International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

A DEEP LEARNING-BASED SENTIMENT ANALYSIS APPROACH FOR ONLINE PRODUCT RANKING WITH PROBABILISTIC LINGUISTIC TERM SETS

1. **Maheshkar Akshay Kumar**, PG Student - M. Tech- Computer Science and Engineering, School of Technology, Dept of CSE, GITAM (Deemed to be University), Hyderabad. amaheshk@gitam.in

2. **Dr. Y. Md. Riyazuddin**, Associate Professor, Dept of CSE, School of Technology, GITAM (Deemed to be University), Hyderabad. rymd@gitam.edu

Abstract: In this paper Develop a novel approach using deep learning and sentiment analysis to enhance online product rankings, addressing the challenge of abundant e-commerce choices and providing a more user-centric evaluation that includes subjective satisfaction factors. Utilize state-of-the-art deep learning techniques to create a robust sentiment analysis model trained on diverse user reviews. Integrate sentiment scores into the existing product ranking algorithm, forming a hybrid approach that combines objective criteria with subjective user experiences. Apply sentiment analysis to unveil insights from unstructured data, benefiting consumers with nuanced product evaluations and offering businesses actionable feedback. Bridge the gap between user feedback and rankings, leading to more informed and satisfactory purchasing decisions. Expect an improved, user-oriented online product ranking system delivering personalized recommendations. Implement diverse deep learning models and assess their performance using standard metrics, benefiting key stakeholders such as e-commerce platforms, businesses relying on user feedback, and consumers seeking informed product recommendations. We extended the project with a highly accurate LSTM , Hybrid model (LSTM + GRU) achieving 99.9% across accuracy, precision, recall, and F1-score. To enhance user testing, we developed a user-friendly Flask-based front end with authentication features, ensuring a secure and personalized user experience.

Index terms *- Sentiment analysis, Text reviews, Text classification, Deep learning, Probabilistic linguistic term set.*

1. INTRODUCTION

Online review systems in current e-commerce platforms become an important information source to affect customers' decisions regarding online shopping [1]. To support customers with extracting useful information from large amounts of reviews and making a purchase decision from a set of products/services regarding multiple criteria, the accurate measurement of online reviews is worth investigating [2], [3]. Probabilities linguistic term set (PLTS) [4], which combines linguistic terms with probabilities to enhance the flexibility and comprehensiveness of uncertain information expression, is an efficient tool to represent sentimental intensities hidden in unstructured text reviews. PLTS has been widely applied to represent linguistic evaluations for text online reviews in multicriteria online product ranking problems under uncertainty [2], [3], [5], [6], [7], [8].

In the current research [1], [9], the common way to address the problem of product ranking based on online reviews is composed of three stages: 1) product features extraction from online reviews, 2) sentiment analysis for calculating the overall sentiment scores of sentiment words of review texts, 3) ranking alternative products based on the results of the first two stages.

Apart from the overall comment for a product, online reviews always contain descriptions and preferences for different product features that will affect the purchasing behavior of a customer. Thus, product features and corresponding sentiment tendencies need to be considered when ranking products. How to effectively extract product features from a large set of online reviews is the basis and an important step of the online review analysis problem [10]. The most widely used method in current research to extract production features is the statistical-based method. The Latent Dirichlet Allocation (LDA) model is a typically generative statistical model. For instance, Tirunillai and Tellis [11] used the LDA to extract the key latent dimensions of consumer satisfaction with quality. Guo et al. [12] and Bi et al. [13] used LDA to extract the features of products/services from online reviews to identify the preferences of customers. The LDA model could identify a number of topics from text documents, where each topic contains several words that are representative of that topic. In other words, the output of the LDA model is the sparse representation of a text, it only keeps the key features which are not related to each other and ignores the irrelevant information. Consequently, the LDA model cannot be applied or extended to retrieve sentiment phrases or sub-sentences that describe a particular feature in a straightforward manner.

2. LITERATURE SURVEY

Over the past few years, more and more consumers have come to read online reviews when they shop online. To support consumers' purchase decisions, many scholars focus on ranking products based on online reviews and propose various methods and techniques [1] . Generally, the process of information fusion for ranking products based on online reviews consists of three stages: product feature extraction, sentiment analysis, and ranking products. In this paper, we review the existing studies on processes and methods of information fusion for each stage. Furthermore, we briefly review the existing research on information fusion based on online reviews in other fields. Finally, we summarize the main conclusions of this paper and point out the future research direction[1,9].

