah, Saudi Arabia.
Authors' addresses: D. K. Jain, Key Laboratory of Intelligent Control and Optimization for Industrial Equipment of Ministry of Education, Dalian University of Technology, Dalian, School of Artificial Intelligence, Dalian, China, and Symbiosis Institute of Technology, Symbiosis International University, Pune, India, China; Y. García-Martínez Eyre, Universidad Internacional de La Rioja (UNIR), Avda. de la Paz, Logroño (La Rioja). España; A. Kumar (corresponding author), Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, United Kingdom; email: akshi.kumar@mmu.ac.uk; B. B. Gupta (corresponding author), Department of Computer Science and Information Engineering, Asia University, Taichung 413, Taiwan; K. Kotecha, Symbiosis Centre for Applied Artificial Intelligence, Symbiosis International University, Pune, India and School of Mathematical Sciences, Sunway University, Malaysia.

2375-4699/2024/05-ART63
https://doi.org/10.1145/3583686

data has been extracted using Histogram Equalization based Global Local Entropy (HEGLE) and classified using Kernel-based Radial basis Function (Ker_Rad_BF). These architectures could be utilized to process natural language. We present solutions to the multilingual sentiment analysis issue in this research article by implementing algorithms, and we compare precision factors to discover the optimum option for multilingual sentiment analysis. For the HASOC dataset, the proposed HEGLE_ Ker_Rad_BF achieved an accuracy of 98%, a precision of 97%, a recall of 90.5%, an f-1 score of 85%, RMSE of 55.6%, and a loss curve analysis attained 44%. For the TRAC dataset, the accuracy of 98%, the precision attained is 97%, the Recall is 91%, the F-1 score is 87%, and the RMSE of the proposed neural network is 55%.

CCS Concepts: • **Information systems** → **Social networks** • **Computing methodologies** → **Natural language processing**; **Neural networks**;

Additional Key Words and Phrases: Sentiment analysis, deep learning, multilingual data processing, Histogram Equalization based Global Local Entropy, Kernel-based Radial basis Function

ACM Reference format:
Deepak Kumar Jain, Yamila García-Martínez Eyre, Akshi Kumar, Brij B. Gupta, and Ketan Kotecha. 2024. Knowledge-based Data Processing for Multilingual Natural Language Analysis. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 23, 5, Article 63 (May 2024), 16 pages.
https://doi.org/10.1145/3583686

1 INTRODUCTION

NLP is a rapidly expanding field of study aimed at teaching computers to understand human language in general. This is a tough goal to fulfil because the computer must comprehend many parts of language, such as syntax and semantics, as well as deal with a range of input formats, such as raw text, texts, and speech. Thanks to improved computational capabilities, a better knowledge of the techniques, and more annotated data, numerous advances in NLP have recently been made. A typical NLP assignment entails annotating a large amount of text and then using supervised ML techniques to solve it. If we want to do **part-of-speech (POS)** tagging, for example, we could annotate every word in a sentence with the appropriate POS tags, such as Noun, Verb, or Adjective, and then train a statistical classifier [1, 2]. Annotated data is critical in this method because it is the model's only source of guidance. Annotated resources have long been a staple of NLP conferences, recognizing the relevance of annotated data. Furthermore, the **Language Resources and Evaluation Conference (LREC)** is an important language resources conference.

Modern speech and language processing research has made substantial use of ML techniques. Pertinent studies report recent trends in deep learning-enabled NLP [3]. Transfer learning too has emerged as an intriguing method. Transfer learning, in general, refers to any strategy that uses auxiliary resources to improve model learning for the target task [4]. Because human speech as well as languages are so diverse and unbalanced, this is critical for speech and language studies. There are more than 5,000 languages spoken around the world, with dialects adding to the total 389 languages (almost 6% of the total) accounting for 94% of the world's population, whereas hundreds of languages are spoken by only a few people. Only a few of the 389 'major' languages have adequate speech and language research resources. Perhaps only English falls within the category of 'rich resource' languages. Furthermore, even for English, resources in various domains are grossly uneven. Data scarcity is a problem in practically all speech and language research. More significantly, because human language is so dynamic that new words, as well as domains, develop every day, no model learned at one moment will always be true. It's nearly impossible for speech as well as language researchers to learn a method from a single data source as well as then put it on the shelf since there's so much diversity, variation, imbalance, and dynamics. We'll need to turn to smarter methods that can learn from various languages, data, and topics while also adapting the model [5].

To evaluate human sentiments, natural language is considered as the vital source [6] and examination of sentiment is described through the limited issues in the **natural language processing (NLP)** that offer information related to the expression of human, languages and articulated language emotions through textual or non-textual information [6]. The earlier examination related to sentiment analysis highly concentrated on the performance of sentiment analysis through English [7] and with the utilization of multimodal sentimental analysis [8]. Recently, the researches were subjected to poor management of resource and multilingual languages need to be evaluated.

The contribution of this paper has been given below:

- To propose multilingual data processing using feature extraction with classification using deep learning architectures
- To extract the features of input text data using Histogram Equalization based Global Local Entropy (HEGLE)
- To classify extracted deep features using Kernel-based Radial basis Function (Ker_Rad_BF)
- We compare accuracy factors to discover the best answer for multilingual sentiment analysis by using these research solutions to multilingual sentiment analysis problems by implementing algorithms.

2 RELATED WORKS

Even when combined with relatively simple classifiers, we present a contextual word embedding process for classification. Despite neglecting the grammatical ordering of words, **deep averaging networks (DAN)** [9] can perform as well as or better than considerably more advanced approaches. Instead, paragraph-Vec [10] uses an approach similar to CBOW [11] to include such ordering as well as contextual data of paragraphs, resulting in better results than earlier methods. The most prevalent RNN variant is **long short-term memory networks (LSTMs)**, which address gradient disappearing or exploding concerns that normal RNN architectures have [12]. Many TC techniques based on are developed, and we discuss a few of them below. The authors of [13] presented an LSTM structure for the computation of sequential structure trains for the representation in the perfect fit phrase. In [14] they combine the latent topics in RNNS that are long-range dependencies for the improvement in baselines [15]. [16] presented the advantages of integrated invariance quality position in CNN network through **gated recurrent units (GRU)**. [17] improves the feature extraction based RNNs capabilities. The examination is performed for the consideration of different datasets to achieve outcomes better than the current state of the art on various benchmark datasets. Improvements to the architectures of CNN-based models [18] have been made. The authors of [19] present a novel CNN method that makes two changes to Kim-CNN architecture. To collect additional fine-grained information from various document areas, a dynamic max-pooling approach is used first. Improvements to the architectures of CNN methods [18] have been made. The authors of [20] present a novel CNN method that makes two changes to Kim-CNN architecture. To collect additional fine-grained information from various areas of the document, a dynamic max-pooling approach is used first. For TC, character-level CNNs are investigated. [21] proposes one of the earliest such models. As shown in Figure 6, the model receives fixed-size characters encoded as one-hot vectors as input and runs them through a DCNN method with six convolutional layers with pooling operations, as well as three fully connected layers. [22] describes a method for encoding text with CNNs that significantly decreases the amount of memory and training time. This method scales well with alphabet size, allowing more data from the original text to be preserved, resulting in improved classification performance. A neural language method that learns distributed representations for words was proposed in [23]. The authors claimed that by combining the words that are represented in sentences with the computation of

