
Optimal Trained Bi-Long Short Term Memory for Aspect Based Sentiment Analysis with Weighted Aspect Extraction

Archana Nagelli and B. Saleena*

*School of Computer Science and Engineering, Vellore Institute of Technology,
Chennai Campus, Chennai, India*

E-mail: archana.nagelli2018@vitstudent.ac.in; saleena.b@vit.ac.in

**Corresponding Author*

Received 04 March 2022; Accepted 13 November 2022;
Publication 28 December 2022

Abstract

Sentiment analysis based on aspects seeks to anticipate the polarities of sentiment in specified targets related to the text data. Several studies have shown a strong interest in using an attention network to represent the target as well as context on generating an efficient representation of features used for tasks while sentiment classification. Still, the attention score computation of the target using an average vector for context is unequal. While the interaction mechanism is simplistic, it needs to be overhauled. Therefore, this paper intends to introduce a novel aspect-based sentiment analysis with three phases: (i) Preprocessing, (ii) Aspect Sentiment Extraction, (iii) Classification. Initially, the input data is given to the preprocessing phase, in which the tokenization, lemmatization, and stop word removal are performed. From the preprocessed data, the weighted implicit and weighted explicit extraction is determined in the Aspect Sentiment Extraction. Moreover, the weighted implicit aspect extraction is done by Stanford Dependency Passer (SDP) method, and the weighted explicit extraction is done through proposed Association Rule Mining (ARM). Subsequently, the extracted features are

Journal of Web Engineering, Vol. 21.7, 2115–2148.

doi: 10.13052/jwe1540-9589.2176

© 2022 River Publishers

provided to the classification phase in which the Optimized Bi-LSTM is utilized. For making the classification more accurate and precise, it is planned to tune the weights of Bi-LSTM optimally. For this purpose, an Opposition Learning Cat and Mouse-Based Optimization (OLCMBO) Algorithm will be introduced in this work. In the end, the outcomes of the presented approach are calculated to the extant approaches with respect to different measures like F1-measure, specificity, Negative Predictive Value (NPV), accuracy, False Negative Rate (FNR), sensitivity, precision, False Positive Rate(FPR), and Matthew's correlation coefficient, respectively.

Keywords: Aspect-based sentiment analysis, Stanford dependency passer, association rule mining, optimized Bi-LSTM, optimization.

Nomenclature

Abbreviation	Description
FPR	False Positive Rate
ABSA	Aspect Based Sentiment Analysis
TSHM	Two-Stage Hybrid Model
MTKFN	Multisource Textual Knowledge Fusing Network
FDN	Feature Distillation Network
SOBA	Semi-automated Ontology Builder for ABSA
FNR	False Negative Rate
GANN	Gated Alternate Neural Network
SVM	Support vector Machine
NPV	Net Predictive Value
Bi-LSTM	Bidirectional Long Short Term Memory
NLP	Natural Language Processing
OLCMBO	Opposition Learning Cat and Mouse-Based Optimization
CNNs	Convolutional Neural Networks
ARM	Association Rule Mining
LSTM	Long Short Term Memory
GTR	Gate Truncation RNN
CMBO	Cat and Mouse-Based Optimization
RNNs	Recurrent neural networks
SDP	Stanford Dependency Passer
ALDONAr	A Lexicalized Domain Ontology and a Regularized Neural Attention
ReMemNN	Recurrent Memory Neural Network

1 Introduction

Sentiment analysis has elevated the importance of NLP research in recent years. Moreover, it is also utilized in Intelligent Recommender Systems, Question Answering, Data Mining, Information Retrieval, summarization, and other areas [1, 2]. Large volumes of review texts are published in e-commerce platforms, expressing consumers negative or positive feelings about various elements of the services and products they obtained, such as pricing, logistics, quality, design, and so on [3, 4]. This big data revolution is a critical tool for assisting consumers in reducing information overload and valuable information [5]. Any attribute or property of a single entity could be considered as Aspect. For instance, the product is normally the entity, and everything associated to it (such as price, quality, and so on) is a Aspect in product reviews [6, 7].

ABSA intends to establish the sentiment polarity (e.g., neutral, negative, or positive) that is an Aspect of a single entity of a sentence directed at a target [8, 9]. Sentiment categorization and Aspect identification are two subtasks of the ABSA task [10, 11]. ABSA is a critical task in NLP that has attracted a lot of research [12]. To overcome this problem, early efforts are mostly used with ML methods. Based on the effectiveness in automated feature extraction, NNs [13] have recently become popular solutions for sentiment analysis. Sentiment analysis using NN like LSTM, CNN, attention networks, and Tree LSTM has equaled or surpassed the accuracy of approaches relying on manual features [14, 15]. On the other hand, the ABSA job is distinct from the sentiment categorization. Multiple targets may exist in a single sentence in the ABSA task, and each target contains associated terms that can be used to change it. The attention mechanism utilized in the ABSA challenge to acquire context features linked with the target for sentiment analysis has proven to be beneficial [16, 17].

In an Aspect Based Sentimental Analysis (ABSA) challenge, RNNs are frequently employed to represent sentiment polarity toward a categorization or an aspect word [13]. RNN can represent sentence sequence data and capture long-distance dependence, but it still exhibits sensitivity and position invariance to the local key pattern [18, 19]. Several constraints may impair their performance when RNNs are used in the ABSA task. However, CNNs have position invariance or capture essential local properties. Nevertheless, they are helpless when simulating long-distance reliance [20, 21]. Additionally, CNN does not pay attention to word order, which might lead to incorrect decisions, such as “not good, but poor” and “not poor, but good.”

The attention mechanism is one possible technique. Aspect Based Sentiment Analysis (ABSA) is a fine-grained branch of sentiment analysis that identifies and extracts the various characteristics of the subjects [40]. Layer-wise pre-training based DBN totally extracts the nonlinear features and invariant structures of each frequency [45]. However, it will generate some noise, leading to the model acquiring irrelevant information about the target's current aspect. As a consequence, minimizing such noise is also necessary [22]. The motivation of this work is to investigate the semiautomatic method in sentiment analysis context for Aspect extraction. This research work enhances the quality of the constructed words. The subset was correctly classified through the proposed approach. A context learning function was designed for capturing the relationship among local context words. The proposed method was appropriately compared with the domain adaptation.

The major contribution of this paper is given below:

- The weighted implicit aspect extraction is done by Stanford Dependency Parser (SDP) method, and the weighted explicit extraction is done via the proposed Association Rule Mining (ARM).
- For tuning the weights of Bi-LSTM optimally, an Opposition Learning Cat and Mouse-Based Optimization (OLCMBO) Algorithm was implemented.

In this paper, an exhaustive review of the related work on ABSA model is discussed in Section 2. Overall description of the Aspect-based sentiment analysis model is portrayed in Section 3. Pre-processing phase and Aspect sentiment extraction phase are elaborated in Section 4. Section 5 describes the classification via an optimized bidirectional long short-term memory. Section 6 depicts the weight optimization of BI-LSTM via opposition learning cat and mouse-based optimization algorithm. Finally, Section 7 determines the result and discussion and Section 8 discusses about the conclusion of this paper.

2 Review of Related Work

In 2020, Donatas et al. [23] developed a hybrid approach for sentence-level ABSA using the ALDONAr framework. The “bidirectional context attention method” was created to determine the impact of a phrase in an aspect on each word's sentiment value. The categorization module was created to deal with a sentence's complicated structure. To use field-specific knowledge, and manually generated lexicalized domain ontology was incorporated.

ALDONA employed the Adam optimizer, BERT word embeddings, different model initialization, and a regularization than the previous ALDONA model. Furthermore, two 1D CNN layers were added to its classification module resulting in better results on typical datasets.

In 2020, Ning et al. [24] developed ReMemNN to address the identified issues by researchers. To address the shortcomings of pre-trained word embeddings, ReMemNN includes an embedding adjustment learning module that converts pre-trained into adjustment embeddings of word. Furthermore, an explicit memory module was created to store these representations and construct representations as well as hidden states. ReMemNN outperformed standard baselines and achieved traditional outcomes in all datasets, according to extensive experimental results. Furthermore, these findings show that the proposed model was language and dataset independent.

In 2019, Lisa et al. [25] had examined the possibility of increasing the efficiency and efficacy of knowledge-driven ABSA. A SOBA model was used in this research. Semi-automation of the ontology creation process might result in more comprehensive ontologies with less time spent developing them. Additionally, SOBA wanted to increase the usefulness of its ontologies in ABSA by associating semantics given by a semantic lexicon with ideas. To assess SOBA's performance, laptop domains were developed using the ontology builder and ontologies for the restaurant. These ontologies were computed to manually produced ontologies in the TSHM, a traditional knowledge-driven ABSA approach. Thus, the results suggested that a machine cannot match the human-made ontology quality since SOBA would not increase the TSHM efficacy while reaching equal outcomes.

In 2020, Kai et al. [26] suggested an FDN for minimizing noise and distilling aspect-relevant sentiment characteristics. Researchers revealed the sensitive sentiment words in aspect had various polarities of sentiment for attention-based techniques, as well as the different aspects, could contribute noise to sentiment categorization based on aspects. A unique double-gate technique implemented the interactions among aspects and their respective contexts at a fine resolution. As a result, the FDN offers better accuracy and efficiency ranging from 1.0 to 2.0% on all ABSA work standards.