The Online reviews play an important role in online shopping. Previous studies on Multi-Criteria Decision Making (MCDM) based on customer reviews focused too much on sentiment words in text reviews but ignored the personalized semantics of linguistic terms [5]. In addition, only considering the qualitative information of products/services is not enough to simulate the purchase behaviors of customers given that the customers are also concerned with quantitative parameters. To bridge these research gaps, this study models the personalized cognition of customers on both quantitative and qualitative information, and proposes an MCDM framework for online shopping. Firstly, we determine the personalized semantics of linguistic terms through the emotional consistency between star ratings and text reviews. Afterwards, we investigate the “psychological intensity” based on Weber-Fechner's law to determine the utilities of quantitative parameters. A utility-based translation method is then developed to express both quantitative parameters and text reviews as probabilistic linguistic term sets. The unified information is further aggregated to represent the performance of products/services. The applicability of the proposed method is illustrated by a case study of television selection from Amazon.com. The results demonstrate that the personalized cognition has an influence on the judgments of products/services [5].

Online reviews play an important role for the purchasing decision of customers. One challenge is that different reviewers have different judgment benchmarks when making online reviews, which can mislead purchasing decisions. Specifically, the same star rating may correspond to different levels of sentiment for different reviewers because of the explicit preference differences in individuals. [3]This study explores the personal judgment benchmarks through a preference learning process. Considering the nonlinear cognition of reviewers, we propose a marginal value function with smooth shapes and clear parameters to model the scores of online reviews. A mathematical programming model is established to predict the specific marginal value function for each reviewer. Two kinds of performance accurateness are defined to measure the performance of the learning model. We evaluate two empirical data sets extracted from TripAdvisor.com to deepen the understanding of personal judgment benchmarks. A simulation study is conducted to validate the proposed model. The results have important theoretical and practical implications for purchasing decisions based on online reviews [1,19,20].

When expressing preferences in qualitative setting, several possible linguistic terms with different weights (represented by probabilities) may be considered at the same time [4]. The probabilistic distribution is usually hard to be provided completely and ignorance may exist. In this paper, we first propose a novel concept called probabilistic linguistic term set (PLTS) to serve as an extension of the existing tools [19,21,22]. Then we put forward some basic operational laws and aggregation operators for PLTSs. After that, we develop an extended TOPSIS method and an aggregation-based method respectively for multi-attribute group decision making (MAGDM) with probabilistic linguistic information, and apply them to a practical case concerning strategy initiatives. Finally, the strengths and weaknesses of our methods are clarified by comparing them with some similar techniques.

Online reviews have become an increasingly popular information source in consumer’s decision making process. To help consumers make informed decisions, how to select products through online reviews is a valuable research topic. This work deals with a personized product selection problem with review sentiments under probabilistic linguistic circumstances. To this end, we propose a multi-criteria decision making (MCDM) method incorporating personalized heuristic judgments in the prospect theory (PT) [5]. We focus on the role of personalized heuristic judgments on review helpfulness in the final decision outcomes. We demonstrate the consistency between the three

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

common heuristic judgments (with respect to review valence, sentiment extremity, and aspiration levels) and the three behavioral principles of the PT. Then, the products are ranked with the probabilistic linguistic term set (PLTS) input, based on the proposed adjustable PT framework, in which the coefcients of negativity bias are derived from the consumer's heuristic judgments [19,21,22]. Finally, a real case on TripAdvisor.com and two simulation experiments are given to illustrate the validity of the proposed method.

3. METHODOLOGY

i) Proposed Work:

The proposed system introduces a novel approach by incorporating sentiment analysis into online product ranking through advanced deep learning techniques. The proposed system aims to extract product features and retrieve corresponding texts that represent only the selected features, eliminating irrelevant information. This method utilizes natural language processing (NLP) techniques to make full use of the features and sentiment tendencies of input reviews. Through advanced deep learning and sentiment analysis, the system extracts specific product features, improving the precision and relevance of understanding user sentiments. The system narrows down information retrieval to selected features, eliminating irrelevant details. This enhances the efficiency of sentiment analysis, providing users with more focused and useful insights. [5,18] Leveraging NLP techniques, the system maximizes the utilization of features and sentiment tendencies in reviews, contributing to a sophisticated understanding of user sentiments. By integrating sentiment analysis and feature extraction, the system tailors insights to represent user sentiments toward selected product features. This personalized approach enhances decision-making for consumers.

We implemented a LSTM ,Hybrid model (LSTM + GRU) that achieved exceptional accuracy, precision, recall, and F1-score, reaching 99.9%. To facilitate user testing and engagement, we built a user-friendly front end using the Flask framework. The front end will incorporate authentication features, ensuring a secure and personalized experience for users interacting with the system.

