International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I10P228

Volume 71 Issue 10, 314-325, October 2023

Original Article

CoSFGCN: Co-Sensitive Fusion Graph Convolution Network for Sentiment Analysis

M. Priya Alagu Dharshini[1], S. Antelin Vijila[2], S. P. Victor[3]

[1]*Manonmaniam Sundaranar University, Tirunelveli, Tamil Nadu, India.*
[2]*Department of Computer Science & Engineering, Manonmaniam Sundaranar University, Tirunelveli, Tamil Nadu, India.*
[3]*Department of Computer Science, St. Xavier's College, Tirunelveli, Tamil Nadu, India.*

[1]*Corresponding Author : preethi.murugiah10@yahoo.com*

Received: 12 July 2023 Revised: 08 September 2023 Accepted: 13 September 2023 Published: 03 October 2023

Abstract - *Review analysis greatly influences the business industry to a greater level because it expresses the notion of customers. Many researchers addressed this issue by classifying the sentiments with better compositionality. The main objective of this research work is to handle the issues of excessive content, language imperfections, and less-intensive emotional words which affect the performances of sentiment analysis. This paper proposes a novel framework known as a Co-Sensitive Fusion Graph Convolution Network (CoSFGCN), which is based on a Graph Convolutional Network (GCN). This framework incorporates the properties of Syntactic and Co-Sensitive Specific Semantic GCN (CS^3GCN) to utilise both the semantics and syntax of the words to provide additional weightage for graph learning. The performance analysis done on five benchmark datasets, LAP14, TWITTER_15, REST16, REST15, and REST14, shows better results when compared with the previous methods.*

Keywords - *CoSFGCN, GCN, Graph learning, Semantic graph model, Sentiment analysis.*

1. Introduction

The promotion of the business industry is greatly influenced by their views of it by customers. The business industry needs to attract customers, whether it is a shop or an e-commerce platform. Word of mouth or written reviews are the basis for this industry. Various stakeholders will write reviews in distinct dimensions. The abundance of reviews about each system is very complex to process. Sentiment analysis, a popular text mining research, helps in this research. When a person tries to buy pizza from a shop, he looks for the reviews. Some example reviews are 'pizza was burnt', 'poor quality food', and 'food delivery was late by 40 min'. In this case, the customer will hesitate to buy the pizza from that restaurant. At the same time, the owners can try to find a solution for this issue to regain the customers' trust. Management reviewing, repairing, or replacing the oven, giving training for staff, etc., may be the remedy taken for solving the issue. Thus, analysis of these reviews is most significant to evaluate the commodity purchased for both customers and business owners. The incredible growth of communication techniques like Facebook, Twitter, YouTube, and personal websites aid customers in posting their reviews. Analysing and classifying these reviews based on their sentiments can help this work. This process is the classification of the text focusing on the positive or negative sentiment of the text [1]—the easiness and high accuracy of techniques using machine learning aid in the semantic analysis. Still, the huge quantity of available reviews makes manual processing very complex. Admitting the fact that the high influence of these in business decisions demands effective processing of this data, which can be satisfied by good machine learning frameworks. The framework should be able to process this huge data effectively and help in decision-making. Most of the traditional system finds the sentiment of review with its whole content representation. Hence, it lacks prediction in the case of a sentence that may look positive but, at the same time, maybe neutral. This issue can be handled by a technique which analyses the syntactic and semantic content of the opinion.

2. Related Work

Deep learning frameworks recently implemented sentiment analysis with promising results. A recursive form of neural tensor network model was designed by Socher et al. [2] for binary sentiment classification over a sentiment treebank. A Convolutional Neural Network (CNN) was used by Kim [3], whereword2vec word embeddings are concatenated as the input layer. It follows convolution, max pooling, and SoftMax layers. The convolution layers are used with different filters. CNN and parallel CNN were proposed by Johnson et al. [4]. They created feature vectors using bag-of-words model variation. Two or more parallel layers with convolution layers are used in parallel CNN. Another improvement was CNN with seven layers [5], which analysed the sentiment of movie reviews. An unsupervised algorithm named paragraph vectors [6] was proposed that learns variable-length text, like paragraphs, sentences, etc., to form feature vectors with fixed length. A neural network model [7] was proposed, which captures the user information along with the semantics of the review.

One layered CNN was proposed by Chen et al. [8], which produces 300-dimensional vectors by learning varying length reviews. The analysis was done in Yelp datasets and the Internet Movie Database (IMDB) embeddings. Variation in length was handled by padding. A Recurrent Neural Network (RNN) was used to extract the temporal data. Naïve Bayes classifier and decision tree classifier were used by Callen Rain [9] for categorising reviews. Different machine learning algorithms like naive Bayes and perceptron algorithms were analysed by XuYun et al. [10] to predict the review rank by doing cross-validation. Multinomial Naive Bayesian [11] network was used for analysing sentiment from customer reviews as a model for business. A semantic role labelling task was done by Ronan Collobert [12] et al. Using CNN to avoid more operation-oriented engineering characterisation.

A semantic orientation calculator [13] was proposed for predicting sentiment. They used polarity-annotated word dictionaries annotated with their corresponding strength. A lexicon-based algorithm [14] by combining sentiment normalisation, sentiment terms, and evidence-based combination function was proposed. Domain-specific terms, emotions, and modifiers are integrated by Asghar et al. [15] for sentiment analysis of the comments made by users. A Unigram Mixture Model (UMM) [16] based work was proposed for weakly-labeled and properly labelled emotion text. A package of RTextTools and LIWC2015 lexicon [17] was proposed as a machine learning package for analysing sentiment based on the lexicon. WKWSCI [18], a sentiment lexicon, was proposed by Khoo and John. Chinese microblog [19] based sentiment classification was done with sentiment lexicons of negative words, the lexicon of words in the network, and an added degree adverb lexicon. Lexicons and corpus [20] were used to construct the adaptive sentiment vocabulary of Weibo. A graph with two levels of layer model [21] was designed with the help of the sentiment of candidate words and emoji. Another work based on CNN [22] was proposed to analyse the polarity of the sentiment using the n-gram feature input and the statistical features of the word co-occurrence properties in the tweet. A Target-dependent Convolutional Neural Network (TCNN) [23], which uses the correlation of the target word with the surrounding words, was proposed by Hyun et al. for learning the effect of neighboring words on the target words.