joint probability in the sequences of words, they were able to generate an exponential number of semantically adjacent phrases. The work of [24] demonstrated the value of pre-trained word embeddings. They devised an NN method that is at the heart of many contemporary methods. In addition, the research established through the NLP based application with the embedding process. However, the **continuous bag-of-words (CBOW)** uses the bi-gram classifier model to estimate the vector distribution in a high-quality manner for the embedded words. The composite vector comprises of the two-words vector for the semantic separate word, e.g., 'man' + 'royal' = 'king'. In [25] the authors presented the phenomenon for the composition evaluation for the assumption of met in the constraints words that are distributed evenly in the space for embedding. Additionally, Glove [8] evaluate the embedding process measured with a count-based approach for the normalization in the log operation and the co-occurrence matrix is processed. That factor effectively provides the representation in the low-dimensional term with the minimization of reconstruction loss.

3 SYSTEM MODEL

The present section provides the multilingual data processing model with the feature extraction and classification model with DL technique. The data provided as the input text are collected and processed with the different language data with the removal of the null values. With the adoption of **Histogram Equalization based Global Local Entropy (HEGLE)** and **Kernel-based Radial basis Function (Ker_Rad_BF),** features are extracted in the data. Figure 1 illustrates the overall architecture of the proposed model.

3.1 Data Pre-processing

The first is real-world data from a variety of real-world experiments. The second type is synthetic data, which is created artificially to mirror real-world trends. Instead of using genuine data, synthetic data is manufactured. These datasets are particularly useful in situations where the volume of data required is substantially more than what is currently available, or where privacy is critical and stringent, such as in the healthcare industry. Toy datasets are the third category, which is used for presentation and visualization. They are usually created artificially; there is often no requirement to depict real-world data patterns.

The first stage in the pre-processing procedure will be to convert our tweets to lowercase. This prevents duplicate copies of the same words and conceptual language from being created. For example, 'Analytics' and 'analytics' are treated as separate terms for calculating word count. The next step is to eliminate punctuation, which adds no further data to text data. Then to remove the stop words, either a list of stop words can be created manually, or it can be used from predefined libraries. By the pre-processing, we must remove the repeated words, so it must be checked for the 10 most frequently occurring words in the text data, then make a call to remove or retain. So far, the pre-processing has been done for basic data clean-up. The next step is to pre-process the data using NLP techniques. The NLP technique that has been done in pre-processing of data is as follows:

- **Term frequency.** The ratio of the number of words in a sentence to the length of a sentence is known as term frequency. As a result, the term frequency is defined as follows:

$$TF(t) = \frac{Within\ a\ document\ time\ term\ t\ count}{Total\ terms\ count\ within\ the\ document}$$

- **Inverse Document Frequency.** It is computed based on the concept of words that are not useful to those included in each document. This resulted in the IDF of every word based

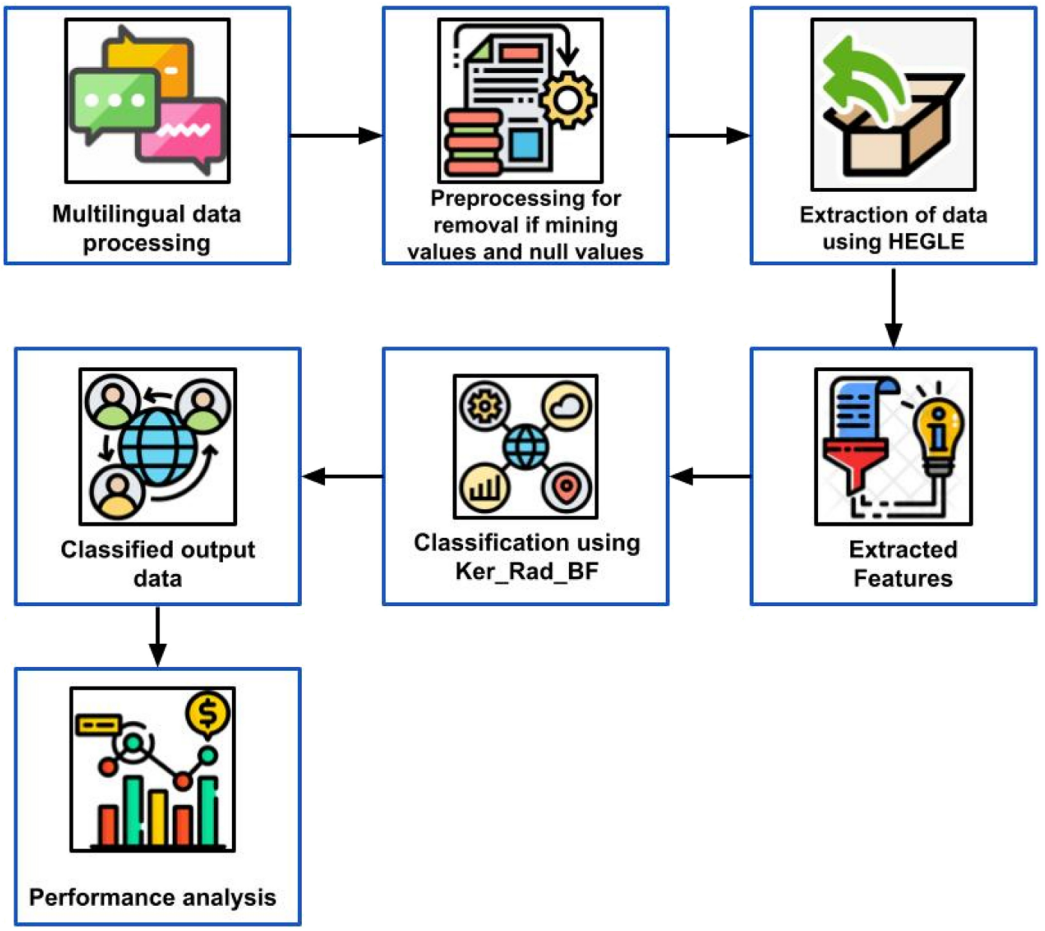

Fig. 1. Overall proposed architecture.

on the ratio log to the total rows count to the rows count that appear in each word.

$$IDF = log(N/n),$$

where N represents the rows count and n denotes the words count in rows.