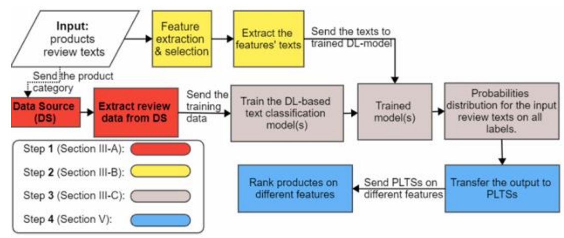

Fig 1 System Architecture

iii) Dataset collection:

The study utilizes the sentiment dataset to understand its structure, features, and sentiment labels. This step includes loading the dataset , checking for missing values, and gaining insights into the distribution of sentiment classes.

iv) Data Processing:

In the context of data processing, data refers to numbers or characters that constitute measurements from the real world. Once professionals have measured the information present in multiple data, they can derive useful information algorithmically and deduce it statistically. This information may serve as a solution to different organisational

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362
IJARST

problems like high data management costs or inefficient supply chain processes. For instance, if a business has gathered data relative to its operations, the next step is to develop this data into meaningful and easy-to-access presentations for the management. Executive managers can use this structured information to make decisions that may increase revenue and decrease the loss. To convert data into organised information, organisations typically use data processing applications. This data may include operational or transactional details and sales, inventory or payroll information.

- Removing URL and Other Characters: Eliminates irrelevant characters and URLs that may not contribute to sentiment analysis.

- Remove Punctuations: Helps in cleaning the text data by removing unnecessary punctuations. - Remove Stop Words: Eliminates common words that don't contribute much to sentiment.

- Normalization of Data: Standardizes the text data to a consistent format.

- Tokenize and Lemmatize: Breaks down sentences into individual words and reduces them to their base or root form.

- Vectorize the Text (Lexicon-Based Approach): Converts text data into numerical form, essential for machine learning models.

v) Feature selection:

Feature selection is the process of isolating the most consistent, non-redundant, and relevant features to use in model construction. Methodically reducing the size of datasets is important as the size and variety of datasets continue to grow. The main goal of feature selection is to improve the performance of a predictive model and reduce the computational cost of modeling. Feature selection, one of the main components of feature engineering, is the process of selecting the most important features to input in machine learning algorithms. Feature selection techniques are employed to reduce the number of input variables by eliminating redundant or irrelevant features and narrowing down the set of features to those most relevant to the machine learning model. The main benefits of performing feature selection in advance, rather than letting the machine learning model figure out which features are most important.

vi) Algorithms:

A Convolutional Neural Network (CNN) is a specialized architecture for visual data processing, employing convolutional layers to learn hierarchical features and spatial patterns [32,33]. Widely used in image recognition, CNNs capture local features in input data. Char CNN is designed for character-level operations, utilizing convolutional layers to extract features from text. Employed in the project to capture intricate patterns in user reviews, it enhances understanding of nuances often missed by word-level models.

```python
def tokenizer(x_train, y_train, max_len_word):
    # because the data distribution is imbalanced, "stratify" is used
    X_train, X_val, y_train, y_val = train_test_split(x_train, y_train,
                                                       test_size=.2, shuffle=True,
                                                       stratify=y_train, random_state=0)

    # Tokenizer
    tokenizer = Tokenizer(num_words=5000)
    tokenizer.fit_on_texts(X_train)
    sequence_dict = tokenizer.word_index
    word_dict = dict((num, val) for (val, num) in sequence_dict.items())

    # Sequence data
    train_sequences = tokenizer.texts_to_sequences(X_train)
    train_padded = pad_sequences(train_sequences,
                                 maxlen=max_len_word,
                                 truncating='post',
                                 padding='post')

    val_sequences = tokenizer.texts_to_sequences(X_val)
    val_padded = pad_sequences(val_sequences,
                               maxlen=max_len_word,
                               truncating='post',
                               padding='post', )

    print(train_padded.shape)
    print(val_padded.shape)
    print('Total words: {}'.format(len(word_dict)))
    return train_padded, val_padded, y_train, y_val, word_dict

X_train, X_val, y_train, y_val, word_dict = tokenizer(df.text, df.sentiment, 100)
```

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
IJARST
ISSN: 2457-0362

Fig.2 CNN

A Recurrent Neural Network (RNN) is designed for sequential data processing, maintaining memory of past inputs through recurrent connections. Suited for tasks like natural language processing and time series analysis. [32,33] Text RNN processes sequential data, maintaining hidden states for context from previous inputs. Employed in the project to capture sequential dependencies in user reviews, aiding in understanding sentiments in complex linguistic structures.

```
def textrnn():
    inputs = Input(name='inputs',shape=[max_len])
    layer = Embedding(max_words,50,input_length=max_len)(inputs)
    layer = SimpleRNN(64)(layer)
    layer = Dense(256,name='FC1')(layer)
    layer = Activation('relu')(layer)
    layer = Dropout(0.5)(layer)
    layer = Dense(1,name='out_layer')(layer)
    layer = Activation('sigmoid')(layer)
    model = Model(inputs=inputs,outputs=layer)
    return model
```

```
SINGLE_ATTENTION_VECTOR = False
APPLY_ATTENTION_BEFORE_LSTM = False
def attention_3d_block(inputs):
    # inputs.shape = (batch_size, time_steps, input_dim)
    input_dim = int(inputs.shape[2])
    a = Permute((2, 1))(inputs)
    a = Reshape((input_dim, TIME_STEPS))(a) # this line is not useful. It's just to know which dimension is what.
    a = Dense(TIME_STEPS, activation='softmax')(a)
    if SINGLE_ATTENTION_VECTOR:
        a = Lambda(lambda x: K.mean(x, axis=1), name='dim_reduction')(a)
        a = RepeatVector(input_dim)(a)
    a_probs = Permute((2, 1), name='attention_vec')(a)
    # output_attention_mul = merge([inputs, a_probs], name='attention_mul', mode='mul')
    output_attention_mul = multiply([inputs, a_probs])
    return output_attention_mul
```