An individual word's polarity may affect the emotion of a sentence in different ways. To acquire this, attention mechanisms were introduced by researchers. Some features will be dominant, and others will be recessive. For combining these two features, an extended LSTM [24] was proposed with the name Sentic LSTM. An output gate is inserted separately in the model unit to add concept-level input and token-level memory. MLSTM [25] was another model proposed for imparting attention implemented based on memristor. Bid et al. [26] proposed a combined algorithm where RNN is used for locating, and CNN is used for capturing the long-term dependencies. A divide-and-conquer method [27] was proposed, where a sequence model based on a neural network was used for classifying the sentences, and CNN was used for sentiment classification by inputting each set of sentences. The LSTM model, along with a keyword vocabulary, was proposed by Hu et al. [28] for doing attention-based sentiment classification of short texts.

Graph Neural Networks (GNN) [29, 30] was introduced as a tool for extracting the co-occurrence between the words. This network preserves structured information globally. The relational structure of GNN will be rich enough to preserve it and utilise graph embeddings for classification. This concept was further used for the classification of texts [31]. Graph Convolutional Network (GCN) [32] proposed a new model based on GNN, which is used for constructing a single large graph. The nodes of this network contain words and documents from an entire corpus that captures neighbourhood information in high order. The edges are formed by co-occurrence information of words. In the same way, the frequency of word and word documents is used to form the edge between word and document nodes.

This work proposes a novel algorithm for analysing the review. The framework was implemented based on Graph CNN with VADER [33] based polarity extraction to refine the GCN. Further, the model is improved by using an attention scheme.

3. Proposed Co-Sensitive Fusion Graph Convolution Network (CoSFGCN)

This work proposes a novel framework for review analysis based on GCN [32]. The proposed Co-Sensitive Fusion Graph Convolution Network (CoSFGCN) is a fusion of Syntactic sand Co-Sensitive Specific Semantic GCN (CS^3GCN) Models which utilise both syntax and semantics of the words while considering a sentence.

Problem Definition: Let $S = \{w_1, w_2, w_3, \ldots, w_n\}$ be the sentence with n words used for the review. This proposed work aims to classify S as positive, negative or neutral feedback. The sentence is applied to embedding ($\mathfrak{E}$) and BiLSTM ($\mathcal{B}$). This output is fed into three parallel frameworks.

$$SynGCN, Attention(Att)\ model, and\ S^3GCN.$$

These outputs are merged to form the final feature vector, F, by using equation (1).

$$F = O(SynGCN) + O(Att) + O(CS^3GCN) \quad \text{M} \tag{1}$$

F is used for classifying the review. Softmax classification is used for finding the output as in equation (2).

$$Rev = SoftMax(F) \tag{2}$$

GCN: GCN is a technique for learning graphs by semi-supervised learning. An example for GCN ($\boldsymbol{\mathcal{G}}$) with ƞ nodes is given in Fig 1. Let E_{ij} be the edge formed between i^{th} and j^{th} nodes. The adjacency matrix M_{ij} gives the availability of edge E_{ij}. The information from the immediate neighbors is encoded, and syntactic constraints are generated using convolution operations. This work creates a dependency graph by using. E_{ij} based on which adjacency matrix M is calculated.

$\mathcal{G}$ will have many layers, with layer l having the hidden output of the i^{th} node as o_i^l output for individual nodes in that layer. ɥ be the nonlinear function (Relu), t^l be the weight for transformation and b^l be the layer l bias. Then o_i^l can be calculated as in equation (3).

$$o_i^l = \text{ɥ}\left(\sum_{j=1}^{n} M_{ij} t^l o_i^{l-1} + b^l\right) \qquad (3)$$

The proposed CS^3GCN model addresses the noises caused by the information and imperfect parsing. This can also gather information about disconnected words. Fig 2 gives the proposed architecture.