In 2019, Sixing et al. [27] presented a unified MTKFN model that leverages and contains various knowledge resources to increase ABSA effectiveness. Structure information was acquired through clause recognition and integrated in the system by constructing the representations of multiple context to compel the system. Developing a generic classification method using document sentiment labels and combining it with pre-trained specialized

layers to identify contextual signals but more precisely predict sentiment polarity was used to leverage sentiment information. Further, the selected model used conjunction information to capture the relationships between sentences and give different emotional traits. The method's effectiveness was proven by experimental findings on five publicly accessible ABSA datasets, demonstrating that many sources of information may work together to improve the model.

In 2019, LiuBo et al. [28] had determined that a sentence was made up of certain emotion clues, and its sentence clue was made up of many words. The GAN, a unique NN topology, was suggested to solve the shortcomings above. For learning the useful representations of aspect-dependent sentiment clue in GANN, a specifically created module called the GTR was employed. The relative distance among each sequence information, context word, aspect target, and semantic reliance inside a sentiment cue were all recorded simultaneously in these representations. Furthermore, a gating system was created to regulate information flow and filter noise to achieve more exact representations. Finally, to capture essential local emotion hint features and obtain feature position invariance. The experimental results have shown that GANN produced better outcomes and that the suggested approach was language-independent.

In 2019, Chao et al. [29] had discovered an attention mechanism that alternately models target-level and context-level attention to develop more effective context representation by focusing on target-level key phrases. A co-attention LSTM network was built on this foundation that learned the nonlinear representations of target and context at the same time and extracted several sentiment features in the co-attention mechanism. The proposed network with a multiple-hops co-attention model was presented to enhance the sentiment classification outcome. Furthermore, a novel location weighted function was presented to improve the performance of the co-attention mechanism by taking location information into account. Additional testing on 2 public datasets confirmed the efficacy of all suggested approaches, and the results give fresh insight for future advances of combining attention mechanisms and DNN for ABSA.

In 2019, Xingwei et al. [30] had determined an approach for learning aspect embeddings depending on the relationship among aspect-terms and aspect-categories. The restriction was successfully eased in the embeddings of aspects trained via the suggested technique as per the cosine measure. Furthermore, the misalignment constraint in the aspect embeddings was relieved using the suggested approach dubbed AAE to adapt aspect embeddings based

on the relationship among aspect-terms as well as aspect categories. To tackle the ACSA objective, the learned aspect embeddings were employed as initialization in existing models. Furthermore, the trials were carried out using SemEval datasets for the ACSA task, and the findings show the pre-trained aspect embeddings can improve sentiment analysis performance.

2.1 Research Gaps

Table 1 illustrates the review on ABSA scheme. Originally, the ALDONAR model was used in [23], which offers higher testing accuracy and maximum training accuracy; however, it needs to investigate semi-automatic methods in a sentiment analysis context. Moreover, the ReMemNN model was deployed in [24] that provides better accuracy, low Time complexity, and maximum macro-F1. Nevertheless, the proposed model was not combined with the word order. SOBA model were exploited in [25] that offer higher accuracy, best standard deviation, and greater efficiency, but need to investigate ways for enhancing the constructed ontology quality. Likewise, FDN method was determined in [26], which provides improved accuracy and the best macro-F1 score. However, we need to focus on the reasoning and understanding of implicit semantics. MTKFN scheme was portrayed in [27], which has maximum accuracy and highest Macro-F1; however, novel subsets were not correctly classified through the previous approach. In addition, GANN model was introduced in [28], which offers the highest accuracy, better macro-F1, maximum precision, and increased recall. However, the common knowledge was not integrated with the proposed model. Co-attention LSTM network was suggested in [29] that offers effectiveness and increased accuracy rate. However, a context learning function was not designed for capturing the relationship among local context words. Finally, AAE model was introduced in [30], which provides higher accuracy, but the proposed method was not comparable with the domain adaptation. Thus, the challenges were considered effectively based on the ABSA scheme in the present work.

3 Overall Description of the Aspect Based Sentiment Analysis Model

This paper aims to implement a novel ABSA scheme with three phases: (i) Preprocessing, (ii) Aspect Sentiment Extraction, and (iii) Classification. Initially, the input data is given to the Preprocessing phase, in which the tokenization, lemmatization, and stop word removal is performed. The weighted

Table 1 Reviews of conventional aspect-based sentiment analysis models: Features and Challenges

Author [Citation]	Adopted Scheme	Features	Challenges	Dataset	Results
Donatas et al. [23]	ALDONAr model	<ul style="list-style-type: none"> • Maximum training accuracy • Higher testing accuracy 	<ul style="list-style-type: none"> • Need to investigate semi-automatic methods in sentiment analysis context. 	<ul style="list-style-type: none"> • SemEval 2014 Task 4 and • ACL14 Twitter datasets 	<ul style="list-style-type: none"> • The accuracy is 80.79%
Ning et al. [24]	ReMemNN model	<ul style="list-style-type: none"> • Better accuracy • Low Time complexity • Maximum macro-F1 	<ul style="list-style-type: none"> • The proposed model was not combined with the word order. 	<ul style="list-style-type: none"> • Two different set of reviews gathered from Amazon 	<ul style="list-style-type: none"> • The classification precision is 0.750
Lisa et al. [25]	SOBA model	<ul style="list-style-type: none"> • Higher accuracy • Best standard deviation • Greater efficiency 	<ul style="list-style-type: none"> • Need to examine ways for enhancing the quality of the constructed ontology. 	<ul style="list-style-type: none"> • Evaluate our approaches on the two datasets of the SemEval-2014 Task 4 competition 	<ul style="list-style-type: none"> • The accuracy is 81.38%
Kai et al. [26]	FDN model	<ul style="list-style-type: none"> • Improved accuracy • Best macro-F1 score 	<ul style="list-style-type: none"> • Need to focus on the reasoning and understanding of implicit semantics. 	<ul style="list-style-type: none"> • Conducted the experiments on a large-scale education dataset containing students' reviews on MOOCs collected from Coursera 	<ul style="list-style-type: none"> • The F1-Score is 82.08%
Sixing et al. [27]	MTKFN model	<ul style="list-style-type: none"> • Maximum accuracy • Highest Macro-F1 	<ul style="list-style-type: none"> • Novel subsets were not correctly classified through the previous approach. 	<ul style="list-style-type: none"> • The dataset used are Twitter corpus, Lap14 and Rest14, Rest15, Rest16 and SpATSA5. 	<ul style="list-style-type: none"> • The accuracy is 82.05%
LiuBo et al. [28]	GANN model	<ul style="list-style-type: none"> • Highest accuracy • Better macro-F1 • Maximum precision • Increased recall 	<ul style="list-style-type: none"> • The common knowledge was not integrated with the proposed model. 	<ul style="list-style-type: none"> • SemEval2014 restaurant and laptop datasets 	<ul style="list-style-type: none"> • The accuracy is 92.22%
Chao et al. [29]	Coattention-LSTM network	<ul style="list-style-type: none"> • Effectiveness • Increased accuracy rate 	<ul style="list-style-type: none"> • A context learning function was not designed for capturing the relationship among local context words. 	<ul style="list-style-type: none"> • SemEval 2014 Task 41 and Twitter 	<ul style="list-style-type: none"> • The classification accuracy is 0.797
Xingwei et al. [30]	AAE method	<ul style="list-style-type: none"> • Higher accuracy 	<ul style="list-style-type: none"> • The proposed method was not comparable with the domain adaptation. 	<ul style="list-style-type: none"> • Restaurant datasets from the ABSA tasks of 410 SemEval workshops 2014 and 2016 	<ul style="list-style-type: none"> • The accuracy is 79.48%

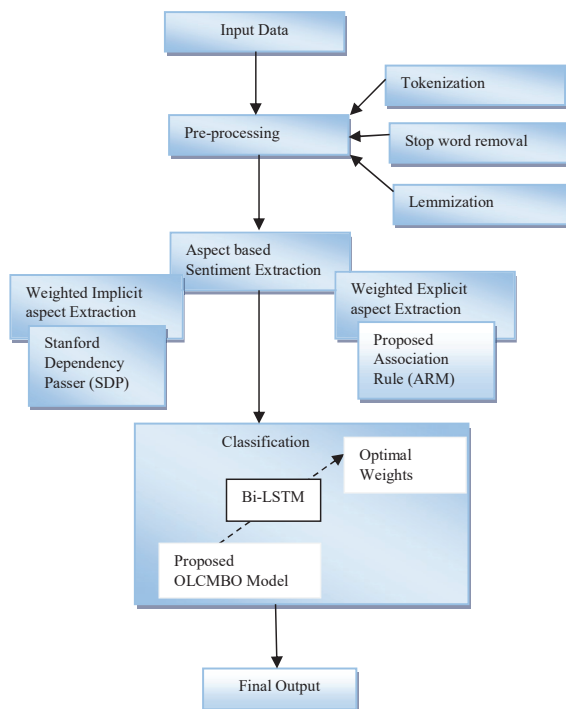


Figure 1 Overall framework of the adopted model.

implicit and explicit aspect extraction is performed in the aspect sentiment extraction phase from the preprocessed data. Moreover, the weighted implicit aspect extraction is done by SDP method, and the weighted single and multi-word extraction is done by the proposed ARM. Finally, the extracted features are subjected to the Classification phase, where Optimized Bi-LSTM is carried out. To make the classification more precise and accurate, it is planned to tune the weights of Bi-LSTM optimally. For this purpose, an OLCMBO algorithm is introduced in this work. Thus, the final output is obtained effectively. Figure 1 illustrates the framework of the proposed work.