- **Term Frequency – Inverse Document Frequency (TF-IDF).** The multiplication of TF and IDF that we calculated earlier is TF-IDF. Because terms like 'don't, "can't,' and 'use' are often used, TF-IDF penalizes them. It has, however, given "disappointed" a high weighting because it will be highly valuable in determining the sentiment.
- **Bag of Words.** The term "bag of words" refers to a text representation that defines the presence of words within text data. This is based on the assumption that two similar text fields will include similar types of words as well as have a similar bag of words. Furthermore, it deduces something about the document's significance solely from the text.
- **Word Embeddings.** Representation of text in the form of vectors is known as word embedding. These tokens are used as a starting point for stemming as well as lemmatization (text pre-processing's next stage). Stemming is the process of creating morphological variants of a root/base word. For example, phrases "chocolates," "chocolatey," and "choco" are

minimized to root word "chocolate," and "retrieval," "retrieved," and "retrieves" are minimized to stem "retrieve". Stemming attempts to reach the root word by deleting trailing characters from a word. As a result, it's possible that the root word isn't a dictionary word. The fundamental benefit of stemming is that it reduces the feature space or the number of distinct words in the corpus on which the model can train. Lemmatization is another approach to getting to the underlying word. Unlike stemming, lemmatization employs a dictionary-based method, reducing words to their dictionary roots in most cases. The speed of processing suffers as a result of this. Before segmentation, the text is preprocessed to shrink and denoise the input from the database. When the text is utilized for the feature extraction stage's training and testing phase, it must be resized to change the visualization. In addition, the text must be denoised to remove undesirable noise and smooth the text.

3.2 Data Segmentation

The segmentation of text is stated as the components of the written language that are difficult to split. In case of segmentation in text N-gram classifier is incorporated. The N-gram classifier combines the different words integrated together. The comparison of N-gram classifier with the other classifiers such as bigrams and trigrams requires minimal data volume. The n-gram classifier records the structures of language that are considered as the basic concepts that comprise the likely questions. Based on n-gram classifiers, higher contexts are processed and examined with the determined ideal length for the applications. In case the words are long, the missed out words are considered as the universal knowledge-based capturing instead of the specific factors.

- **GLEHE-based feature extraction.** The proposed GLEHE combines MI for parameter identification with an **FCM (fuzzy c Means)** classifier for segmentation. The FCM technique estimates the center of clusters related to text and segmentation conducts based on the fuzzy membership function. Through the reduction of the distance between the cluster c items FCM combines the data cluster in group Equation (1) and provides the fuzzy cluster dimension determined with the FCM methods. Based on estimation of cluster data point centre the membership functions are assigned to the corresponding centre of each cluster. The centres of clusters are closer to the data that are likely related to the centre of the cluster.

$$F = \{F_1, F_2, \ldots., F_h, \ldots, F_z\} \tag{1}$$

The objective function is denoted as F that is utilized for the clustering process and the process of minimization is represented as in Equation (2)

$$M_\mu = \sum_{i=1}^{q} \sum_{h=1}^{r} d_{ih}^{\mu} \times E_{ih} \tag{2}$$

Where, $E_{ih} = e_i - F_h$. Using Equation (3), the cluster centres are calculated based on the membership function, a membership function comprised within the fuzzy set A. Each element is denoted as X is a value in the range of 0 - 1, and X is described as μA:X $\rightarrow$ [0,1]. The element in X associated degree is related to the set A in fuzzy whose values are stated as the value of membership function or degree.

$$F_h = \frac{\sum_{i=1}^{l} d_{ih}^{\mu} \cdot e_i}{\sum_{i=1}^{l} d_{ih}^{\mu}} \tag{3}$$

The FCM algorithm completes the process until all of the texts have been segmented, at which point the segmented texts are represented as $Q_{u,y}^{FCM}$.

For segmentation, the proposed GLEHE employs **Mutual Information (MI)**. MI is designed to effectively classify text components based on the consideration of particular selection parameters. The segmentation of the proposed GLEHE is investigated using the equations below (4).

$$Q_{u,y} = \begin{cases} Q^{A}_{n,v}; & \text{if } Q^{A}_{u,v} == Q^{FCM}_{n,y} \\ M; & \text{if } Q^{A}_{u,y} \neq Q^{FCM}_{u,v} \end{cases} \tag{4}$$

Where $Q^{A}_{u,v}$ denotes segmentation output of MI model and $Q^{FCM}_{u,v}$ is the segmentation output of the FCM model. The criterion for the processing of the MI condition are stated as M. Then, in the MI criterion M is determined by the candidates which are chosen. The MI of both FCM as well as AC segments are evaluated in order to calculate M. The MI for the text determines how close the text is to other pixels, for the selection of higher MI value in the texts that are highly preferable. The computation of MI with AC's are computed as in Equation (5)

$$M^{A} = MI\left(Q^{A}_{u,y}\right) \tag{5}$$

$MI(Q^{A}_{u,v})$ is the MI of the active contour segments, which can be calculated using two windows W_1, W_2. W_1 has a window size of 3×3 and W_2 has a window size of 4×4. The conventional MI calculation formula is based on Equation (6).

$$MI\left(W_1, W_2\right) = E\left(W_1\right) + E(W_2) - E\left(W_1, W_2\right) \tag{6}$$

$E(W_1)$ denotes the entropy of window W1, while $E(W_2)$ denotes the entropy of windowW_2. The term $E(W_1, W_2)$ refers to the windows W_1, W_2 entropy. Equation (7) represents the expression for the global and local entropy features with joint probability (Equation 8),

$$E(W_1) = -\sum_{u} pw_1(u)\log pw_1(u) \tag{7}$$

$$E\left(W_1, W_2\right) = -\sum_{u,y} p_{w_1,w_2}(u,v)\log pw_1w_2(u,v) \tag{8}$$

The value of conditional probability is stated as $pw_1(u)$ with the FCM based segments to evaluate the MI as presented in Equation (9)

$$M^{FCM} = MI\left(Q^{FCM}_{u,y}\right) \tag{9}$$

The segmentation criterion for M is computed as in Equation (10)

$$M = \begin{cases} Q^{A}_{u,v}; & \text{if } Q^{A}_{u,y} == Q^{FCM}_{u,v} \\ Q^{FCM}_{u,v}; & \text{else} \end{cases} \tag{10}$$

As stated earlier, the segments computed with the hybrid method achieve the higher MI value and the proposed model achieves the MI function as F = F1, F2.