Fig 3 RNN

Text CNN is a convolutional neural network tailored for text-based tasks, utilizing convolutional layers to capture local patterns and hierarchical representations in word sequences. Employed in the project to efficiently recognize important phrases and word combinations in user reviews, enhancing the understanding of sentiments expressed in textual data.

```
tokenizer = Tokenizer(num_words=10000)
tokenizer.fit_on_texts(X_train)

sequences = tokenizer.texts_to_sequences(X_train)

tr_x = pad_sequences(sequences, maxlen=50)
tr_y = to_categorical(Y_train)

sequences = tokenizer.texts_to_sequences(X_test)
val_x = pad_sequences(sequences, maxlen=50)
val_y = to_categorical(Y_test)

sequences = tokenizer.texts_to_sequences(X_test)
ts_x = pad_sequences(sequences, maxlen=50)
ts_y = to_categorical(Y_test)
```

```
max_words = 10000
max_len = 50
embedding_dim = 64
```

Fig 4 Text CNN

Seq2Seq is a neural network architecture designed for sequence-to-sequence tasks, utilizing recurrent neural networks (RNNs) or transformers to capture dependencies in sequential data. Employed in the project for tasks like text summarization, language translation, or potentially generating product descriptions and translating user reviews in the e-commerce context.

```
from keras.layers import Dense, Input, Flatten
from keras.layers import GlobalAveragePooling1D, Embedding
from keras.models import Model

EMBEDDING_DIM = 50
N_CLASSES = 3

# input: a sequence of MAX_SEQUENCE_LENGTH integers
sequence_input = Input(shape=(MAX_SEQUENCE_LENGTH,), dtype='int32')

embedding_layer = Embedding(MAX_NB_WORDS, EMBEDDING_DIM,
                            input_length=MAX_SEQUENCE_LENGTH,
                            trainable=True)
embedded_sequences = embedding_layer(sequence_input)

average = GlobalAveragePooling1D()(embedded_sequences)
predictions = Dense(N_CLASSES, activation='softmax')(average)

model = Model(sequence_input, predictions)
model.compile(loss='categorical_crossentropy',
              optimizer='adam', metrics=['accuracy',f1_m,precision_m, recall_m])
```

```
hist = model.fit(x_train, y_train, validation_split=0.1,
          epochs=10, batch_size=8)
```

Fig 5 Seq2Seq

BERT, a transformer-based neural network, emphasizes understanding human-like language. It employs an encoder-only architecture, prioritizing input sequence comprehension over sequence generation. Unlike traditional models, BERT considers both left and right context simultaneously,

IJARST
International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

enabling a more comprehensive understanding of textual data. In the project, BERT is likely used for tasks requiring deep contextual understanding, such as sentiment analysis, feature extraction, or other e-commerce-related tasks.

```
def bert_encode(data,maximum_length) :
    input_ids = []
    attention_masks = []


    for i in range(len(data.text)):
        encoded = tokenizer.encode_plus(

        data.text[i],
        add_special_tokens=True,
        max_length=maximum_length,
        pad_to_max_length=True,

        return_attention_mask=True,

      )

        input_ids.append(encoded['input_ids'])
        attention_masks.append(encoded['attention_mask'])
    return np.array(input_ids),np.array(attention_masks)
```

Fig 6 BERT

Fast Text is a text classification algorithm utilizing word embeddings. When combined with a Convolutional Neural Network (CNN), it efficiently captures local patterns in text data. In the project, Fast Text CNN is likely employed for tasks like sentiment analysis in an e-commerce setting, focusing on understanding specific word combinations or phrases.

```
class FastText(nn.Module):
    def __init__(self, vocab_size, embedding_dim, hidden_dim, output_dim, n_layers,
                 bidirectional, dropout, pad_idx):

        super().__init__()

        self.embedding = nn.Embedding(vocab_size, embedding_dim, padding_idx = pad_idx)

        self.rnn = nn.LSTM(embedding_dim,
                           hidden_dim,
                           num_layers=n_layers,
                           bidirectional=bidirectional,
                           dropout=dropout)

        self.fc1 = nn.Linear(hidden_dim * 2, hidden_dim)

        self.fc2 = nn.Linear(hidden_dim, 1)

        self.dropout = nn.Dropout(dropout)

    def forward(self, text, text_lengths):

        # text = [sent len, batch size]

        embedded = self.embedding(text)

        # embedded = [sent len, batch size, emb dim]

        #pack sequence
        packed_embedded = nn.utils.rnn.pack_padded_sequence(embedded, text_lengths)

        packed_output, (hidden, cell) = self.rnn(packed_embedded)

        hidden = self.dropout(torch.cat((hidden[-2,:,:], hidden[-1,:,:]), dim = 1))
        output = self.fc1(hidden)
        output = self.dropout(self.fc2(output))

        #hidden = [batch size, hid dim * num directions]

        return output
```