4 Preprocessing Phase and Aspect Sentiment Extraction Phase

Preprocessing phase and Aspect sentiment extraction phase are elaborated in this section.

4.1 Preprocessing

Pre-processing is a necessary stage in the sentiment classification process. The technique that eliminates the frequent usage words which are meaningless and useless for the text classification. This reduces the corpus size without losing important information. Data pre-processing is critical because it ensures that the data is ready for detailed analysis in the most meaningful way possible. At first, the preprocessing is done under certain processes.

- Tokenization
- Lemmatization
- Stop word removal

Tokenization [31]: The procedure of converting text into tokens before transforming to vectors is known as tokenization. The tokenization technique entails breaking down large amounts of text into smaller pieces. Tokens are created from raw texts (phrases and words) during the tokenization process. As a consequence, the reviews are tokenized into words. For example, a document could be broken into sentences or phrases and then words. Finally, the tokens would aid in determining the NLP framework or interpreting the context. By examining the word sequence, tokenization aids in understanding the meaning of the text.

Imagine the sentence “He is dancing,” which is tokenized as “He, is,” and “dancing.” Tokenization may be accomplished using a variety of libraries and approaches. Additionally, tokenization is used to divide the words or phrases. Sentence tokenization is the process of dividing sentences in the same way.

Lemmatization module [32]: The goal is to find a “lexical headword” or a basic word form for a specific word. Moreover, the utilization of lemmatization preprocessing is significant for extremely inflected languages in different NLP tasks such as keyword detection and retrieval of information.

The use of lemmatization results in the least count of processed words in various form of the word being considered 1. A lemmatizer was utilized in the lemmatization module. Furthermore, this lemmatizer is derived from data containing pairings such as base word form and complete word form. A base word emerges from the vocabulary along with a group of lemmatization rules on the data. Consequently, a collection of lemmatization instances is examined, which are not included in the lexicon of base word forms but utilized for the word lemmatization.

Stop word removal [33]: Stop words are terms that are filtered out either during NLP data (text) processing. Nonetheless, the “stop words” represent ordinary words in a language. Furthermore, certain systems omit to remove stop words to facilitate phrase searches. The collection of words chosen as stop words is determined on their significance.

There are certain frequent words and less function words for certain search engines, including “the, is, at, which, and on,” and so on. The stop words create problems whenever looking for sentences like “The Who,” “The The,” or “Take That.” Other search engines eliminate some of the most popular terms and lexical words including “want” from a query to optimize efficiency.

4.2 Aspect Sentiment Extraction

The weighted implicit as well as explicit aspect extraction is performed in the aspect sentiment extraction phase from the preprocessed data. Moreover, the weighted implicit aspect extraction is done by SDP method, and the proposed ARM does the weighted single and multi-word extraction.

Weighted implicit aspect extraction: Furthermore, the SDP approach considers the relationship between opinion and aspects and extracts implicit aspects.

(i) *SDP*: Only a few researchers have examined into implicit aspects extraction for sentiment analysis based on aspects. They aimed to illustrate in this work that the relationship among aspects and opinions can aid in determining implicit aspects via capturing grammatical relations with dependency parsers. To determine the relationships that were advantageous to discovering implicit aspects, many sorts of interdependence have been used. Table 2 gave an illustration of the categories of dependency.

Table 2 Description of categories of dependency

Typed Dependency	Grammatical Relation Description
dobj-direct object	an accusatory object of the verb
amod-adjectival modifier	descriptive phrase related to a noun phrase
xcomp-open clausal complement	In a verb or an adjective is a predictive or clausal complement without its own subject.
neg-negation modifier	The relation between a negation word and the word it modifies.
Nsubj-nominal subject	It is a noun phrase which is the syntactic subject of a clause.
Det-determiner	Specific reference to a noun phrase
Advmod-adverbial modifier	Descriptive phrase related to a verb
Aux-auxiliary	A form of ‘be,’ ‘do’ or ‘have’ – action phrase

In this process, consider an instance 1 “Govt removing Motability cars from disabledpeople who can only walk 50 m and will then demonise those that have to give up work because of it.”

The SDP model provided the subsequent outcomes: “root(ROOT-0, Govt-1) acl(Govt-1, removing-2) compound(cars-4, Motability-3) dobj(removing-2, cars4) case(ppl-7, from-5) amod(ppl-7, disabled-6) nmod(removing-2, ppl-7) nsubj(walk-11, who-8) aux(walk-11,can-9) advmod(walk-11, only-10) acl:relcl(ppl-7, walk-11) dobj(walk-11, 50m-12) cc(walk-11, and-13) aux(demonise16, will-14) advmod(demonise-16,then-15) conj(walk-11,demonise-16) dobj(demonise-16, those-17) nsubj(have19, that-18) acl:relcl(those-17, have-19) mark(give-21,to-20) xcomp(have-19, give-21) compound:prt(give-21,up-22) dobj(give-21, work-23) case(it-26, because-24)mwe(because-24, of-25) nmod(work-23, it-26)”.

Direct as well as transitive dependencies were used in the process. For Example 1, an implicit feature of the tweet, particularly ‘disabled people’, was recognized using the relation amod(ppl-7, disabled-6)’. Furthermore, if a negation modifier relationship were discovered, the tweet would be changed to the opposite emotion.

Weighted explicit aspect extraction: Furthermore, the work was completed using the ARM and POS patterns to identify the explicit single and multi-word features.

(i) *Proposed ARM:* It is used to determine the most important details about a certain target. Further, the association rules are formed through analyzing data for common ‘if/then’ patterns. The most important associations are determined using the confidence and supporting criteria. Furthermore, the confidence reflects the count of times ‘if/then’ statements are confirmed right, but the supporting criteria denote the frequency with the items in the database. There are no rules produced, and the commonly used things are used.

For the supplied target entities, they used ARM [33] to determine the essential single or multi-word aspects. The ARM algorithm based on the “Apriori algorithm” was used in this experiment. The Apriori technique is used to extract the most common (essential) characteristics from a group of transactions and is split into two parts. It discovers every frequent itemset which that fulfill user-specified minimum support in the first phase from a collection of transactions. Then, it produces rules from the found frequent itemsets in the second stage. For our objective, they simply need to do the first phase, which would be to locate frequent itemsets that are candidate characteristics. Furthermore, they consider that a product feature should consist not more than 3 words. Thus, they just need to locate common itemsets with 3

Table 3 POS tags used for proposed ARM method

Descriptions	Tag
Adverb, comparative	RBR
Noun, singular or mass	NN
Verb, past participle	VBN
Noun, plural	NNS
Adjective	JJ
Proper noun, plural	NNPS
Verb, base form	VB
Determiner	DT
Verb, past tense	VBD
Verb, non-3rd person singular present	VBP
Adverb, superlative	RBS
Verb, gerund or present participle	VBG
Verb, 3rd person singular present	VBZ
Adverb	RB
Proper noun, singular	NNP

words or less in this job (this restriction was relaxed easily). In this study, the created frequent itemsets, also known as frequent candidate features, are saved to the feature set for further analysis.

The model begins by gathering review sentences, after which POS tagging is applied to each sentence, and candidates for aspects are retrieved and stemmed. A POS Tagger is an amount of software that reads text and assigns POS tag to each word, including noun, verb, adjective, and so on. Table 3 depicts the list of POS tags used for ARM Method.

In the trials, several minimum support as well as minimal confidence levels were used. During the experiments, 0.1 was found to be an excellent number for minimal support, while 0.5 was shown to be a good value for minimum confidence.

Set of rules: Aspects as well as opinions are gathered from the rules. Opinions are selected depending on the number of frequent items in the collection. The following rules are used to choose an aspect. Table 4 lists the steps of the rules for extracting aspect-sentiment information. For the aforementioned example: “The pictures are absolutely amazing”, the detected POS tags: “pictures/NNS” and “amazing/JJ” gets matches with the P3 of the Set of aspect-sentiment detection show in Table 4. The words “wonderful” and “images” here refer to the opinion and aspect, respectively. The collected aspects as well as related opinions from the POS-tagged phrases are saved for further processing in an Excel file.

Table 4 Rule based aspect extraction

Rule/ Pattern	Patterns			Example (s)
	1st Word	2nd Word	3rd Word	
P1	JJ (Adjective)	NN/NNS Noun)	–	Beautiful pictures, blurry pictures, smooth touch
P2	JJ (Adjective,	NN/NNS Noun,	NN/NNS Noun)	Simple cell phone
P3	NN/NNS (Noun	JJ Adjective)	–	Software terrible
P4	VB (Verb	NN/NNS Noun)	–	Recommended camera
P5	AVB (Adverb,	JJ Adjective, NN/NNS	NN/NNS Noun)	Very nice pictures
P6	NN (Noun	NN/NNSNoun,	JJ Adjective)	Battery life lasts
P7	NN (Noun	IN IN	NN Noun)	Quality of photo
P8	JJ (Adjective,	TO TO,	VB Verb)	Easy to use
P9	VB (Verb,	JJ Adjective)	–	Looks great
P10	VB (Verb,	VB Noun,	NN Noun)	Superb voice quality”

Opinion Words Extraction: People utilize opinion words to communicate their positive or negative feelings. They could retrieve opinion words from the review database via identifying that people commonly express their thoughts about a product feature utilizing opinion words which are positioned around the feature in the sentence utilizing the remaining frequent characteristics. Consider the following sentences: “The strap is terrible and gets in the way of parts of the camera you need to access.” In the first phrase, the feature strap is close to opinion word terrible. They can remove opinion words in the subsequent fashion as a result of this analysis:

- If a common attribute appears in a text in the review database, extract the adjective that is closest to it. If an adjective of this type is discovered, it is categorized as a word of opinion. A near adjective is a neighboring adjective which alter a noun/ phrase as it has a shared property.