3.3 Classification using Kernel-based Radial basis Function (Ker_Rad_BF)

The RBF method's approximant is stated in a variety of ways. Kansa's, as well as Fasshauer's approximants, are provided as well as discussed for clarity. Consider that xi represents points that belong to a bounded domain's Ω with the point set of χ. The domain boundary ($\partial\Omega$) is partitioned based on the interior point I with the boundary value B. Through time-dependent interpolation the PDEs are computed for the instances to achieve the PDE as stated in Equation (11)

$$\begin{aligned} \frac{\partial f(x)}{\partial t} &= \mathcal{L}\left[f(x)\right] \; x \in \Omega, \\ f(x) &= g(x) \; x \in \partial\Omega, \end{aligned} \tag{11}$$

The purpose of this technique is to use function to approximate a given dataset as given in Equation (12):

$$f(\mathbf{x}) = \sum_{j=1}^{M} c_j \phi(\mathbf{x} - \boldsymbol{\xi}_j) \tag{12}$$

The vector weights are denoted as c = (c1, … , cM) and the minimization of quadratic form is stated as T in Equation (13)

$$\frac{1}{2}\mathbf{c}^T \mathbf{Q}\mathbf{c} \tag{13}$$

where Q is represented as the definite positive symmetric value in M × M dimensions. This quadratic form is minimised under N with the linear constraints value of Ac = h, the ranking process of N × M matrix is presented as in right-hand side h = (h1, …, hN) T is supplied. As a result, the constrained quadratic minimization problem is an LSE in Equation (14):

$$F(\mathbf{c}, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{c}^T \mathbf{Q}\mathbf{c} - \boldsymbol{\lambda}^T (\mathbf{A}\mathbf{c} - \mathbf{h}) \tag{14}$$

$$\frac{\partial F(\mathbf{c}, \boldsymbol{\lambda})}{\partial \mathbf{c}} = \mathbf{Q}\mathbf{c} - \mathbf{A}^T \boldsymbol{\lambda} = 0 \frac{\partial F(\mathbf{c}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \mathbf{A}\mathbf{c} - \mathbf{h} = 0$$
$$\begin{pmatrix} \mathbf{Q} & -\mathbf{A}^T \\ \mathbf{A} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{h} \end{pmatrix} \tag{15}$$

where symmetric matrix is denoted as Qi,j = ϕ(kξi − ξjk) and Q with the computation in Equation (16) as follows:

$$f(\mathbf{x}) = \sum_{j=1}^{M} c_j \phi(r_j) = \sum_{j=1}^{M} c_j \phi(\mathbf{x} - \boldsymbol{\xi}_j) \tag{16}$$

Where MRBFs the sum values are computed with the approximation function f(x), the reference point distance is presented as ξj that are combined and the weighted suitable coefficients are denoted as cj. The dataset for the LSE is computed as in the Equation (17) as follows:

$$h_i = f(\mathbf{x}_i) = \sum_{j=1}^{M} c_j \phi(\mathbf{x}_i - \boldsymbol{\xi}_j) = \sum_{j=1}^{M} c_j \phi_{i,j} \; i = 1, \ldots, N. \tag{17}$$

The linear method in the matrix equation is computed as in Equation (18)

$$\mathbf{A}\mathbf{c} = \mathbf{h} \tag{18}$$

where N, M represents the rows count, and M defines the unknown weights count of [c1, …, cM] T, i.e. the number of reference points Equation (19) is written as follows:

$$\begin{pmatrix} \phi_{1,1} & \cdots & \phi_{1,M} \\ \vdots & \ddots & \vdots \\ \phi_{i,1} & \cdots & \phi_{i,M} \\ \vdots & \ddots & \vdots \\ \phi_{N,1} & \cdots & \phi_{N,M} \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_M \end{pmatrix} = \begin{pmatrix} h_1 \\ \vdots \\ h_i \\ \vdots \\ h_N \end{pmatrix} \tag{19}$$

The polynomial function Pk(x) of degree k is commonly used to extend the RBF approximant. The approximated value can now be stated as follows in Equation (20):

$$f(\mathbf{x}) = \sum_{j=1}^{M} c_j \phi(\mathbf{x} - \boldsymbol{\xi}_j) + P_k(\mathbf{x}) \tag{20}$$

where ξj are user-specified reference points. As a result, the LSE can be solved in Equation (21):

$$h_i = f(\mathbf{x}_i) = \sum_{j=1}^{M} c_j \phi(\mathbf{x}_i - \boldsymbol{\xi}_j) + P_k(\mathbf{x}_i) = \sum_{j=1}^{M} c_j \phi_{i,j} + P_k(\mathbf{x}_i) \quad i = 1, \ldots, N \tag{21}$$

We can write for E 2 using matrix notation Equation (22):

$$\begin{pmatrix} \phi_{1,1} & \cdots & \phi_{1,M} & x_1 & y_1 & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \phi_{i,1} & \cdots & \phi_{i,M} & x_i & y_i & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \phi_{N,1} & \cdots & \phi_{N,M} & x_N & y_N & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_M \\ a_x \\ a_y \\ a_0 \end{pmatrix} = \begin{pmatrix} h_1 \\ \vdots \\ h_i \\ \vdots \\ h_N \end{pmatrix} \tag{22}$$

As shown in the equation, it expresses the approximant (sk (x)) without change in terms of radial basis function (Equation 23):

$$s_k(x) = \sum_j \alpha_j \phi(r(x, x_j), \varepsilon) \tag{23}$$

The estimation is evaluated based on the basic coefficient calculated based on approximant for the interpolations and each bias association coefficient is stated as j. The approximation of location with the same data is accomplished with the denoted data stated as xi for the same data (f(xi)) presented in Equation (24).

$$s_k(x_i) = \sum_j \alpha_j \phi(r(x_i, x_j), \varepsilon) = f(x_i) \tag{24}$$

Equation (25) is a matrix-formatted linear system:

$$\left[\phi|_{\chi\chi}\right][\alpha] = \left[f|_{\chi}\right]. \tag{25}$$

If there are no repeated points in matrix $\phi\ \chi\chi$, it is symmetric and invertible. When solving PDEs, this characteristic is useful because it lowers computational cost. To solve PDE, differential operator approximates at interior locations in Equation (26).

$$\begin{bmatrix} \mathcal{L}\phi|_{I_\chi} \\ \phi|_{B\chi} \end{bmatrix} [\alpha] = \begin{bmatrix} \mathcal{L}f|_I \\ g|_B \end{bmatrix} \tag{26}$$