Fig 7 Fast text CNN

LSTM, an improved version of recurrent neural networks, excels in capturing long-term dependencies crucial for sequence prediction tasks. With three gates controlling information flow, LSTMs selectively retain or discard data, making them suitable for tasks like language translation, speech recognition, and time series forecasting. In the project, LSTM is employed for sequential data tasks, such as user reviews or product descriptions, effectively capturing dependencies for sentiment analysis and contextual understanding.

```
embed_dim = 128 #dimension of the word embedding vector for each word in a sequence
lstm_out = 196  #no of lstm layers
lstm_model = Sequential()
lstm_model.add(Embedding(num_words, embed_dim,input_length = X_train.shape[1]))
#Adding dropout
lstm_model.add(LSTM(lstm_out, dropout=0.2, recurrent_dropout=0.2))
#Adding a regularized dense layer
lstm_model.add(layers.Dense(32,kernel_regularizer=regularizers.l2(0.001),activation='relu'))
lstm_model.add(layers.Dropout(0.5))
lstm_model.add(Dense(3,activation='softmax'))
lstm_model.compile(loss = 'categorical_crossentropy', optimizer='adam',metrics = ['accuracy',f1_m,precision_m, recall_m])
print(lstm_model.summary())
```

Fig 8 LSTM

The hybrid model combining LSTM and GRU (Gated Recurrent Unit), both types of RNN architectures, serves as a simpler alternative to LSTM networks. GRU uses gating mechanisms to selectively update the hidden state, with the reset gate controlling forgetting of the previous state and the update gate determining the influence of new input. In the project, this hybrid model is employed for comprehensive user reviews, leveraging the strengths of both architectures for improved training efficiency and performance in handling long sequences.

LSTM + GRU

```
model = Sequential()
model.add(Embedding(num_words, embed_dim,input_length = X_train.shape[1]))
model.add(LSTM(64,dropout=0.4, recurrent_dropout=0.4,return_sequences=True))
model.add(GRU(32,dropout=0.5, recurrent_dropout=0.5,return_sequences=False))
model.add(Dense(3,activation='softmax'))
model.compile(loss = 'categorical_crossentropy', optimizer='adam',metrics = ['accuracy',f1_m,precision_m, recall_m])
print(model.summary())
```

IJARST
International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

Fig 9 LSTM + GRU

4. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{\mathit{True\ Positive}}{\mathit{True\ Positive} + \mathit{False\ Positive}}$$