From the above instance, terrible is the adjective so as to modify strap.

Moreover, the opinion word can be detected by using the semantic similarity. It is calculated as per Equation (1)

$$Sim = \frac{((S_1 \cap S_2) + (G_1 \cap G_2)) * wF}{((S_1 \cup G_1) \cup (S_2 \cup G_2))} \quad (1)$$

In Equation (1), wF_i denotes the weight assigned based on frequency of the word, S_1 and S_2 refers to the synsets after lemmatization and NLP passer, G_1 and G_2 apply text processing methods pos tagging etc for each synsets.

Here, the extracted features FE are obtained by merging the opinion and aspects.

5 Classification Via Optimized Bidirectional Long Short Term Memory

This section describes the classification via an optimized Bidirectional Long Short-Term Memory.

5.1 Optimized Bi-LSTM

The extracted features (FE) are given to the Bi-LSTM [34]. The LSTM network is an efficient approach to solve the issues of gradient desertion through implementing a linear connection and gate control unit. Thus, the LSTM network captured the strong dependence among the time-series data.

The LSTM formation includes the sequences of persistent LSTM cells. Let $H = \{h_1, h_2, h_3, \dots, h_y\}$ indicates the group of encoded IDs of source codes. Here, the RNN determined for each encoded ID h_y for $y = 1$ to n . Further, the output vector of RNN Y_y is determined in Equations (1) and (2).

$$g_y = \tanh(W_{ig}i_y + W_{gg}g_{y-1} + b_g) \quad (2)$$

$$Y_y = W_{go}g_y + b_Y \quad (3)$$

Where, W indicates a weight matrix (W_{ig} is a weight connecting input (i) to hidden layer (g)), g_y denotes the hidden state output, \tanh is hidden layer's activation function, as well as \hat{b} is a bias vector. The hidden state output is calculated by Equation (3), where the hidden state obtains the outcomes of the preceding state. Not all incoming sequences are utilised efficiently in an RNN due to the issue of gradient disappearing. The RNN is expanded to LSTM to avoid the difficulty and create a greater outcome. LSTM is implemented in Equations (4) to (8).

$$h_y = \varsigma(W_{ih}i_y + W_{gh}g_{y-1} + W_{uh}u_{y-1} + b_h) \quad (4)$$

$$J_y = \varsigma(W_{ij}i_y + W_{gj}g_{y-1} + W_{uj}u_{y-1} + b_j) \quad (5)$$

$$u_y = J_y u_{y-1} + h_y \tanh(W_{iu}i_y + W_{gu}g_{y-1} + b_u) \quad (6)$$

$$O_y = \varsigma(W_{io}i_y + W_{go}g_{y-1} + W_{uo}u_y + b_o) \quad (7)$$

$$g_y = O_y \tanh(u_y) \quad (8)$$

Where, ς indicates a sigmoid function; u, J, h , as well as O are the cell state, forget gate, input, and output, correspondingly; as well as b are biases.

Nevertheless, LSTM still has a flaw in that it only analyses the input's past context and does not evaluate any future (i.e., following) context.

They used the BiLSTM model to overcome this issue, which allows us to consider both the past and future context of source codes. The forward hidden layer as well as the reverse hidden layer are the two unique hidden layers here. Moreover, the forward hidden layer g_y^J considers the input in ascending order, i.e., $y = 1, 2, 3, \dots, Y$. Then, the backward hidden layer g_y^b and the input in descending order, i.e., $y = Y, \dots, 3, 2, 1$. At the end, g_y^J and g_y^b are combined to generate output Y_y . The Bi-LSTM model is represented in Equations (9) to (11).

$$g_y^J = \tanh(W_{ig}^J i_t + W_{gg}^J g_{y-1}^J + b_g^J) \quad (9)$$

$$g_y^b = \tanh(W_{ig}^b i_t + w_{gg}^b g_{y-1}^b + b_g^b) \quad (10)$$

$$Y_y = W_{go}^J g_y^J + W_{gY}^b g_y^b + b_Y \quad (11)$$

The bidirectional LSTM network is used as the core processing unit for code evaluation and training. The output of Bi-LSTM is denoted as $CL_{Bi-LSTM}$.

6 Weight Optimization of BI-LSTM Via Opposition Learning Cat and Mouse Based Optimization Algorithm

This section depicts the weight optimization of BI-LSTM via opposition learning cat and mouse-based optimization algorithm.

6.1 Objective Function and Solution Encoding

The Bi-LSTM weights are optimally tuned via an adopted OLCMBO method. Figure 2 represents the input solution of the adopted SSSO scheme. Where, W specifies the weights of OLCMBO, and N indicates the total count of weights. The objective function is determined in Equation (12). Here, (err) specifies the error function in OLCMBO.

$$Obj = \min(err) \quad (12)$$



Figure 2 Solution encoding.

6.2 Proposed OLCMBO Model

Despite the fact that the traditional CMBO [35] model is utilised in optimum designs and engineering sciences, wherein chosen variables should be carefully selected to improve performance of the device; still, it will not address another optimization problem. For overcoming these issues, the OLCMBO model is suggested in this paper. Normally, self-enhancement has been shown to be possible in conventional optimization techniques.

The proposed concept deploys the OBL [36] that is modelled for generating the opposite solutions. The points and its opposite points are simultaneously calculated to carry on with the best one. The OBL based initialization guarantees improved convergence rate, thus attaining enhanced solutions quickly.

The CMBO is a population-based method inspired by the natural behaviors of a cat attacking a mouse and the mouse escaping to a refuge. The suggested method divides the search agents into 2 categories: cats and mice, which wander around randomly in the issue search space. In two phases, the proposed method updates population members. The 1st phase models the cats movement toward mice, while the second part models mice fleeing to safe havens to save their lives.

From a mathematical standpoint, every member of the population represents a possible solution to the issues. As a result, each person in the population is a vector, and the values of the vectors define the variables in the issue. The OLCMBO model may be found below. The initial population of M search agents is initialized. The parameters of M, M_c, M_v, T are initialized. Here, N is the count of members in population matrix Z .

The population of this model is examined by a matrix known as the population matrix as per Equation (13).

$$Z = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_M \end{bmatrix}_{M*v} = \begin{bmatrix} Z_{1,1} & \cdots & Z_{1,d} & \cdots & Z_{1,v} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ Z_{j,1} & \cdots & Z_{j,d} & \cdots & Z_{j,v} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ Z_{M,1} & \cdots & Z_{M,d} & \cdots & Z_{M,v} \end{bmatrix}_{M*v} \quad (13)$$

Here, $Z_{j,d}$ denotes the d^{th} problem variable attained through the j^{th} search agent, Z denotes the population matrix of CMBO, Z_j is the j^{th} search agent, M specifies the count of population members, and v refers to the count of problem variables. The obtained values for the objective function are given

in Equation (14).

$$F = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_M \end{bmatrix}_{M*1} \tag{14}$$

In Equation (14), F denotes the objective function values vector as well as F_j indicates the j^{th} search agent objective function value. Using Equations (15) to (16), update the sorted population matrix Z^R . Here, i^{th} population of sorted population matrix is denoted as $z_{j,d}^R$. In addition, F^R is the sorted objective function based vector.

$$Z^R = \begin{bmatrix} Z_1^R \\ Z_2^R \\ \vdots \\ Z_M^R \end{bmatrix}_{M*v} = \begin{bmatrix} z_{1,1}^R & \cdots & z_{1,d}^R & \cdots & z_{1,v}^R \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ z_{j,1}^R & \cdots & z_{j,d}^R & \cdots & z_{j,v}^R \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ z_{M,1}^R & \cdots & z_{M,d}^R & \cdots & z_{M,v}^R \end{bmatrix}_{M*v} \tag{15}$$

$$F^R = \begin{bmatrix} F_1^R & \min(F) \\ F_2^R & \min(F) \\ \vdots & \vdots \\ F_M^R & \min(F) \end{bmatrix}_{M*1} \tag{16}$$

Where, $z_{j,d}^R$ denotes the value for the d^{th} problem variable attained through the j^{th} search agent, Z^R indicates the sorted population matrix depending on objective function value, Z_j^R refers to the j^{th} member of sorted population matrix, as well as F^R denotes the sorted vector of an objective function. Using Equation (17), the mice population is chosen

$$I = \begin{bmatrix} I_1 = X_1^R \\ \vdots \\ I_j = X_j^R \\ \vdots \\ I_{M_v} = X_{M_v}^R \end{bmatrix}_{N_m*m} = \begin{bmatrix} z_{1,1}^R & \cdots & z_{1,d}^R & \cdots & z_{1,v}^R \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ z_{j,1}^R & \cdots & z_{j,d}^R & \cdots & z_{j,v}^R \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ z_{M_v,1}^R & \cdots & z_{M_v,d}^R & \cdots & z_{M_v,v}^R \end{bmatrix}_{M_v*v} \tag{17}$$

Using Equation (18), the cat population is selected.

$$\begin{aligned}
 C &= \begin{bmatrix} C_1 = X_{M_v+1}^R \\ \vdots \\ C_k = X_{M_v+k}^R \\ \vdots \\ C_{M_a} = X_{M_v+M_a}^R \end{bmatrix}_{M_a * v} \\
 &= \begin{bmatrix} z_{M_v+1,1}^R & \cdots & z_{M_v+1,d}^R & \cdots & z_{M_v+1,v}^R \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ z_{M_v+k,1}^R & \cdots & z_{M_v+k,d}^R & \cdots & z_{M_v+k,v}^R \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ z_{M_v+M_a,1}^R & \cdots & z_{M_v+M_a,d}^R & \cdots & z_{M_v+M_a,v}^R \end{bmatrix}_{M_a * v} \quad (18)
 \end{aligned}$$

Here, I , M_v , I_j , C , M_a , C_k points to the mice population, count of mice, k^{th} mice, cat population, count of cats and j^{th} cat, respectively.