In Equation (26), the values are approximated with the differential operator to achieve the accurate information that is not utilized for the resolving PDEs for the reduction of computational cost. The analytical Equation (27) provides the coefficient αj with the inverse matrix value of $\phi\ \chi\chi$ as the original solution.

$$\underbrace{\begin{bmatrix} \mathcal{L}\phi|_{I_X} \\ \phi|_{B\chi} \end{bmatrix} \left[\phi|_{\chi X}\right]^{-1}}_{W} \left[f|_{\chi}\right] = \begin{bmatrix} \mathcal{L}f|_I \\ g|_B \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial t}\big|_I \\ f|_B \end{bmatrix} \tag{27}$$

Without explicitly evaluating coefficients, discrete differential operator (W) is created. Before beginning the time iteration, the calculation must be completed once. The differential operator (W) with discrete process is processed at each time stamp with the differential equations stated in Equation (28).

$$s_h(x) = \sum_j^{\forall x_j \in \partial\Omega} \alpha_j \phi(r(x, \xi), \varepsilon)\Big|_{\xi = x_j} + \sum_j^{\forall x_j \in \{\Omega \setminus \partial\Omega\}} \beta_j \mathcal{L}^{\xi} \phi\left(r(x, \xi), \varepsilon\right)\Big|_{\xi = x_j} \tag{28}$$

To find interpolation coefficients, interpolation restrictions should be imposed on approximant, which is performed through approximant at positions xi to f. (xi) in Equation (29).

$$s_h(x_i) = \sum_{j}^{\forall x_j \in \partial\Omega} \alpha_j \mathcal{B}^{\xi} \phi(r(x_i, \xi), \varepsilon)\Big|_{\xi = x_j} + \sum_{j}^{\forall x_j \in \{\Omega \setminus \partial\Omega\}} \beta_j \mathcal{L}^{\xi} \phi(r(x_i, \xi), \varepsilon)\Big|_{\xi = x_j} = f(x_i). \tag{29}$$

It can be represented using matrices as a set of equations Equation (30):

$$\underbrace{\begin{bmatrix} \mathcal{L}^{\xi}\phi\big|_{II} & \phi|_{IB} \\ \mathcal{L}^{\xi}\phi\big|_{BI} & \phi|_{BB} \end{bmatrix}}_{A} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} f|_I \\ f|_B \end{bmatrix} \tag{30}$$

The coefficients αj and βj is stated as the matrix interpolation based on Kansa's method that is not symmetric in interpolation. To minimize PDE, differential operator L must be applied to inner points to generate PDE matrix in Equation (31):

$$\underbrace{\begin{bmatrix} \mathcal{L}\mathcal{L}^{\xi}\phi\big|_{II} & \mathcal{L}\phi|_{IB} \\ \mathcal{L}^{\xi}\phi\big|_{BI} & \phi|_{BB} \end{bmatrix}}_{A_{\mathcal{L}}} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \mathcal{L}f|_I \\ g|_B \end{bmatrix} \tag{31}$$

Solving interpolation system of equations yields coefficients vector values. The differential operator in discrete term (w) is substituted in the Equation (32)

$$\underbrace{[A_{\mathcal{L}}]\,[A]^{-1} \begin{bmatrix} f|_I \\ f|_B \end{bmatrix} = \begin{bmatrix} \mathcal{L}f|_I \\ g|_B \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial t}\big|_I \\ f|_B \end{bmatrix}}_{W} \tag{32}$$

For the Gaussian function, it can be calculated as Equation (33):

$$\phi, H_n(\varepsilon) = \begin{cases} \frac{n! r^n}{(n/2)! R^{n+1}} \sqrt{\pi} (-1)^{n/2} & \text{even } n \\ 0 & \text{odd } n \end{cases}, \tag{33}$$

Equation (34) is swapped into Equation (18) to get the following expression:

$$\frac{\partial a_n}{\partial x_a} = B_n x_\alpha \frac{n r^{n-2} - r^n}{R^{n+3}} \tag{34}$$

Bn = (1 n/2 / (2 n (n/2)!) is a constant that is proportional to n. Those values comprise of the odd terms with zero values. This allows previous Equation (35) to be expressed only in terms of even n values (n = 2k):

$$\frac{\partial^2 a_n}{\partial x_\alpha \partial x_\beta} = x_a x_\beta B_n = B_n x_\alpha \frac{n r^{n-2} - r^n}{R^{n+3}} \tag{35}$$

As a result, n is now deemed even. Below are the first three nonzero values of an Equation (36):

$$a_0 = +\frac{1}{R} \tag{36}$$

It's worth noting that at ε = 1, a0 is related to the multi-quadractic inverse operation $(1/\sqrt{(1+\varepsilon^2 r^2)})$ RBF. Because a0 is dominant term as form parameter approaches zero, we believe this insight contributes to stability of current approach derivatives to be calculated by

Table 1. Specifications of Hyperparameters

Parameter	Value
Rate of Learning	1×10^{-5}
epsilon	1×10^{-8}
Epochs	3
warmup ratio	0.1
Normalized values	1.0
steps	0
gradient accumulation steps	1
max seq. length	120

Table 2. Comparative Analysis of HASOC Dataset

Metrics	LSTM	CNN	HEGLE_ Ker_Rad_BF
Accuracy	95.2	96.9	98
Precision	95	96	97
Recall	87.5	89.5	90.5
F1-score	82	83	85
RMSE	52.1	53.6	55.6
Loss curve	46	45	44

Equation (37):

$$\frac{\partial a_n}{\partial x_\alpha} = \frac{x_\alpha}{r}\left(\frac{\partial a_n}{\partial r}\right)$$
$$\frac{\partial^2 a_n}{\partial x_\alpha \partial x_\beta} = \frac{x_a x_\beta}{r^2}\left(\frac{\partial^2 a_n}{\partial r^2} - \frac{1}{r}\frac{\partial a_n}{\partial r}\right) + \frac{1}{r}\frac{\partial a_n}{\partial r}\delta_{\alpha\beta r} \tag{37}$$

where $\delta\alpha\beta$ represented as the Kronecker delta. The derivatives that are first are considered in space related to the first two consecutive derivatives as r, that offers the results as in Equation (38)

$$\frac{\partial a_n}{\partial r} = B_n \frac{nr^{n-1} - r^{n+1}}{R^{n+3}} \frac{\partial^2 a_n}{\partial r^2} = B_n \frac{2r^{n+2} - (5n+1)\, r^n + n\,(n-1)\, r^{n-2}}{R^{n+5}} \tag{38}$$