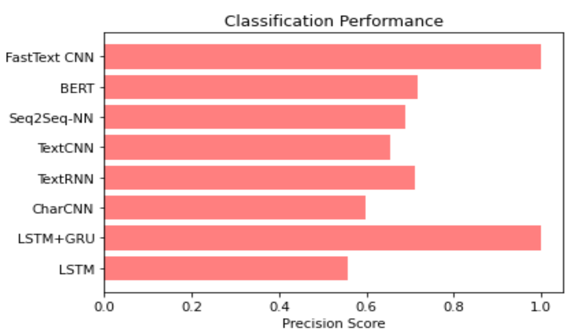

Fig 10 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

$$\mathit{Recall} = \frac{TP}{TP + FN}$$

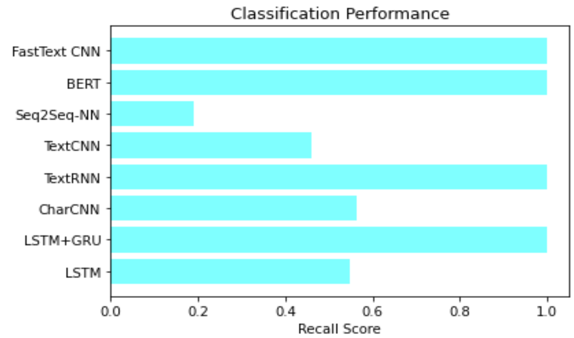

Fig 11 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

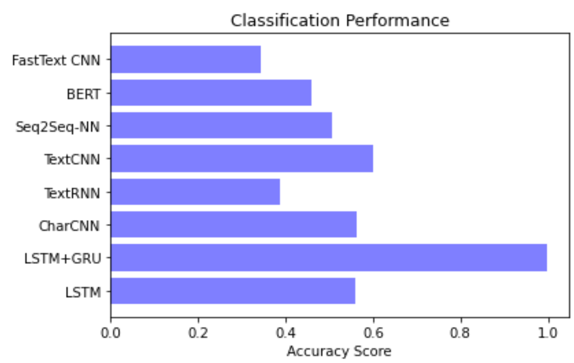

Fig 12 Accuracy graph

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

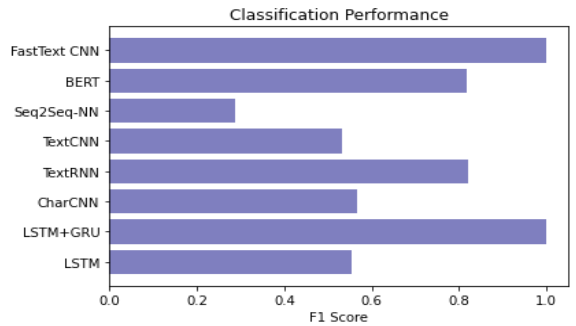

Fig 13 F1Score

ML Model	Accuracy	Precision	Recall	F1 - score
Extension LSTM	0.559	0.557	0.549	0.553
Extension LSTM+GRU	0.999	0.999	0.999	0.999
Char CNN	0.562	0.597	0.562	0.566
Text RNN	0.386	0.711	1.000	0.822
Text CNN	0.601	0.655	0.460	0.533
Seq2Seq-NN	0.508	0.688	0.189	0.286
BERT	0.458	0.717	1.000	0.817
Fast Text CNN	0.343	1.000	1.000	1.000