For updating the search parameters, the change of location of cats is modelled in the first phase based on natural cat behaviour and moment towards mice. Further, this phase of the suggested CMBO's update is described mathematically by Equations (19) to (21)

$$C_k^{new} : C_{k,d}^{new} = a_{k,d} + r * (v_{l,d} - V * a_{k,d}) \& k = 1 : M_a, \quad (19)$$

$$d = 1 : v, l \in 1 : M_v$$

$$\text{Here, } V = \text{round}(1 + \text{rand}) \quad (20)$$

$$C_k = \begin{cases} C_k^{new}, & |F_k^{a,new} < F_k^a, \\ C_k, & |else \end{cases} \quad (21)$$

Here, $C_{k,d}^{new}$ denotes the novel value for the d^{th} problem variable attained through the k^{th} cat, $F_k^{a,new}$ denotes the objective function value depending on new status of the k^{th} cat, C_k^{new} refers to the novel status of the k^{th} cat, r refers to a random number among $[0, 1]$, and $v_{l,d}$ specifies the d^{th} dimension of the l^{th} mouse.

The escape of mice to refuge is simulated in the planned OLCMBO's second phase. In OLCMBO, each mouse is believed to have a random

havenin these havens, and mice seek shelter. The havens placements in the search space are generated at random by patterning the positions of various algorithm elements. As per the proposed OLCMBO model, this phase of the mice position updating is given in Equations (22) to (24).

$$B_j = p_{j,d} = z_{Vd} \& j = 1 : M_v, d = 1 : v, q \in 1 : M \quad (22)$$

$$I_j^{new} : v_{j,d}^{new} = v_{j,d} + r * (p_{j,d} - V * v_{j,d}) * Sign(F_j^v - F_j^B) + Levy(\beta) \\ \text{where, } j = 1 : M_v, d = 1 : v \quad (23)$$

$$I_j = \begin{cases} I_j^{new} & | F_j^{v,new} < F_j^v \\ I_j & | else \end{cases} \quad (24)$$

Here, $Levy(\beta)$ indicates the levy updation function, r is calculated using the tent map, B_j points to the haven for the j^{th} mouse, F_j^B denotes its objective function value, I_j^{new} indicates the novel status of the j^{th} mouse, and $F_j^{v,new}$ is the objective function value, respectively.

The algorithm begins the next iteration when all members of the algorithm population were updated, and the iterations of the method proceed until the stop condition is achieved based on Equations (17) to (24). A specified count of iterations or a defined satisfactory error among attained solutions in successive iterations could be used to terminate optimization processes. Furthermore, the algorithm might well be stopped when a particular amount of time has passed. The OLCMBO returns the best found quasi-optimal solution when all iterations have been completed and the method has been fully implemented on the optimization issue. The pseudocode is given in Algorithm 1.

The solutions are randomly chosen, the best position is 33.18969 and when the solution get updated then the next best updated position is 13.8374 and the overall fitness is calculated as 1.021

7 Results and Discussions

Here, in this section we discuss simulation procedure and the results.

7.1 Simulation Procedure

The adopted Bi-LSTM+OLCMBO with ABSA model was executed in **PYTHON** and their outcomes were confirmed. Furthermore, the performance of the presented Bi-LSTM+OLCMBO scheme was computed over the

Algorithm 1 Proposed OLCMBO method

Begin OLCMBO
Objective function, constraints, and variables.
Group of search agents (M) and iterations (T).
Create an initial population matrix randomly.
Determine the objective function.

for $t = 1 : T$
 The population matrix depending on objective function value in Equations (15) and (16).
 Choose mice population I in Equation (17)
 Choose cats population C in Equation (18)
 Phase 1: update status of cats.
 for $k = 1 : M_a$
 Inform status of the k^{th} cat in Equations (19) to (21)
 end for
 Phase 2: update status of mice.
 for $j = 1 : M_v$
 Generate haven for the j^{th} mouse in Equation (22).
 Proposed status Update of the j^{th} mouse in Equations (23) and (24).
 end for
end for

Output best solution
End CMBO

conventional schemes such as Bi-LSTM+SSO [37], Bi-LSTM+CMBO [35], Bi-LSTM+PRO [38], and Bi-LSTM+SMO [39], correspondingly. In addition, the performance was computed by altering the learning percentage from 40, 50, 60, 70 for different performance measures including precision, sensitivity, accuracy, specificity, FNR, F-measure, MCC, FPR, and NPV, respectively.

7.2 Dataset Description

The dataset was collected from [41]. “This is a list of over 34,000 consumer reviews for Amazon products like the Kindle, Fire TV Stick, and more provided by Datafiniti’s Product Database. The dataset includes basic product information, rating, review text, and more for each product”. The testing data for learning percentage 60 is (2284*2), while the training data is (3423*4). The testing data for learning percentage 70 is (1712*4), while the training data is (3993*4). The testing data for learning percentage 80 is (1141*4), while the training data is (4564*4). The testing data for learning percentage 90 is (571*4), while the training data is (5731*4).

7.3 Performance Analysis

The performance analysis of the presented Bi-LSTM+OLCMBO scheme is computed over the existing schemes like Bi-LSTM+SSO, Bi-LSTM+CMBO, Bi-LSTM+PRO, and Bi-LSTM+SMO, correspondingly in terms of certain measures and it is illustrated in Figures 3 to 5. Moreover, the positive measures such as precision, sensitivity, accuracy, specificity are illustrated in Figure 3. Likewise, the adopted Bi-LSTM+OLCMBO scheme attains higher accuracy (~ 0.9) for learning percentage 70 than the other existing schemes like Bi-LSTM+SSO, Bi-LSTM+CMBO, Bi-LSTM+PRO, and Bi-LSTM+SMO, correspondingly in Figure 3(b). This proves the accuracy of adopted model is superior to the traditional models. Further, the specificity of the adopted Bi-LSTM+OLCMBO scheme for learning percentage 40 is 27.27%, 36.36%, 13.63%, and 18.18% superior to the existing schemes like Bi-LSTM+SSO, Bi-LSTM+CMBO, Bi-LSTM+PRO, and Bi-LSTM+SMO, respectively as shown in Figure 3(a). Likewise, the adopted

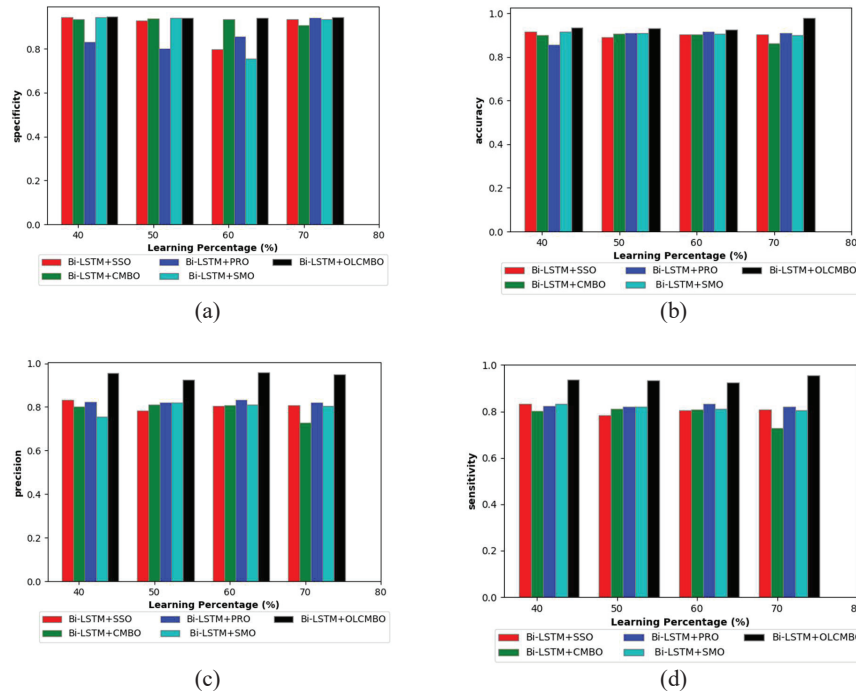


Figure 3 Performance analysis of the adopted scheme to the traditional approaches for (a) specificity (b) accuracy (c) precision (d) sensitivity.