To achieve the explicit derivatives formula in the concern space, the Equation (39) is derived as follows:

$$\frac{\partial a_n}{\partial x_a} = B_n x_\alpha \frac{nr^{n-2} - r^n}{R^{n+3}} \frac{\partial^2 a_n}{\partial x_\alpha \partial x_\beta} = x_a x_\beta B_n \left(\frac{3r^n - (6n)\, r^{n-2} + n\,(n-2)\, r^{n-4}}{R^{n+5}}\right) + \delta_{a\beta} B_n \frac{nr^{n-2} - r^n}{R^{n+3}} \tag{39}$$

4 PERFORMANCE ANALYSIS

The developed model is examined with the Nvidia Tesla K80 GPU with the data ration of 0.8:0.2 for splitting utilized for the training and validation set as shown in Table 1. To achieve significant results in validation, manual fine-turning needs to be performed based on the epochs to perform classification. For every language, the learning rate optimal value is represented as 1 105 to derive best value with the epochs count of 3. Therefore, only training data was used to train the models. We stopped early if evaluation loss did not enhance after 10 calculation rounds. Table 2 comprises of the generated hyperparameters to compute the findings. An ‡ denotes the hyperparameters that

Table 3. Comparative Analysis of TRAC Dataset

Parameters	LSTM	CNN	HEGLE_ Ker_Rad_BF
Accuracy	96	97	98
Precision	94.8	95.5	97
Recall	88	89	91
F1-score	83	85	87
RMSE	52	53	55
Loss curve	47	45	43

have been tuned, and optimal values are computed and presented. The values of the remaining hyperparameters are maintained fixed.

4.1 Dataset description

Indian language-based datasets are available in a vast range that offers the standard resources for the language processing. The HASOC dataset comprises of the datasets that are available between the year 2019 – 2021. Similarly, TRAC tasks are computed between the year 2018 – 2020 and Dravidian LangTech comprises of the task performing the offensive detection in language obtained in year 2021. The TRAC dataset performs the tasks that are processed under two iterations for the aggressive identification for the conjunction. In COLING the TRAC 2018 participants comprise of the training and testing dataset based on the Facebook comments for the tweets in Hindi and English. Conventionally, the SVM is implemented in ML to evaluate the performance. The TRAC 2020 model LREC comprises of the comments in YouTube that are in English, Bengali, and Hindi. The dataset comprises of two tasks that are subtask A with the three courses such as TRAC 2018 and two classes in subtask B for the identification of violation in gender through messages. The transformer implemented with pretrained model exhibits the significant performance for the BERT. The assigned Indo-European language HASOC model is assigned which stands for "hate speech and offensive material identification," and is likely the most well-known competition involving Indian languages. It was held at the FIRE in 2019 and 2020. HASOC 2019 offered datasets in English, German, and Hindi, while HASOC 2020 included datasets in all three languages as well as Tamil and Malayalam. Methods based on NN designs are proved to be competitive in terms of performance. With the addition of Marathi, HASOC 2021 is currently under progress.

The comparative analysis of the proposed model with the existing model is comparatively examined with the data processing in multilingual to perform feature extraction and classification with the deep learning model. The parametric analysis with HASOC and TRAC dataset is performed with multilingual data as shown in Table 3. The analysis is performed with the consideration of different datasets such as RMSe, recall, accuracy, precision, loss curve, and F1-score. The performance of proposed HEGLE_ Ker_Rad_BF is comparatively examined with LSTM and CNN. The estimation of parameters based on epochs are computed for training process in the NN. The proposed model HEGLE_ Ker_Rad_BF examination with HASOC dataset achieves the 98% accuracy, recall value of 97%, 85% F1-score, 55.6% of RMSE, and 44% for the loss curve to perform feature-based extraction and classification in web-based text as presented in Figure 2(a)-(f). The conventional LSTM attained 95.2% accuracy, 95% precision, recall as 87.5%, 82% F-1 score, 52.1% RMSE, and loss analysis obtained is 46%; CNN obtained accuracy of 96.9%, precision of 96%, recall of 89.5%, f-1 score of 83%, RMSE of 52.1%, and loss curve analysis attained is 46%.

The examination of the proposed model with the HASOC dataset expressed that the proposed model achieves the optimal results for the multilingual data classification. Secondly, for TRAC