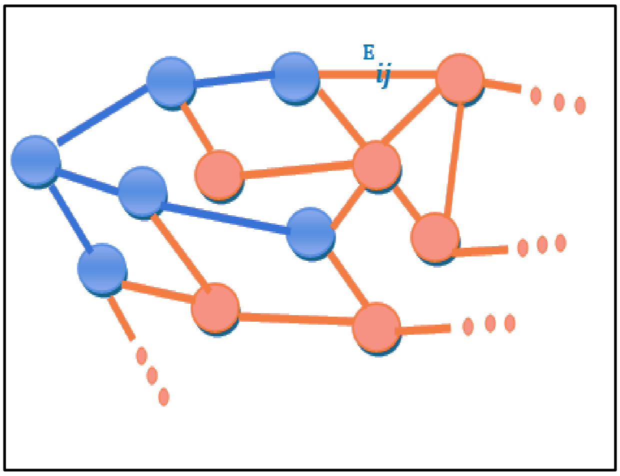
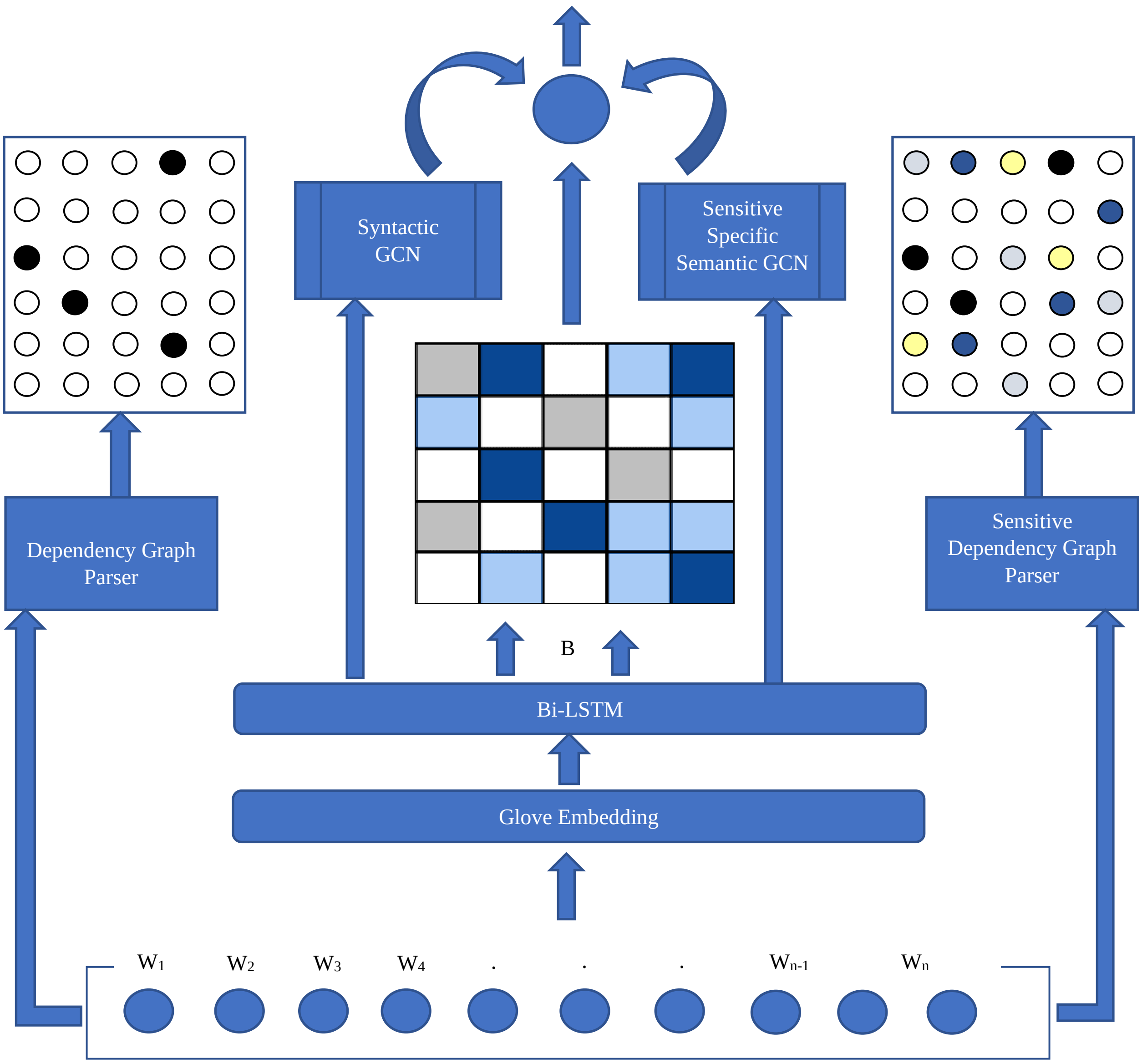


Fig. 1 A GCN example



$ The Laptop Gives Better Performance for this Work

Fig. 2 Proposed architecture of Co-Sensitive fusion GCN

3.1. Bidirectional LSTM Brach

Consider a sentence, $S = \{w_1, w_2, w_3, \ldots, w_n\}$, which has n words. The embedding of the words is used for gathering the relationships between the words in an effective manner. Every word (w_i) in the sentence S is fed into the embedding block to get the embedded real-valued vector ($\mathfrak{E} \in \mathbb{R}^{|N| \times dim_e}$). dim_ehas a lower dimension than that of the word in the sentence with vocabulary size |N|. This work uses Global Vectors for Word Representation (GloVe) [34] for embedding, which utilises the co-occurrences of words. Embedded words are converted to hidden vectors ($\mathcal{B}$) using the biLSTM model [35, 36].

3.2. Syntactic Graph Convolutional Network (SynGCN)

Encoding syntactic information of the input as graph representation is done in $SynGCN$ module. The hidden vectors of BiLSTM output ($\mathcal{B}$) are fed into Syntactic GCN as initial node form. This $SynGCN$ model utilises the adjacency matrix with the help of dependency sparser in the form of graph mode. The syntactic representation ($\mathcal{B}^s$) formed from $\mathcal{B}$ using the equation 3, it can be denoted as $\mathcal{B}^s = \{b_1^s, b_2^s, \ldots, b_n^s,\}$.Each b_i^s gives the representation of i^{th} node.