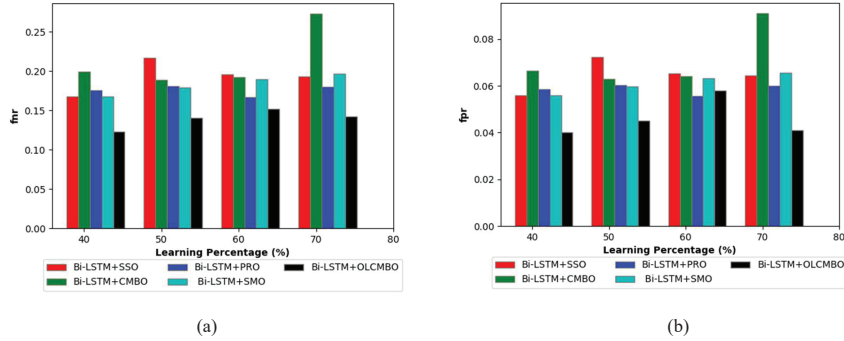


Figure 4 Performance analysis of the adopted scheme to the traditional approaches for (a) FNR (b) FPR.

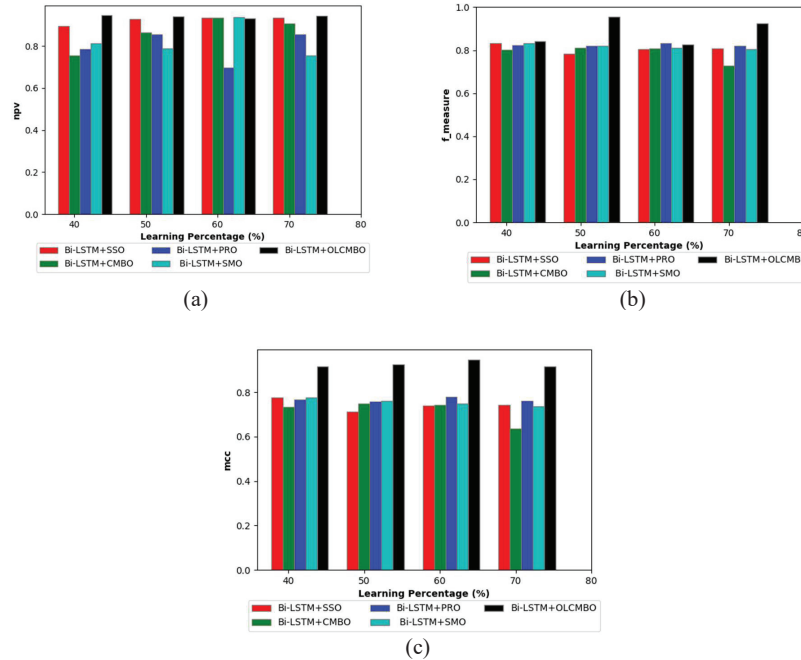


Figure 5 Performance analysis of the adopted scheme to the traditional approaches for (a) NPV (b) F-measure (c) MCC.

Bi-LSTM+OLCMBO scheme attains higher sensitivity (~ 0.99) for learning percentage 50 than the other existing schemes in Figure 3(d). Further, the proposed Bi-LSTM+OLCMBO scheme has shown higher precision value with better performance than other conventional models at learning percentage 60

in Figure 3(c). This analysis outcome has proven the impact of Bi-LSTM classifier that gets trained with the appropriate features. For weighted implicit aspect extraction, the SDP technique is utilized, while for weighted explicit aspect extraction, the recommended ARM is employed. The weights of Bi-LSTM will be optimized in order to improve classification accuracy and precision. An Opposition Learning Cat and Mouse-Based Optimization (OLCMBO) Algorithm is employed for this purpose. Further, as the Bi-LSTM weights are tuned optimally, the proposed Bi-LSTM+OLCMBO technique paved the way for better results in ABSA model with lower error.

The negative measures like FPR, and FNR [46] of the adopted Bi-LSTM+OLCMBO scheme to the traditional schemes like Bi-LSTM+SSO, Bi-LSTM+CMBO, Bi-LSTM+PRO, and Bi-LSTM+SMO, respectively is represented in Figure 4. In addition, the variations in learning percentage demonstrate the difference in the performance. The performance has proven that the adopted work has converged with the objective (minimization of error). In addition, the proposed Bi-LSTM+OLCMBO model proves the less FPR value as the better performance than the conventional models at learning percentage 60 in Figure 4(b). Less FNR value of proposed Bi-LSTM+OLCMBO model has proven that the model is less prone to error that direct to precise outcomes at learning percentage 70 in Figure 4(a).

Figure 5 represents the other measures analysis like MCC, NPV, and F-measure of the proposed Bi-LSTM+OLCMBO model over other conventional schemes. Similarly, the F-measure of adopted Bi-LSTM+OLCMBO model for learning percentage 50 in Figure 5(b) is superior to other traditional scheme like Bi-LSTM+SSO, Bi-LSTM+CMBO, Bi-LSTM+PRO, and Bi-LSTM+SMO, respectively. Likewise, the adopted Bi-LSTM+OLCMBO model attains maximum NPV (~ 1.0) for learning percentage 60 than other extant schemes in Figure 5(a). From the graph, it is clearly shown that the MCC of the adopted Bi-LSTM+OLCMBO model attains a higher value for learning percentage 70; however, the compared existing models attain lower values as per Figure 5(c). The SDP technique is used for weighted implicit aspect extraction, whereas the suggested ARM is used for weighted explicit aspect extraction. To improve classification accuracy and precision, the weights of Bi-LSTM will be tuned. This is accomplished using the Opposition Learning Cat and Mouse-Based Optimization (OLCMBO) Algorithm. Therefore, the performance of presented Bi-LSTM+OLCMBO model has shown its improvement over other traditional models.

Table 5 Statistical analysis with respect to accuracy: proposed vs. conventional models

Methods	Best	Worst	Mean	Median	Standard Deviation
Bi-LSTM+CMBO [35]	0.01417	0.108581	0.069912	0.078448	0.039842
Bi-LSTM+SSO [37]	0.013477	0.103327	0.066412	0.074422	0.037727
Bi-LSTM+SMO [39]	0.012707	0.098074	0.063018	0.070645	0.035991
Bi-LSTM+PRO [38]	0.012052	0.093842	0.059927	0.066907	0.034304
Proposed Bi-LSTM+OLCMBO	0.011359	0.087977	0.056385	0.063102	0.0322

7.4 Statistical Analysis

The statistical analysis of the presented Bi-LSTM+OLCMBO approach is computed to the existing scheme based on accuracy measure is represented in Table 5. In Nature, the meta-heuristic algorithms are stochastic; thus, the algorithms are performed several times for determining the achievement of the defined objective. The mean performance of the adopted Bi-LSTM+OLCMBO approach holds better results than the traditional schemes such as Bi-LSTM+SSO, Bi-LSTM+CMBO, Bi-LSTM+PRO, and Bi-LSTM+SMO, correspondingly. The best case scenario proves an enhancement of proposed Bi-LSTM+OLCMBO model attains (~ 0.011) with accurate results than the other traditional models like Bi-LSTM+SSO, Bi-LSTM+CMBO, Bi-LSTM+PRO, and Bi-LSTM+SMO, correspondingly. The proposed Bi-LSTM+OLCMBO model has proved its improvement almost in all cases. For weighted implicit aspect extraction, the SDP technique is utilized, whereas for weighted explicit aspect extraction, the recommended ARM is employed. The weights of Bi-LSTM will be fine-tuned to improve classification accuracy and precision. The Opposition Learning Cat and Mouse-Based Optimization (OLCMBO) Algorithm is used to accomplish this. Therefore, the development of the presented Bi-LSTM+OLCMBO approach has been validated effectively.

7.5 Analysis Based on Features

The analysis based on features work for proposed and conventional features in terms of certain measures is illustrated in Table 6. In addition, the proposed Bi-LSTM+OLCMBO model holds better accuracy than

Table 6 Analysis based on features

Measures	Proposed Model Without Optimization			Semantic Score	Proposed Bi-LSTM+OLCMBRO Model
	SDP	ARM			
Accuracy	0.917576	0.793333	0.917576	0.917576	0.95961
Sensitivity	0.964263	0.80756	0.964263	0.964263	1
Specificity	0.036145	0.333333	0.036145	0.036145	0.946785
Precision	0.949717	0.975104	0.949717	0.949717	0.856459
F-Measure	0.956935	0.883459	0.956935	0.956935	0.92268
MCC	0.00048	0.060468	0.00048	0.00048	0.90049
NPV	0.050847	0.050847	0.050847	0.050847	1
FPR	0.963855	0.666667	0.963855	0.963855	0.053215
FNR	0.035737	0.19244	0.035737	0.035737	0

other feature comparison including proposed model without Optimization, SDP, ARM, and Semantic score, respectively. Further, the proposed Bi-LSTM+OLCMBRO model holds lower FNR with better performance than other feature comparison including proposed model without Optimization, SDP, ARM, and Semantic score, respectively. This has proved that with proposed Bi-LSTM+OLCMBRO model helps to analysis more accurately, whereas other traditional models shows its poor performance with the proposed concept. This absolutely evolves that the proposed combination is fit for ABSA model.