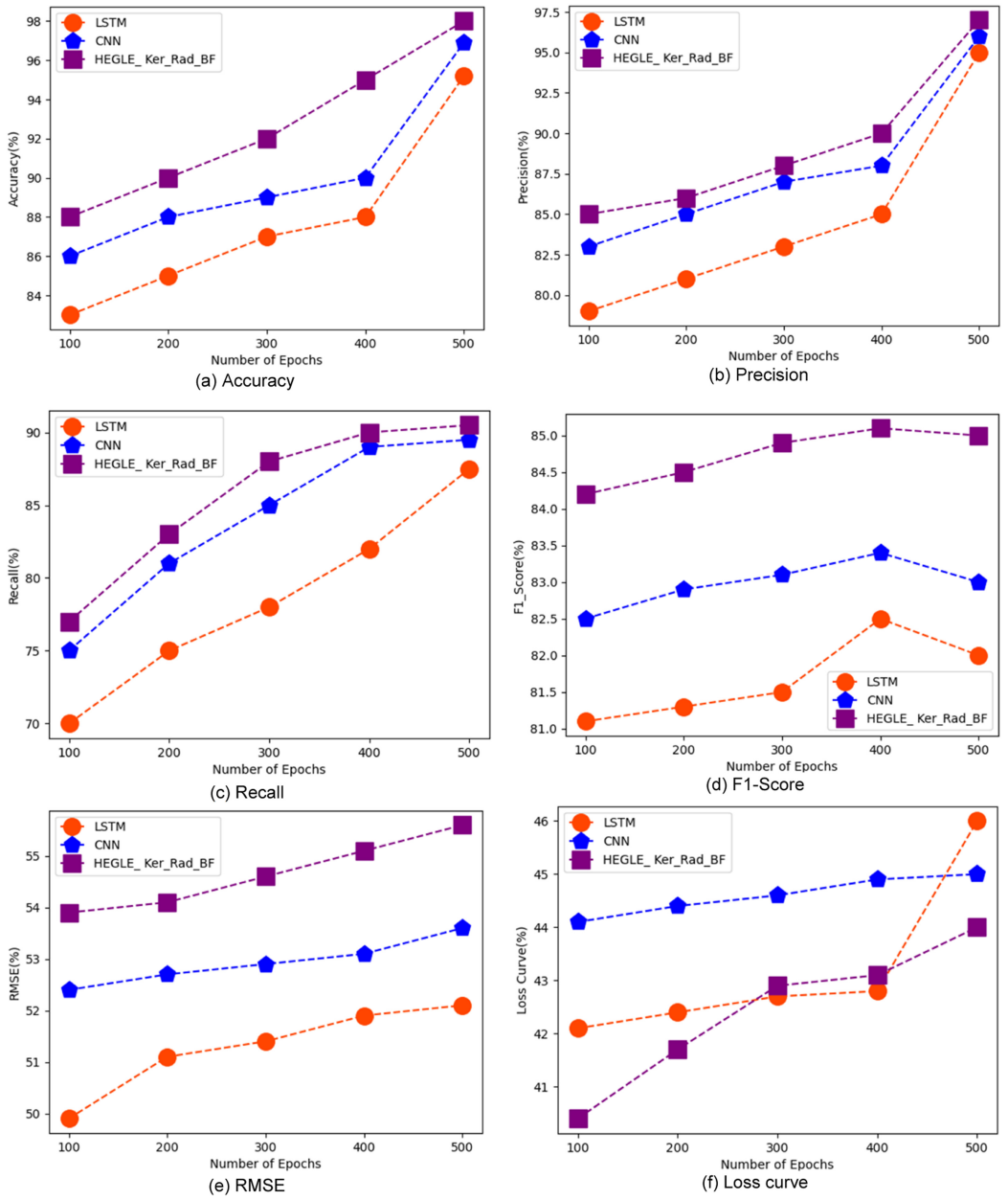

Fig. 2. Comparative analysis for HASOC dataset (a) accuracy, (b) precision, (c) recall, (d) F-1 score, (e) RMSE, and (f) loss curve analysis.

multilingual data analysis performed with the parameters compared with proposed HEGLE_ Ker_Rad_BF acquired accuracy of 98%, precision attained is 97%, Recall is 91%, F-1 score is 87%, and RMSE of the proposed neural network is 55% which also obtained loss curve analysis of 43% as illustrated in Figure 3 (a)-(f). The LSTM existing model attained 96% accuracy, 94.8% precision, recall as 88%, 83% F-1 score, 52% RMSE, and loss analysis obtained is 47%; LSTM obtained 96%

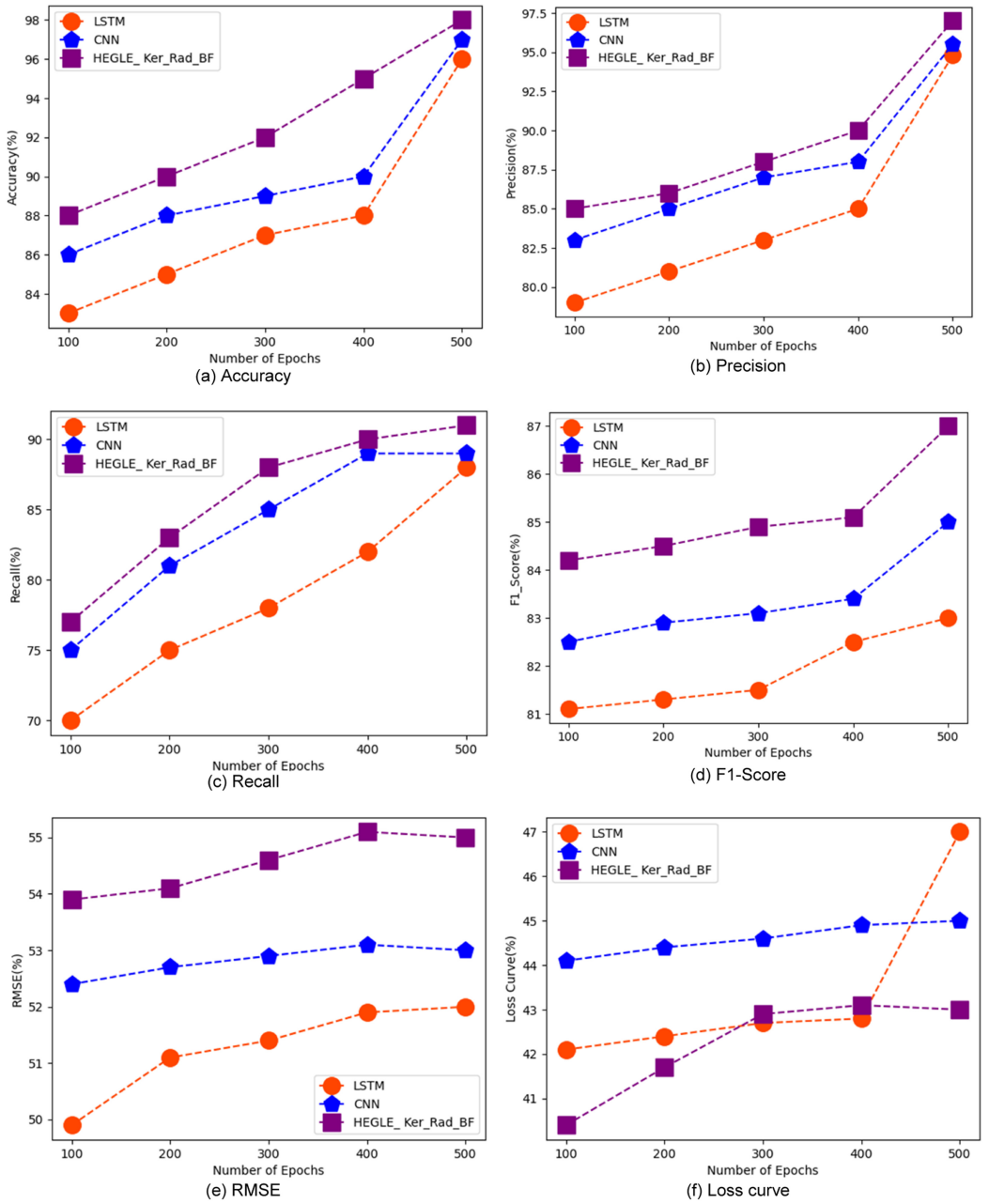

Fig. 3 Comparative analysis for TRAC dataset (a) accuracy, (b) precision, (c) recall, (d) F-1 scor€(e) RMSE, and (f) loss curve analysis.

accuracy, 94.8% of precision value, recall of 88%, 83% is achieved as the F1-Score, and RMSE of 52% and loss curve analysis attained is 47%. This illustrated that the proposed model achieves the increased performance for the multilingual data processing using feature extraction with classification using deep learning architectures.