Fig 14 Performance Evaluation table

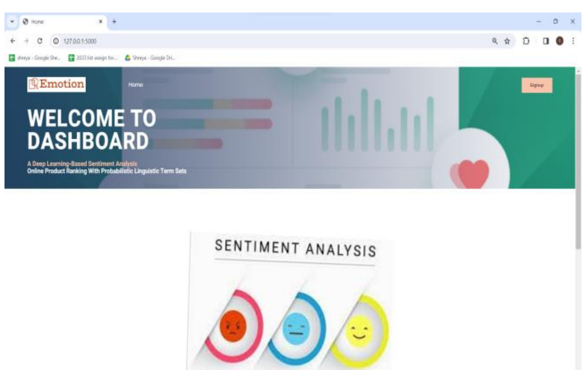

Fig 15 Home page

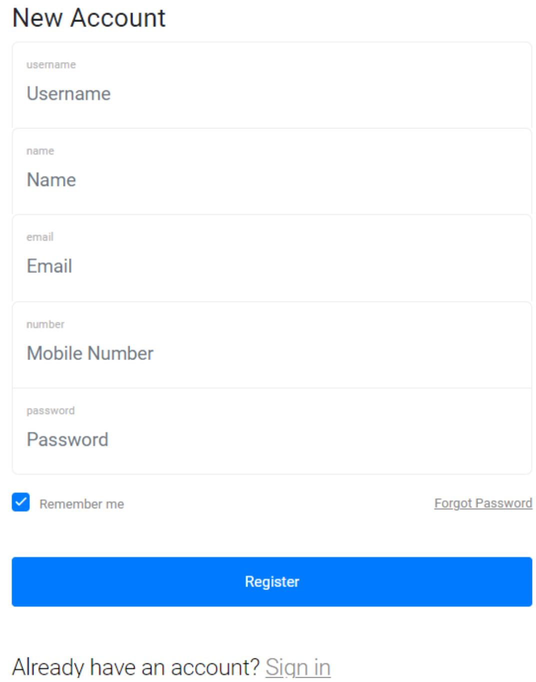

Fig 16 Registration page

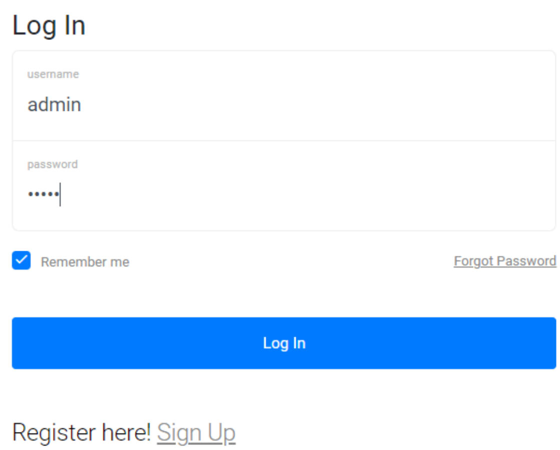

Fig 17 Login page

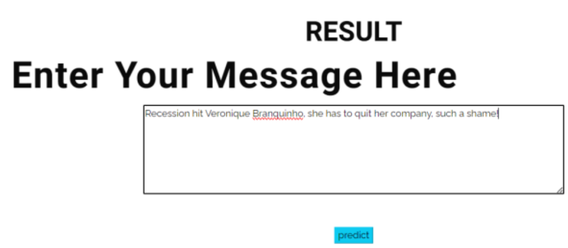

Fig 18 User input

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

Fig 19 Predict result for given input

5. CONCLUSION

The project successfully implemented and evaluated a diverse set of advanced sentiment analysis techniques, including Char CNN, Text RNN, Text CNN, Seq2Seq, BERT, and FastText CNN, showcasing a comprehensive exploration of state-of-the-art models. The project systematically explored various deep learning models, each designed to capture unique aspects of sentiment in textual data. Evaluation metrics, including accuracy, precision, recall, and F1 score, were employed to rigorously assess and compare the performance of these models. The Hybrid model (LSTM + GRU) extension excels with 99.9% accuracy, demonstrating superior performance and robustness, making it an effective solution for various e-commerce data analysis tasks. The integration of the sentiment analysis models into a Flask framework with SQLite for user signup and signin facilitated a user-friendly interface. Users can input text for sentiment prediction, and the final outcomes, along with LDA-based topic modeling results [11,13], are effectively displayed through the frontend, providing an accessible and interactive platform. The project's outcomes hold significant value for various stakeholders, including e-commerce platforms, businesses relying on user feedback, and consumers seeking more informed and nuanced product recommendations. By enhancing user experience, guiding purchasing decisions, and providing valuable insights, the project has the potential to positively impact both businesses and end-users in the realm of sentiment analysis.

6. FUTURE SCOPE

Explore incorporating not only text-based sentiment analysis but also considering visual and auditory cues from user-generated content, expanding the system's capability to understand sentiments across various modalities. Extend the system to provide real-time sentiment monitoring, allowing e-commerce platforms to dynamically adapt rankings based on evolving user sentiments and ensuring up-to-date recommendations [8,18,19]. Integrate user profiling to enhance personalization, understanding individual preferences over time and tailoring product recommendations based on historical sentiment data, providing a more personalized shopping experience. Explore integration with emerging technologies such as blockchain for transparent and secure sentiment data storage, and augmented reality (AR) for immersive product experiences, ensuring the project remains at the forefront of technological advancements in e-commerce.

REFERENCES

[1] Z. P. Fan, G. M. Li, and Y. Liu, “Processes and methods of information fusion for ranking products based on online reviews: An overview,” Information Fusion, vol. 60, pp. 87–97, Aug. 2020.

[2] X. Wu and H. Liao, “Modeling personalized cognition of customers in online shopping,” Omega (Westport), vol. 104, p. 102471, Oct. 2021.

[3] X. Wu and H. Liao, "Learning judgment benchmarks of customers from online reviews," OR Spectrum, vol. 43, no. 4, pp. 1125–1157, 2021.

[4] Q. Pang, H. Wang, and Z. Xu, "Probabilistic linguistic term sets in multiattribute group decision making," Inf Sci (N Y), vol. 369, pp. 128–143, Nov. 2016.

[5] M. Zhao, X. Shen, H. Liao, and M. Cai, "Selecting products through text reviews: An MCDM method incorporating personalized heuristic judgments in the prospect theory," Fuzzy Optimization and Decision Making, 2021.

[6] H. gang Peng, H. yu Zhang, and J. qiang Wang, "Cloud decision support model for selecting hotels on TripAdvisor.com with probabilistic linguistic information," Int J Hosp Manag, vol. 68, pp. 124–138, Jan. 2018.

[7] P. Liu and F. Teng, "Probabilistic linguistic TODIM method for selecting products through online product reviews," Inf Sci (N Y), vol. 485, pp. 441–455, Jun. 2019.

[8] D. Liang, Z. Dai, M. Wang, and J. Li, "Web celebrity shop assessment and improvement based on online review with probabilistic linguistic term sets by using sentiment analysis and fuzzy cognitive map," Fuzzy Optim Decis Making, vol. 19, pp. 561–586, 2020.

[9] G. Kou et al., "A cross-platform market structure analysis method using online product reviews," Technological and Economic Development of Economy, vol. 27, no. 5, pp. 992–1018, 2021. [10] Z. Yan, M. Xing, D. Zhang, and B. Ma, "EXPRS: An extended pagerank method for product feature extraction from online consumer reviews," Information & Management, vol. 52, no. 7, pp. 850–858, Nov. 2015.

[11] S. Tirunillai and G. J. Tellis, "Mining Marketing Meaning from Online Chatter: Strategic Brand Analysis of Big Data Using Latent Dirichlet Allocation," Journal of Marketing Research, vol. 51, no. 4, pp. 463–479, 2014.