3.3. Co-Sensitive Specific Semantic GCN (CS^3GCN)

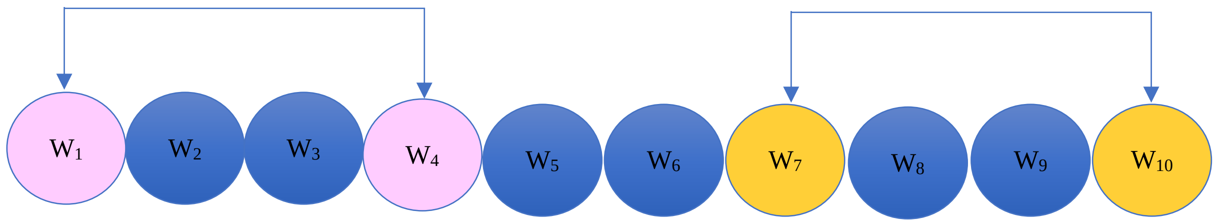
The proposed model improves the strength of the review opinion detection, along with syntactic features and semantic features, especially opinion-sensitive related semantic learning is, introduced in the name of CS^3GCN. It consists of three sections of the process: (i) Sensitive Dependency Graph Parser, (ii) Co-Sensitive Weighting and (iii) Dependency and Co-Sensitive Masking. The following section provides a full description of the model. Let the sentence be S. The direct link (DL) and sensitivity are used for calculating long-range dependencies. The priority of the words in contributing importance is gathered by using this process.

$$S = \{w_1, w_2, w_3, \ldots, w_n\} \tag{4}$$

The GCN is formed with the help of the adjacency matrix, which is estimated with the help of a dependency graph parser. In this proposed model and the traditional Dependency Graph, additional sensitive-based edges are included for GCN learning.

3.3.1. Sensitive Dependency Graph Parser (SDGP)

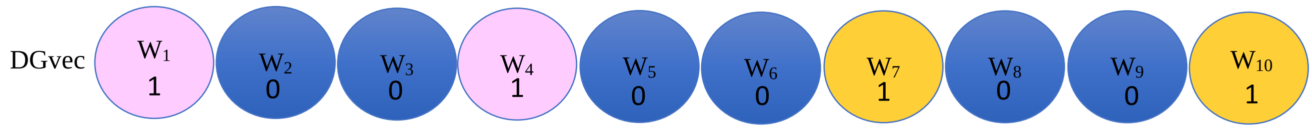
Consider the fig. 3 (a), which shows the dependency of words. If there is a dependency between one word and the other, a link is formed between both words. Based on this link, a dependency graph is formed. For this calculation, word 1 has a dependency on word 4, and word 7 has a dependency on word 10. The corresponding dependency can be denoted in the dependency (adjacency) matrix as in Fig 3 (b) and the Dependency Graph Vector (DGV)in Fig 3(c).



(a)

	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
W1	0	0	0	1	0	0	0	0	0	0
W2	0	0	0	0	0	0	0	0	0	0
W3	0	0	0	0	0	0	0	0	0	0
W4	1	0	0	0	0	0	0	0	0	0
W5	0	0	0	0	0	0	0	0	0	0
W6	0	0	0	0	0	0	0	0	0	0
W7	0	0	0	0	0	0	0	0	0	1
W8	0	0	0	0	0	0	0	0	0	0
W9	0	0	0	0	0	0	0	0	0	0
W10	0	0	0	0	0	0	1	0	0	0

(b)

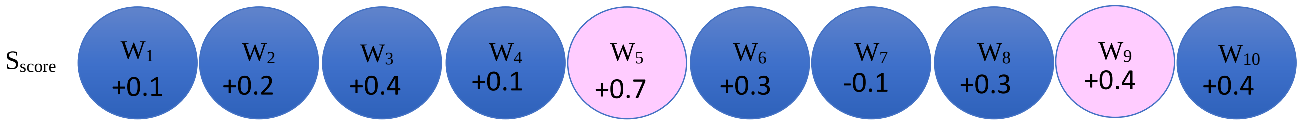


(c)

Fig. 3 Dependency Graph Formulation (a) Dependency Level (b) Corresponding Dependency graph, and (c) Dependency Graph Vector (*DGV*)

Fig 3 (b) shows the graph available between the nodes dependent on each other. The sensitivity of the words should also be incorporated for a better understanding of the importance of words. VADER [33] is used in this work for calculating sentiment scores. This is calculated by mapping the lexical characters to sentiment scores. It will have a dictionary with sentiment scores with VADER's sentiment lexicon available in [34]. Each term will have a sensitive score ($\boldsymbol{Sc_i}$), either positive or negative. When the lexicon score of the word is greater than +0.6 or lesser than -0.3, they are considered more sensitive. Based on this, a sensitive vector is formed. Suppose the positive polarity is greater than 0.6 or the negative polarity is lesser than -0.3. In that case, those terms(words) are also considered to build edges between those sensitive words, even though they have a connection with other terms by the normal dependency parser or not. The sensitive words are mapped to 1 in the dependency graph.

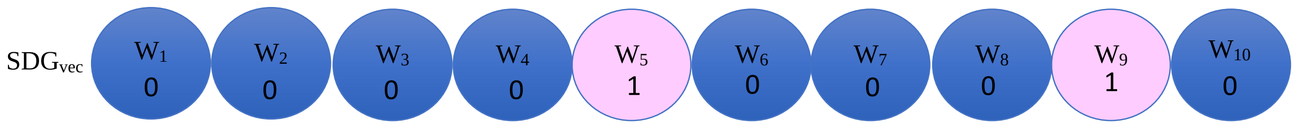
In this work, since the word 5 has a sentiment score of +0.7 and the word 9 has a sentiment score of -0.4, they are assigned value 1 in the dependency graph. If three terms or words attain the specified sensitive score means, it will generate a link between all three terms in all combinations, such as term1 to term2, term1 to term3, and term2 to term3 in bidirectional mode if any word is a dependency vertex in the normal dependency graph. So, the new link will not be created with those words because the vertex is present already. In this example, for the sensitivity score values shown in Fig 4 (a), the sensitive dependency graph can be drawn as Fig 4 (b) and its corresponding sensitive dependency graph vector is shown in Fig 4(c).