7.6 Analysis on Classifiers

The analysis of adopted work based on different classifiers is represented in Table 7. In addition, proposed Bi-LSTM+OLCMBRO model hold superior precision to other feature comparison including LSTM, RNN, GRU, DBN, ALDONAr, and ReMemNN, respectively. Further, the proposed Bi-LSTM+OLCMBRO model has shown lower FPR with better performance when computed to other extant approaches including LSTM, RNN, GRU, DBN, ALDONAr, and ReMemNN, respectively. The accuracy of the proposed methodology is 5.42%, 7.42%, 4.58%, 4.03%, 5.27%, and 1.09% better than the other methods like LSTM, RNN, GRU, DBN, ALDON, ReMem. The SDP approach is used for weighted implicit aspect extraction, and the suggested ARM is used for weighted explicit aspect extraction. It is planned to optimize the weights of Bi-LSTM in order to improve

Table 7 Analysis of proposed work with different classifiers

	LSTM	RNN	GRU	DBN	ALDONAr	ReMemNN	Proposed Bi-LSTM+ OLCMBO
Measures	[42]	[43]	[44]	[45]	[23]	[24]	Model
Accuracy	0.905455	0.922424	0.893333	0.911515	0.917576	0.850303	0.95961
Sensitivity	0.9515	0.970006	0.938736	0.957881	0.964263	0.851308	1
Specificity	0.036145	0.024096	0.036145	0.036145	0.036145	0.831325	0.946785
Precision	0.949077	0.949407	0.94842	0.949399	0.949717	0.989614	0.856459
F-measure	0.950287	0.959596	0.943554	0.953621	0.956935	0.915266	0.92268
MCC	-0.01265	-0.00759	-0.02312	-0.00652	0.00048	0.385845	0.90049
NPV	0.037975	0.040816	0.030303	0.043478	0.050848	0.228477	1
FPR	0.963855	0.975904	0.963855	0.963855	0.963855	0.168675	0.053215
FNR	0.0485	0.029994	0.061264	0.042119	0.035737	0.148692	0

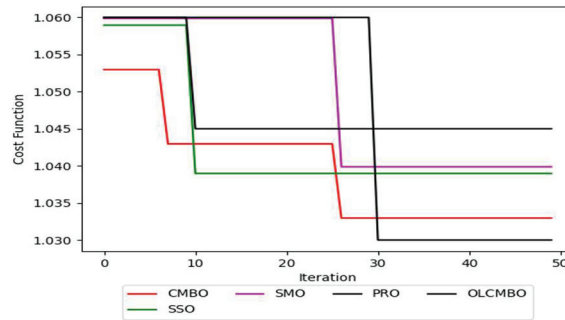


Figure 6 Convergence analysis of proposed approach and existing approaches.

classification accuracy and precision. For this purpose, an Opposition Learning Cat and Mouse-Based Optimization (OLCMBO) Algorithm. Therefore, the betterment of the proposed Bi-LSTM+OLCMBO model has attained effectively.

7.7 Convergence Analysis

The convergence of the presented OLCMBO approach to the traditional schemes is examined by varying the iteration count from 0, 10, 20, 30, 40, and 50, correspondingly. Figure 6 represents the convergence analysis of presented scheme over the traditional schemes. The proposed OLCMBO approach attains the minimum cost function as per the defined objectives in Equation (12). As the count of iteration rises, the cost function of OLCMBO algorithm gets minimized. Moreover, the cost function of

proposed OLCMBO model had a fall in between the range 28th to 30th iteration. The cost function of the adopted OLCMBO scheme provides lower constant value (1.030) from 30th to 50th iteration than other existing models like CMBO, SSO, SMO, and PRO, correspondingly. Consequently, it is shown clearly that the adopted OLCMBO approach had attained the lower cost function with superior outcomes.

8 Discussion

The Stanford Dependency Passer (SDP) approach is used for weighted implicit aspect extraction, and the suggested Association Rule Mining method is used for weighted explicit aspect extraction (ARM). Reviews for Amazon products like the Kindle, Fire TV Stick, and Datafiniti's Product Database is considered in this work. LSTM still has a flaw in that it only analyses the input's past context and does not evaluate any future context. They used the BiLSTM model to overcome this issue, which allows us to consider both the past and future context of source codes. The weighted single and multi-word extraction is done by the proposed ARM. It is planned to optimize the weights of Bi-LSTM in order to improve classification accuracy and precision. For this purpose, an Opposition Learning Cat and Mouse-Based Optimization (OLCMBO) Algorithm. Thus, the betterment of the proposed methodology proved over the traditional method and it is clearly described in the result section.

9 Conclusion

This paper has implemented a novel ABSA model with three phases: (i) Pre-processing (ii) Aspect Sentiment Extraction (iii) Classification. Initially, the input data was given to the Preprocessing phase, in which the tokenization, stemming, lemmatization, and stop word removal was performed. From the preprocessed data, the weighted implicit as well as weighted explicit aspect extraction was performed in the aspect sentiment extraction phase. Moreover, the weighted implicit aspect extraction was done by SDP method, and the weighted explicit extraction was done through proposed ARM. Subsequently, the extracted features were provided to the classification phase in which the Optimized Bi-LSTM was used. For making the classification more accurate and precise, it was planned to tune the weights of Bi-LSTM optimally. For this purpose, an OLCMBO Algorithm was introduced in this work.

At the end, the outcomes of the presented approach was calculated to the extant approaches in with respect to different measures like F1-measure, specificity, NPV, accuracy, FNR, sensitivity, precision, FPR, and MCC, respectively. From the graph, the adopted Bi-LSTM+OLCMBO scheme attains maximum accuracy (~ 0.9) for learning percentage 40% than the other existing schemes like Bi-LSTM+SSO, Bi-LSTM+CMBO, Bi-LSTM+PRO, and Bi-LSTM+SMO, correspondingly. Less FNR value of proposed Bi-LSTM+OLCMBO model has proven that the model was less prone to error that direct to precise outcomes at learning percentage 70%. The best case scenario proves an enhancement of proposed Bi-LSTM+OLCMBO model attains (~ 0.011) with accurate results than the other traditional models like Bi-LSTM+SSO, Bi-LSTM+CMBO, Bi-LSTM+PRO, and Bi-LSTM+SMO, correspondingly.

References

- [1] Feiyang Ren, Liangming Feng, Sheng Cheng, “DNet: A lightweight and efficient model for aspect based sentiment analysis”, *Expert Systems with Applications* 19 March 2020 Volume 151 (Cover date: 1 August 2020) Article 113393.
- [2] Reinald Kim, Amplayo Seanie, Lee Min Song, “Incorporating product description to sentiment topic models for improved aspect-based sentiment analysis”, *Information Sciences* 1 May 2018 Volumes 454–455 (Cover date: July 2018) Pages 200–215.
- [3] C. R. Aydin and T. Güngör, “Combination of Recursive and Recurrent Neural Networks for Aspect-Based Sentiment Analysis Using Inter-Aspect Relations,” in *IEEE Access*, vol. 8, pp. 77820–77832, 2020, doi: 10.1109/ACCESS.2020.2990306.
- [4] Z. Kastrati, A. S. Imran and A. Kurti, “Weakly Supervised Framework for Aspect-Based Sentiment Analysis on Students’ Reviews of MOOCs,” in *IEEE Access*, vol. 8, pp. 106799–106810, 2020, doi: 10.1109/ACCESS.2020.3000739.
- [5] B. Zhang, X. Li, X. Xu, K.-C. Leung, Z. Chen and Y. Ye, “Knowledge Guided Capsule Attention Network for Aspect-Based Sentiment Analysis,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2538–2551, 2020, doi: 10.1109/TASLP.2020.3017093.
- [6] S. Ali, G. Wang and S. Riaz, “Aspect Based Sentiment Analysis of Ridesharing Platform Reviews for Kansei Engineering,” *IEEE*

- Access, vol. 8, pp. 173186–173196, 2020, doi: 10.1109/ACCESS.2020.3025823.
- [7] M. Shams, N. Khoshavi and A. Baraani-Dastjerdi, “LISA: Language-Independent Method for Aspect-Based Sentiment Analysis,” *IEEE Access*, vol. 8, pp. 31034–31044, 2020, doi: 10.1109/ACCESS.2020.2973587.
- [8] Alamanda, M.S. Aspect-based sentiment analysis search engine for social media data. *CSIT* 8, 193–197 (2020). <https://doi.org/10.1007/s40012-020-00295-3>
- [9] Karagoz, P., Kama, B., Ozturk, M. et al. A framework for aspect based sentiment analysis on turkish informal texts. *J Intell Inf Syst* 53, 431–451 (2019). <https://doi.org/10.1007/s10844-019-00565-w>
- [10] A. Ishaq, S. Asghar and S. A. Gillani, “Aspect-Based Sentiment Analysis Using a Hybridized Approach Based on CNN and GA,” *IEEE Access*, vol. 8, pp. 135499–135512, 2020, doi: 10.1109/ACCESS.2020.3011802.
- [11] Z. Jia, X. Bai and S. Pang, “Hierarchical Gated Deep Memory Network With Position-Aware for Aspect-Based Sentiment Analysis,” *IEEE Access*, vol. 8, pp. 136340–136347, 2020, doi: 10.1109/ACCESS.2020.3011318.
- [12] N. Li, C. -Y. Chow and J. -D. Zhang, “SEML: A Semi-Supervised Multi-Task Learning Framework for Aspect-Based Sentiment Analysis,” *IEEE Access*, vol. 8, pp. 189287–189297, 2020, doi: 10.1109/ACCESS.2020.3031665.
- [13] Devi Sri Nandhini, M., Pradeep, G. A Hybrid Co-occurrence and Ranking-based Approach for Detection of Implicit Aspects in Aspect-Based Sentiment Analysis. *SN COMPUT. SCI.* 1, 128 (2020). <https://doi.org/10.1007/s42979-020-00138-7>
- [14] Al-Smadi, M., Talafha, B., Al-Ayyoub, M. et al. Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews. *Int. J. Mach. Learn. & Cyber.* 10, 2163–2175 (2019). <https://doi.org/10.1007/s13042-018-0799-4>
- [15] Ikram, M.T., Afzal, M.T. Aspect based citation sentiment analysis using linguistic patterns for better comprehension of scientific knowledge. *Scientometrics* 119, 73–95 (2019). <https://doi.org/10.1007/s11192-019-03028-9>
- [16] J. Zhou, S. Jin and X. Huang, “ADeCNN: An Improved Model for Aspect-Level Sentiment Analysis Based on Deformable CNN and