5 CONCLUSION

Because of the advent of social media platforms such as Twitter, Facebook, and Instagram, the NLP community is very excited about the prospects of Information Extraction technology. Social media reacts to global events more quickly than traditional news sources, and its sub-communities pay special attention to topics that other sources might overlook. This research proposed multilingual data processing using feature extraction with classification through utilization of deep learning models. Here input data in text form is collected and processed based on consideration of different languages for the elimination of null values. The data processed is extracted through Histogram Equalization based Global Local Entropy (HEGLE) and classified using Kernel-based Radial basis Function (Ker_Rad_BF). These methods could be utilized to process natural language. We present solutions to the multilingual sentiment analysis issue in this research article by implementing techniques, and we compare precision factors to discover the optimum option for multilingual sentiment analysis. The experimental analysis of the proposed model exhibits higher accurate value compared with the conventional technique such as CNN, LSTM, and the proposed model achieves 1.62 times higher timeliness. We can investigate the impact of other feature extractions as well as selection approaches on the method in future work, thereby improving the model's learning capabilities. The suggested method treats distinct characteristics extracted using various extraction techniques in every type of raw data as a separate mode, rather than learning directly from the raw data. More research is needed to figure out how to extract multi-modal fusion low-xdimensional characteristics directly from original multi-modal data. For the HASOC dataset, the proposed HEGLE_ Ker_Rad_BF achieves 98% accuracy, 97% precision value, 90.5% of recall value, 85% of F1-score, RMSE of 55.6%, and a loss curve analysis attained is 44%. For the TRAC dataset, the accuracy is 98%, the precision attained is 97%, the Recall is 91%, the F-1 score is 87%, and RMSE of the proposed neural network is 55%.

REFERENCES

[1] D. Wang, J. Su, and H. Yu. 2020. Feature extraction and analysis of natural language processing for deep learning English language. *IEEE Access* 8 (2020), 46335–46345.

[2] A. Kumar and T. M. Sebastian. 2012. Sentiment analysis on Twitter. *International Journal of Computer Science Issues (IJCSI)* 9, 4 (2012), 372.

[3] T. Young, D. Hazarika, S. Poria, and E. Cambria. 2018. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine* 13, 3 (2018), 55–75.

[4] A. Kumar and V. H. C. Albuquerque. 2021. Sentiment analysis using XLM-R Transformer and zero-shot transfer learning on resource-poor Indian language. *Transactions on Asian and Low-Resource Language Information Processing* 20, 5 (2021), 1–13.

[5] S. Nivetha. 2020. A survey on speech feature extraction and classification techniques. In *2020 International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 48–53.

[6] A. Kumar. 2021. Contextual semantics using hierarchical attention network for sentiment classification in social internet-of-things. *Multimedia Tools and Applications* (2021), 1–16.

[7] A. Kumar and A. Jaiswal. 2020. Systematic literature review of sentiment analysis on Twitter using soft computing techniques. *Concurrency and Computation: Practice and Experience* 32, 1 (2020), e5107.

[8] A. Kumar, K. Srinivasan, W. H. Cheng, and A. Y. Zomaya. 2020. Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data. *Information Processing & Management* 57, 1 (2020), 102141.

[9] A. Kumar, J. M. Chatterjee, and V. G. Díaz. 2020. A novel hybrid approach of SVM combined with NLP and probabilistic neural network for email phishing. *International Journal of Electrical and Computer Engineering* 10, 1 (2020), 486.

[10] S. Chotirat and P. Meesad. 2021. Part-of-speech tagging enhancement to natural language processing for Thai wh-question classification with deep learning. *Heliyon* 7, 10 (2021), e08216.

[11] A. J. Trappey, C. V. Trappey, J. L. Wu, and J. W. Wang. 2020. Intelligent compilation of patent summaries using machine learning and natural language processing techniques. *Advanced Engineering Informatics* 43 (2020), 101027.

[12] P. M. Lavanya and E. Sasikala. 2021. Deep learning techniques on text classification using natural language processing (NLP) in social healthcare network: A comprehensive survey. In *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*. IEEE, 603–609.
[13] K. Pal and B. V. Patel. 2020. Data classification with k-fold cross validation and holdout accuracy estimation methods with 5 different machine learning techniques. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 83–87.
[14] A. Amaar, W. Aljedaani, F. Rustam, S. Ullah, V. Rupapara, and S. Ludi. 2022. Detection of fake job postings by utilizing machine learning and natural language processing approaches. *Neural Processing Letters* (2022), 1–29.
[15] T. Madeira, R. Melício, D. Valério, and L. Santos. 2021. Machine learning and natural language processing for prediction of human factors in aviation incident reports. *Aerospace* 8, 2 (2021), 47.
[16] S. A. Bawazeer, S. S. Baakeem, and A. Mohamad. 2019. A new radial basis function approach based on Hermite expansion with respect to the shape parameter. *Mathematics* 7, 10 (2019), 979.
[17] G. N. Jorvekar and M. Gangwar. 2022. Multi-entity topic modeling and aspect-based sentiment classification using machine learning approach. In *Proceedings of International Conference on Recent Trends in Computing*. Springer, Singapore, 537–547.
[18] A. Alwehaibi and K. Roy. 2018. Comparison of pre-trained word vectors for Arabic text classification using deep learning approach. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 1471–1474.
[19] A. Esmaeilzadeh, M. Heidari, R. Abdolazimi, P. Hajibabaee, and M. Malekzadeh. 2022. Efficient large scale NLP feature engineering with Apache Spark. In *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 0274–0280.
[20] Y. Shi, D. Feng, Y. Cheng, and S. Biswas. 2021. A natural language-inspired multilabel video streaming source identification method based on deep neural networks. *Signal, Image and Video Processing* 15, 6 (2021), 1161–1168.
[21] M. Wang, L. Xu, and L. Guo. 2018. Anomaly detection of system logs based on natural language processing and deep learning. In *2018 4th International Conference on Frontiers of Signal Processing (ICFSP)*. IEEE, 140–144.
[22] S. García-Méndez, F. de Arriba-Pérez, A. Barros-Vila, and F. J. González-Castaño. 2022. Detection of temporality at discourse level on financial news by combining natural language processing and machine learning. *Expert Systems with Applications* 197 (2022), 116648.
[23] M. Muntean and A. Donea. 2018. A multi-agent system based on natural language processing using collective user knowledge-base and GPS databases. *Acta Univ. Apulensis* 56 (2018), 27–33.
[24] S. L. Marie-Sainte, N. Alalyani, S. Alotaibi, S. Ghouzali, and I. Abunadi. 2018. Arabic natural language processing and machine learning-based systems. *IEEE Access* 7 (2018), 7011–7020.
[25] D. Dessì, F. Osborne, D. R. Recupero, D. Buscaldi, and E. Motta. 2021. Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain. *Future Generation Computer Systems* 116 (2021), 253–264.

Received 28 May 2022; revised 24 September 2022; accepted 29 January 2023