(a)

	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}
W_1	0	0	0	1	0	0	0	0	0	0
W_2	0	0	0	0	0	0	0	0	0	0
W_3	0	0	0	0	0	0	0	0	0	0
W_4	1	0	0	0	0	0	0	0	0	0
W_5	0	0	0	0	0	0	0	0	1	0
W_6	0	0	0	0	0	0	0	0	0	0
W_7	0	0	0	0	0	0	0	0	0	1
W_8	0	0	0	0	0	0	0	0	0	0
W_9	0	0	0	0	1	0	0	0	0	0
W_{10}	0	0	0	0	0	0	1	0	0	0

(b)



(c)

Fig. 4 Sensitivity Evaluation. (a): Sensitivity Score Vector (b) Sensitive Dependency Graph Adjacency Matrix (c): Sensitive Dependency Graph Vector

3.4. Attention Mechanism (Att)

Hidden state vectors are produced using the attention mechanism as $O(Att)$. These features are relevant to words for each context word based on attention weights. For the sentence S, consider v_i be the vector associated with every word. v_i will give the word embedding required for each word with dimension d. This forms a sentence Sv with $n\,X\,d$ dimension. The dependency between these sequences of words is required for extracting the attention between those words. For this, Bi-LSTM (Bi-directional LSTM) is used as in equations (7) and (8).

$$\overrightarrow{h_i} = \overrightarrow{LSTM}(\mathrm{v}_i, \overrightarrow{h_{i-1}}) \quad (7)$$

$$\overleftarrow{h_i} = \overleftarrow{LSTM}(\mathrm{v}_i, \overleftarrow{h_{i-1}}) \quad (8)$$

$$h_i = \overrightarrow{h_i} + \overleftarrow{h_i} \quad (9)$$

Both $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ are concatenated as h_i(hidden state). Each LSTM state has a hidden unit number as u. The hidden vector, H of LSTM, is calculated in equation (10) with size $n\ X\ 2u$.

$$H = (h_1, h_2, \dots, h_n) \quad (10)$$

Self-attention is applied to $\mathrm{v}_p H$ to get the output vector with weight wg. A vector v_p with size d_{wg} and a weight matrix, w_s having size $as\ d_{wg}\ X\ 2u$. d_{wg} will be having an arbitrary size. $softmax$ is used to ensure the sum of the weights is 1. $wg\ is$ calculated using the equation (11).

$$wg = softmax(w_s(\tanh \mathrm{v}_p H^T)) \quad (11)$$

LSTM hidden states, H are taken as input and combined to get the output vector, V. This vector gives a set of dependent words reflecting specific semantic-based words. Overall, semantics are represented by all Vs. If we extract multiple hops of data, it will extract more attention-specific information. Sentences with $r$hops of attention need to be done for the sentence, and then equation (12) is used to calculate the annotation matrix, M with 2-layered MLP without bias.

$$M = softmax(w_s(\tanh \mathrm{v}_p H^T)) \quad (12)$$

The annotation matrix M is multiplied with hidden states H to get the sentence embedding matrix (equation (13)).

$$T = M * H \quad (13)$$

4. Experimental Results and Analysis

4.1. Datasets and Experimental Setup

The performance of the proposed work is done by using five datasets such as TWITTER_15 [37], LAP_14 [38], REST_16 [39], REST_15 [40], and REST_14 [38]. SemEval 2014 [38] is used to get LAP_14 and REST_14. SemEval 2015's [40] task 12 is used to get REST_15. Task 5 of SemEval 2016 [39] is used to get REST_16. Restaurant and laptop-based reviews are used for getting these datasets. TWITTER_15 [37] take the posts from Twitter. Datasets and their information are listed in Table 1. All the data are initially pre-processed with the concept of case conversion into lower case, removing URLs, special characters, non-English words, stop words, and non-alphabetic first letters, replacing emoij into corresponding text, Lemmatization and Tokenization. Parameter Settings: The learning rate is scheduled by deploying Adam optimise (learning rate 0.001) with 32 batch sizes and 50 Epochs in 6GB graphics card-enabled device with PyTorch.

Table 1. Datasets with class wise samples

Dataset	Category	REST_16	REST_15	REST_14	LAP_14	TWITTER_15
Train	Positive Samples	1240	912	2164	994	1561
	Negative Samples	439	256	807	870	1560
	Neutral Samples	69	36	637	464	3127
Test	Positive Samples	912	326	728	341	173
	Negative Samples	256	182	196	128	173
	Neutral Samples	36	34	196	169	346

4.2. Results

The performance of the proposed CoSFGCN model is analysed with the help of the following metrics: accuracy and F1-Score. Below are the formulas used for evaluation metrics.

Accuracy (Acc) = (T P + T N)/ (T P + T N + F P + F N) (14)

Precision (Pre)= TP / (TP + F P) (15)

Recall (Rec) = TP / (TP + F N) (16)

F1-Score(F1s) = (2 * Pre * Rec) / (Pre+Rec) (17)

The performance based on the above metrics Accuracy and F1-Score of the proposed model and the existing models is shown in Table 2.

From table 2, it is found that accuracy is the proposed better result for the dataset TWITTER_15, LAP_14, REST_14, REST_15 and REST_16, with the highest score achieved by the methods. The proposed CoSFGCN achieves +1.16% more accuracy than DualGCN [55] for the TWITTER_15 dataset. For the LAP_14 dataset, the proposed approach attains +0.31% more than the BERT SDGCN [48]. Similarly, +0.28% more than BERT RGAT [47] for the REST-14 dataset, +0.28% more than BERT ASGCN [54] for the REST_15 dataset and+0.49% more than DualGCN[55]for the REST_16 dataset.