[12] Y. Guo, S. J. Barnes, and Q. Jia, "Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation," Tour Manag, vol. 59, pp. 467–483, Apr. 2017.

[13] J. W. Bi, Y. Liu, Z. P. Fan, and J. Zhang, "Wisdom of crowds: Conducting importance-performance analysis (IPA) through online reviews," Tour Manag, vol. 70, pp. 460–478, Feb. 2019.

[14] A. Kangale, S. K. Kumar, M. A. Naeem, M. Williams, and M. K. Tiwari, "Mining consumer reviews to generate ratings of different product attributes while producing feature-based review-summary," Int J Syst Sci, vol. 47, no. 13, pp. 3272–3286, 2016.

[15] T.-L. Wong and W. Lam, "Hot item mining and summarization from multiple auction web sites," in Fifth IEEE International Conference on Data Mining (ICDM'05), 2005, pp. 4—pp.

[16] T.-L. Wong and W. Lam, "Learning to extract and summarize hot item features from multiple auction web sites," Knowl. Inf. Syst., vol. 14, no. 2, pp. 143–160, 2008.

[17] J. Cai, J. Li, W. Li, and J. Wang, "Deep learning Model used in text classification," in Proc. 15th Int.

Comput. Conf. Wavelet Act. Media Technol. Inf. Process., 2018, pp. 123–126.

[18] R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in Proc. Conf. Empirical Methods Natural Lang. Process., 2013, pp. 1631–1642.

[19] Z. Zhang, Q. Ye, Z. Zhang, and Y. Li, "Sentiment classification of Internet restaurant reviews written in Cantonese," Expert Syst. Appl., vol. 38, no. 6, pp. 7674–7682, Jun. 2011.

[20] Z. P. Fan, Y. J. Che, and Z. Y. Chen, "Product sales forecasting using online reviews and historical sales data: A method combining the bass model and sentiment analysis," J. Bus. Res., vol. 74, pp. 90–100, May 2017.

[21] D. Zhang, H. Xu, Z. Su, and Y. Xu, "Chinese comments sentiment classification based on word2vec and SVMperf," Expert Syst. Appl., vol. 42, no. 4, pp. 1857–1863, Mar. 2015.

[22] F. Tian et al., "A topic sentence-based instance transfer method for imbalanced sentiment classification of Chinese product reviews," Electron. Commerce Res. Appl., vol. 16, pp. 66–76, Mar. 2016.

[23] Y. Liu, J.-W. Bi, and Z.-P. Fan, "A method for ranking products through online reviews based on sentiment classification and interval-valued intuitionistic fuzzy TOPSIS," Int. J. Inf. Technol. Decis. Mak., vol. 16, no. 6, pp. 1497–1522, Sep. 2017.

[24] U. Schmalz, J. Ringbeck, and S. Spinler, "Door-to-door air travel: Exploring trends in corporate reports using text classification models," Technol. Forecasting Soc. Change, vol. 170, Sep. 2021, Art. no. 120865.

[25] Y. Hu and W. Li, "Document sentiment classification by exploring description model of topical terms," Comput. Speech Lang., vol. 25, no. 2, pp. 386–403, Apr. 2011.

[26] N. A. Vidya, M. I. Fanany, and I. Budi, "Twitter sentiment to analyze net brand reputation of mobile phone providers," Procedia Comput. Sci., vol. 72, pp. 519–526, Jan. 2015.

[27] A. Onan, S. Koruko ̆ glu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," Expert Syst. Appl., vol. 57, pp. 232–247, Sep. 2016.

[28] A. Onan, "An ensemble scheme based on language function analysis and feature engineering for text genre classification," J. Inf. Sci., vol. 44, no. 1, pp. 28–47, Feb. 2018.

[29] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting deep learning models for tabular data," Adv. Neural Inf. Process. Syst., vol. 34, pp. 18932–18943, 2021.

[30] A. Onan and S. Koruko ̆ glu, "A feature selection model based on genetic rank aggregation for text sentiment classification," J. Inf. Sci., vol. 43, no. 1, pp. 25–38, Feb. 2017.

[31] M. Al-Smadi, O. Qawasmeh, M. Al-Ayyoub, Y. Jararweh, and B. Gupta, "Deep recurrent neural network vs. support vector machine for aspectbased sentiment analysis of Arabic hotels' reviews," J. Comput. Sci., vol. 27, pp. 386–393, Jul. 2018.

[32] H. H. Do, P. W. C. Prasad, A. Maag, and A. Alsadoon, “Deep learning for aspect-based sentiment analysis: A comparative review,” Expert Syst. Appl., vol. 118, pp. 272–299, Mar. 2019.

[33] M. E. Mowlaei, M. S. Abadeh, and H. Keshavarz, “Aspect-based sentiment analysis using adaptive aspect-based lexicons,” Expert Syst. Appl., vol. 148, Jun. 2020, Art. no. 113234.

[34] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[35] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.