- Attention,” *IEEE Access*, vol. 8, pp. 132970–132979, 2020, doi: 10.1109/ACCESS.2020.3010802.
- [17] S. M. Al-Ghuribi, S. A. Mohd Noah and S. Tiun, “Unsupervised Semantic Approach of Aspect-Based Sentiment Analysis for Large-Scale User Reviews,” *IEEE Access*, vol. 8, pp. 218592–218613, 2020, doi: 10.1109/ACCESS.2020.3042312.
- [18] H. Liu, I. Chatterjee, M. Zhou, X. S. Lu and A. Abusorrah, “Aspect-Based Sentiment Analysis: A Survey of Deep Learning Methods,” *IEEE Transactions on Computational Social Systems*, vol. 7, no. 6, pp. 1358–1375, Dec. 2020, doi: 10.1109/TCSS.2020.3033302.
- [19] W. Meng, Y. Wei, P. Liu, Z. Zhu and H. Yin, “Aspect Based Sentiment Analysis With Feature Enhanced Attention CNN-BiLSTM,” *IEEE Access*, vol. 7, pp. 167240–167249, 2019, doi: 10.1109/ACCESS.2019.2952888.
- [20] K. Xu, H. Zhao and T. Liu, “Aspect-Specific Heterogeneous Graph Convolutional Network for Aspect-Based Sentiment Classification,” *IEEE Access*, vol. 8, pp. 139346–139355, 2020, doi: 10.1109/ACCESS.2020.3012637.
- [21] S. Rida-E-Fatima et al., “A Multi-Layer Dual Attention Deep Learning Model With Refined Word Embeddings for Aspect-Based Sentiment Analysis,” *IEEE Access*, vol. 7, pp. 114795–114807, 2019, doi: 10.1109/ACCESS.2019.2927281.
- [22] Liu, N., Shen, B., Zhang, Z. et al. Attention-based Sentiment Reasoner for aspect-based sentiment analysis. *Hum. Cent. Comput. Inf. Sci.* 9, 35 (2019). <https://doi.org/10.1186/s13673-019-0196-3>
- [23] Donatas Meškelė, Flavius Frasinca, “ALDONAr: A hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalized domain ontology and a regularized neural attention model”, *Information Processing & Management* 31 January 2020 Volume 57, Issue 3 (Cover date: May 2020) Article 102211.
- [24] Ning Liu, Bo Shen, “ReMemNN: A novel memory neural network for powerful interaction in aspect-based sentiment analysis”, *Neurocomputing* 8 February 2020 Volume 395 (Cover date: 28 June 2020) Pages 66–77.
- [25] Lisa Zhuang, Kim Schouten, Flavius Frasinca, “SOBA: Semi-automated Ontology Builder for Aspect-based sentiment analysis”, *Journal of Web Semantics* 11 December 2019 Volume 60 (Cover date: January 2020) Article 100544.

- [26] Kai Shuang, Qianqian Yang, Mengyu Gu, “Feature distillation network for aspect-based sentiment analysis”, *Information Fusion* 16 March 2020 Volume 61 (Cover date: September 2020) Pages 13–23.
- [27] Sixing Wu, Yuanfan Xu, Xing Li, “Aspect-based sentiment analysis via fusing multiple sources of textual knowledge”, *Knowledge-Based Systems* 25 July 2019 Volume 183 (Cover date: 1 November 2019) Article 104868.
- [28] Ning Liu, Bo Shen, “Aspect-based sentiment analysis with gated alternate neural network”, *Knowledge-Based Systems* 2 September 2019 Volume 188 (Cover date: 5 January 2020) Article 105010.
- [29] Chao Yang, Hefeng Zhang, Keqin Li, “Aspect-based sentiment analysis with alternating coattention networks”, *Information Processing & Management* 21 January 2019 Volume 56, Issue 3 (Cover date: May 2019) Pages 463–478.
- [30] Xingwei Tan, Yi Cai, Qing Li, “Improving aspect-based sentiment analysis via aligning aspect embedding”, *Neurocomputing* 12 December 2019 Volume 383 (Cover date: 28 March 2020) Pages 336–347.
- [31] A F Hidayatullah and M R Ma’arif, “Pre-processing Tasks in Indonesian Twitter Messages”, *Journal of Physics: Conference Series*, 2017.
- [32] Skorkovská, Lucie. Application of Lemmatization and Summarization Methods in Topic Identification Module for Large Scale Language Modeling Data Filtering. 7499. 10.1007/978-3-642-32790-2_23, 2012.
- [33] Zainuddin, N., Selamat, A., Ibrahim, R. Hybrid sentiment classification on twitter aspect-based sentiment analysis. *Appl Intell* 48, 1218–1232 (2018). <https://doi.org/10.1007/s10489-017-1098-6>
- [34] Rahman, Md, Watanobe, Yutaka, Nakamura, Keita. (2021). A Bidirectional LSTM Language Model for Code Evaluation and Repair. *Symmetry*. 13. 247. 10.3390/sym13020247.
- [35] Dehghani, Mohammad, Štípan Hubálovskí, and Pavel Trojovskí. 2021. “Cat and Mouse Based Optimizer: A New Nature-Inspired Optimization Algorithm” *Sensors* 21, no. 15: 5214. <https://doi.org/10.3390/s21155214>.
- [36] Chakraborty, F., Roy, P.K., Nandi, D. Oppositional elephant herding optimization with dynamic Cauchy mutation for multilevel image thresholding. *Evol. Intel.* 12, 445–467 (2019). <https://doi.org/10.1007/s12065-019-00238-1>
- [37] Fouad, Ahmed. (2015). Social Spider Optimization Algorithm. 10.13140/RG.2.1.4314.5361.

- [38] Seyyed Hamid Samareh Moosavi, Vahid Khatibi Bardsiri, “Poor and rich optimization algorithm: A new human-based and multi populations algorithm”, *Engineering Applications of Artificial Intelligence*, Volume 86, (Cover date: November 2019), Pages 165–181, 26 September 2019.
- [39] Sharma, Harish, Garima Hazrati, and Jagdish Chand Bansal. “Spider monkey optimization algorithm.” *Evolutionary and swarm intelligence algorithms*. Springer, Cham, 2019. 43–59.
- [40] Yadav, Rohan Kumar, et al. “Positionless aspect based sentiment analysis using attention mechanism.” *Knowledge-Based Systems* 226 (2021): 107136.
- [41] <https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products>.
- [42] Haoran Yan, Yi Qin, Haizhou Chen, “Long-term gear life prediction based on ordered neurons LSTM neural networks”, *Measurement* 11 July 2020 Volume 165 (Cover date: 1 December 2020) Article 108205.
- [43] Ling-Jing Kao, Chih Chou Chiu, “Application of integrated recurrent neural network with multivariate adaptive regression splines on SPC-EPC process”, *Journal of Manufacturing Systems*, vol. 57, pp. 109–118, 2020.
- [44] Shile Zhang, Mohamed Abdel-Aty, Ou Zheng, “Modeling pedestrians’ near-accident events at signalized intersections using gated recurrent unit (GRU)”, *Accident Analysis & Prevention* 28 October 2020 Volume 148 (Cover date: December 2020) Article 105844.
- [45] H.Z. Wang, G.B. Wang, G.Q. Li, J.C. Peng, and Y.T. Liu, “Deep belief network based deterministic and probabilistic wind speed forecasting approach”, *Applied Energy*, vol. 182, pp. 80–93, 2016.
- [46] Krishnan, Hema, M. Sudheep Elayidom, and T. Santhanakrishnan. “Weighted holoentropy-based features with optimised deep belief network for automatic sentiment analysis: reviewing product tweets.” *Journal of Experimental & Theoretical Artificial Intelligence* (2021): 1–29.

Biographies



Archana Nagelli, is pursuing her Ph.D (External Part-time) in School of Computer Science and Engineering at Vellore Institute of Technology, Chennai Campus, Chennai. She is having teaching experience of more than 15 years. She completed her Masters in Software Engineering. She published few papers in International conferences and reputed journals. Her research area includes Data Mining and Data Analytics.



B. Saleena is currently working as Professor in School of Computer Science and Engineering, at Vellore Institute of Technology, Chennai Campus, Chennai, India. She has completed her Ph.D in Computer Science from VIT Vellore, India. She has published more than 25 papers in reputed journals and conferences. Her current research interests include Data Mining, Software Engineering, Machine Learning and Semantic web Technologies.