Table 2. Accuracy and F1-Score analysis with existing models

Model	TWITTER_15		LAP_14		REST_14		REST_15		REST_16	
	Acc.	F1S	Acc.	F1S	Acc.	F1S	Acc.	F1S	Acc.	F1S
SVM [37]	63.4	63.3	70.49	N/A	80.16	N/A	N/A	N/A	N/A	N/A
LSTM [42]	69.56	67.7	69.28	63.09	78.13	67.47	77.37	55.17	86.8	63.88
MemNet [43]	71.48	69.9	70.64	65.17	79.61	69.64	77.31	58.28	85.44	65.99
AOA [44]	72.3	70.2	72.62	67.52	79.97	70.42	78.17	57.02	87.5	66.21
IAN [45]	72.5	70.81	72.05	67.38	79.26	70.09	78.54	52.65	84.74	55.21
TNet-LF [41]	72.98	71.43	74.61	70.14	80.42	71.03	78.47	59.47	89.07	70.43
ASCNN [54]	71.05	69.45	72.62	66.72	81.73	73.1	78.47	58.9	87.39	64.56
DT ASGCN [54]	71.53	69.68	74.14	69.24	80.86	72.19	79.34	60.78	88.69	66.64
DG ASGCN [54]	72.15	70.4	75.55	71.05	80.77	72.02	79.89	61.89	88.99	67.48
BASEBERT [46]	-	-	79.73	75.5	82.74	73.73	82.16	64.96	89.43	74.2
PTBERT [53]	-	-	78.89	75.89	85.92	79.12	-	-	-	-
BERT RGAT [47]	-	-	80.94	78.2	86.68	80.92	-	-	-	-
BERT SDGCN [48]	-	-	81.35	78.34	83.57	76.47	-	-	-	-
BERT SKGCN [50]	-	-	79	75.57	83.48	75.19	83.2	66.78	87.19	72.02
BERT DGEDT [49]	-	-	79.8	75.6	86.3	80	84	71	91.9	79
BERTCDM LCF [51]	-	-	80.3	76.85	86.28	80.24	83.83	69.97	90.62	76.93
BERTCDW LCF [51]	-	-	79.73	76.07	86.16	80.12	83.77	69.03	91	77.1
BERTCDM LCFS [52]	-	-	79.99	76.51	86.31	80.32	83.4	68.81	90.81	75.86
BERTCDW LCFS [52]	-	-	80.25	76.72	86.43	80.84	84.07	69.67	90.35	76.28
BERT ASGCN [54]	-	-	79.83	75.89	84.76	77.94	84.22	72.9	91.05	77.05
DualGCN[55]	75.43	74.24	78.37	76.33	84.20	78.24	83.03	73.46	93.18	79.40
CoSFGCN	76.59	75.39	81.66	80.2	86.96	81.66	84.5	75.57	93.67	80.86

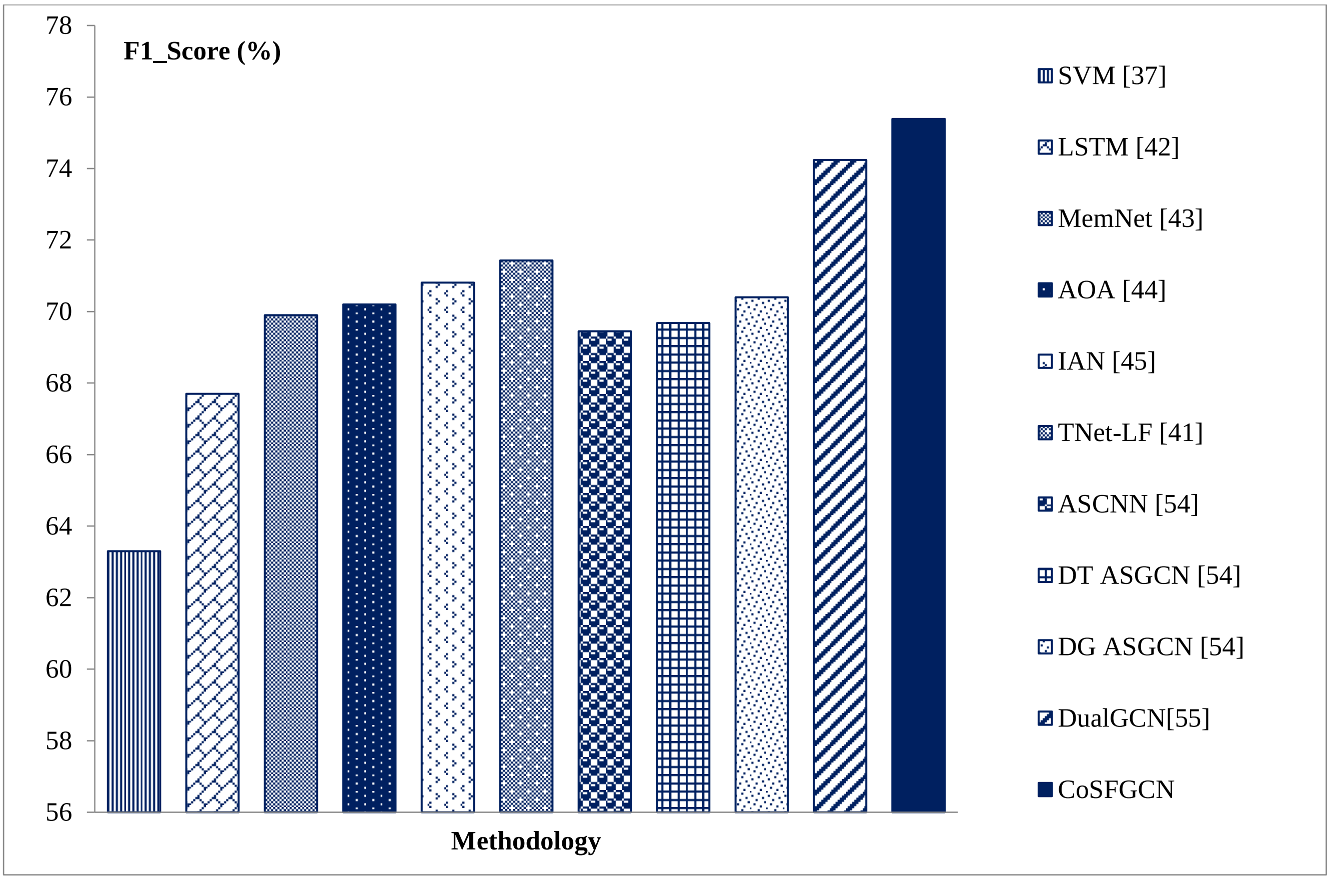


Fig. 5 F1-score for Twitter_15 dataset Analysis

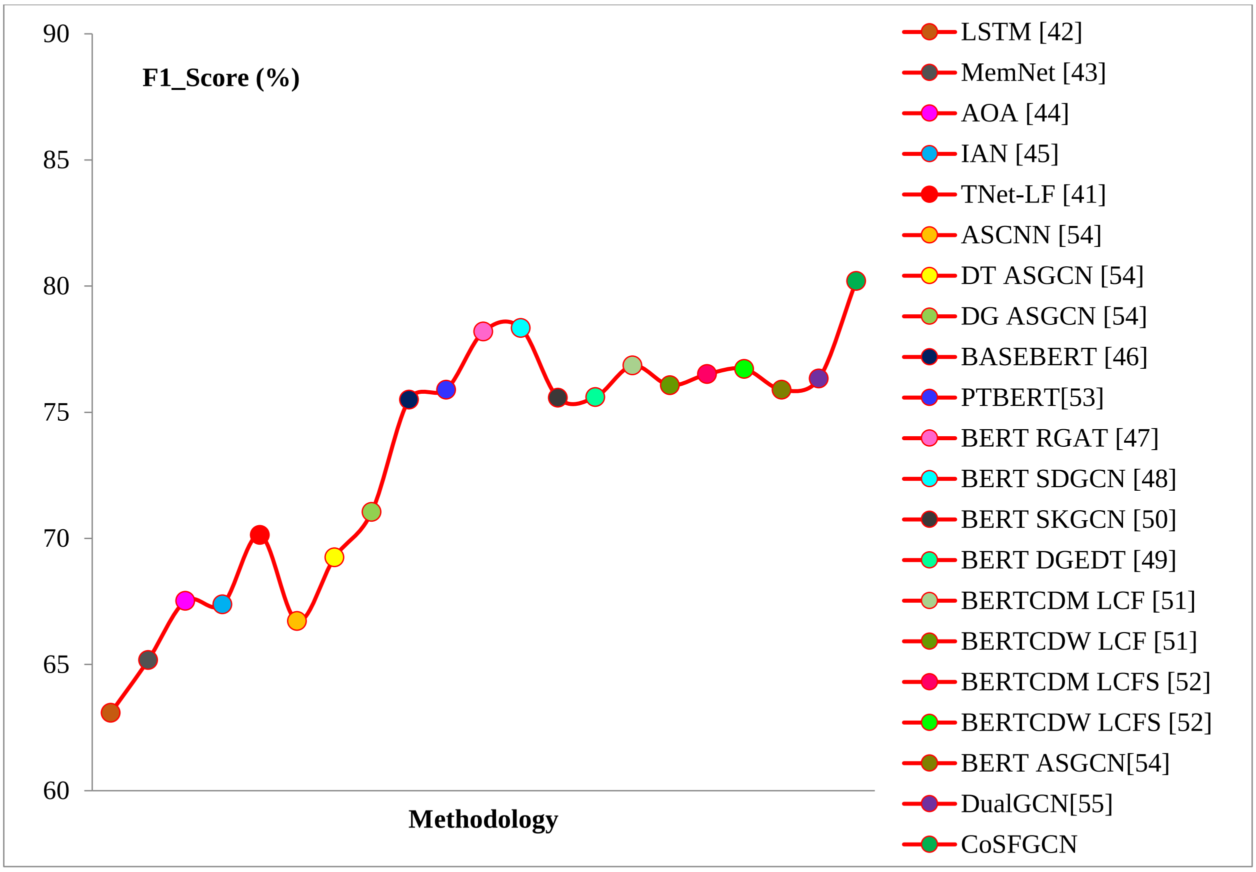


Fig. 6 F1-score for LAB_14 dataset analysis

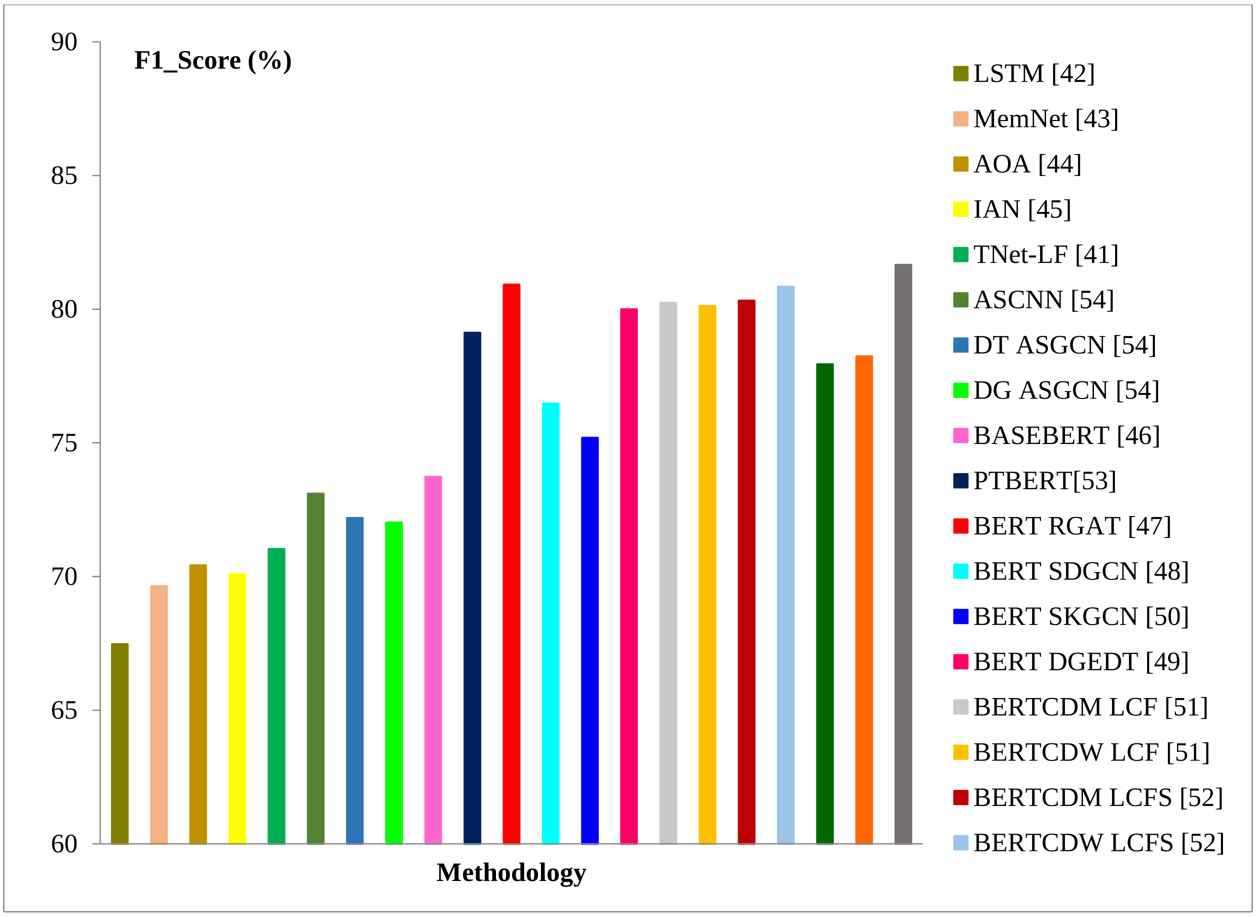


Fig. 7 F1-score for REST_14 dataset analysis

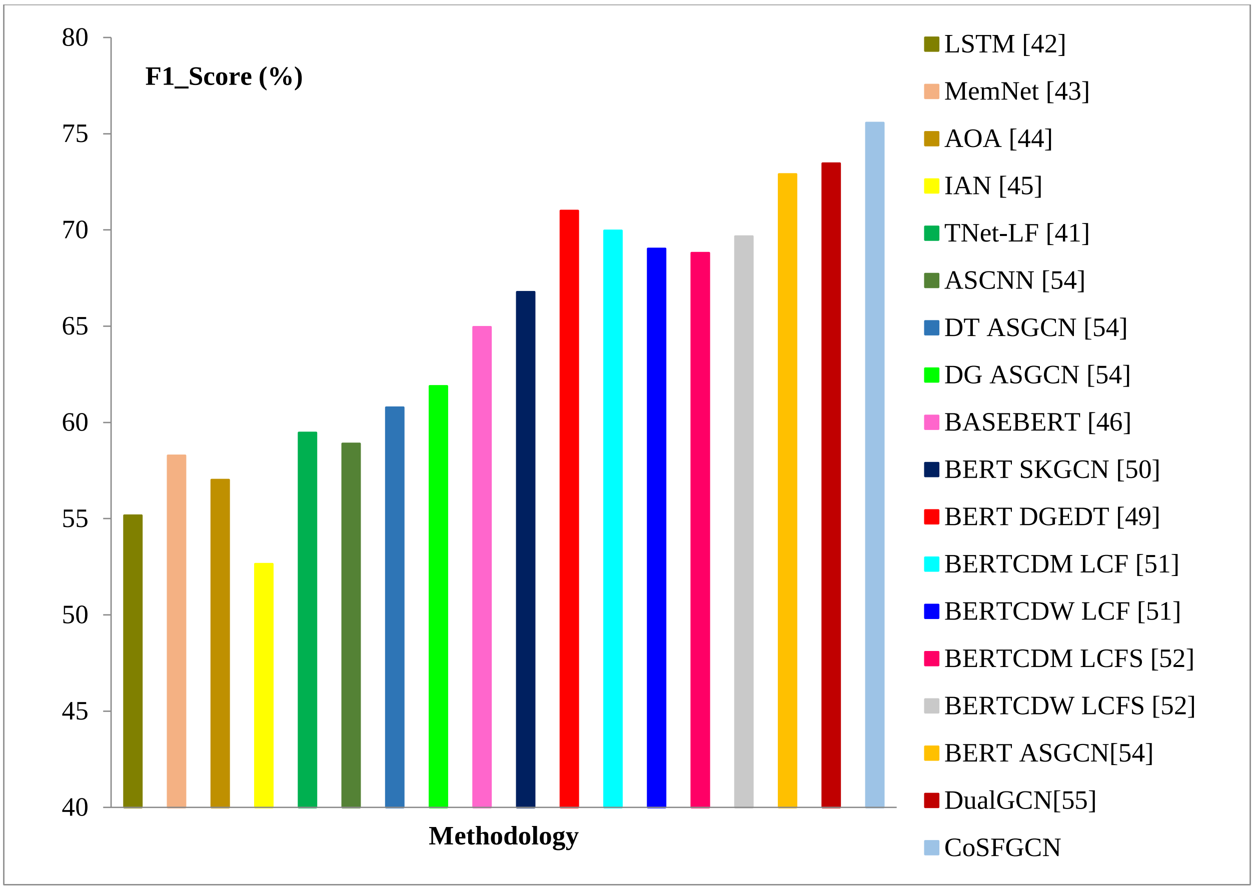


Fig. 8 F1-score for REST_15 dataset analysis

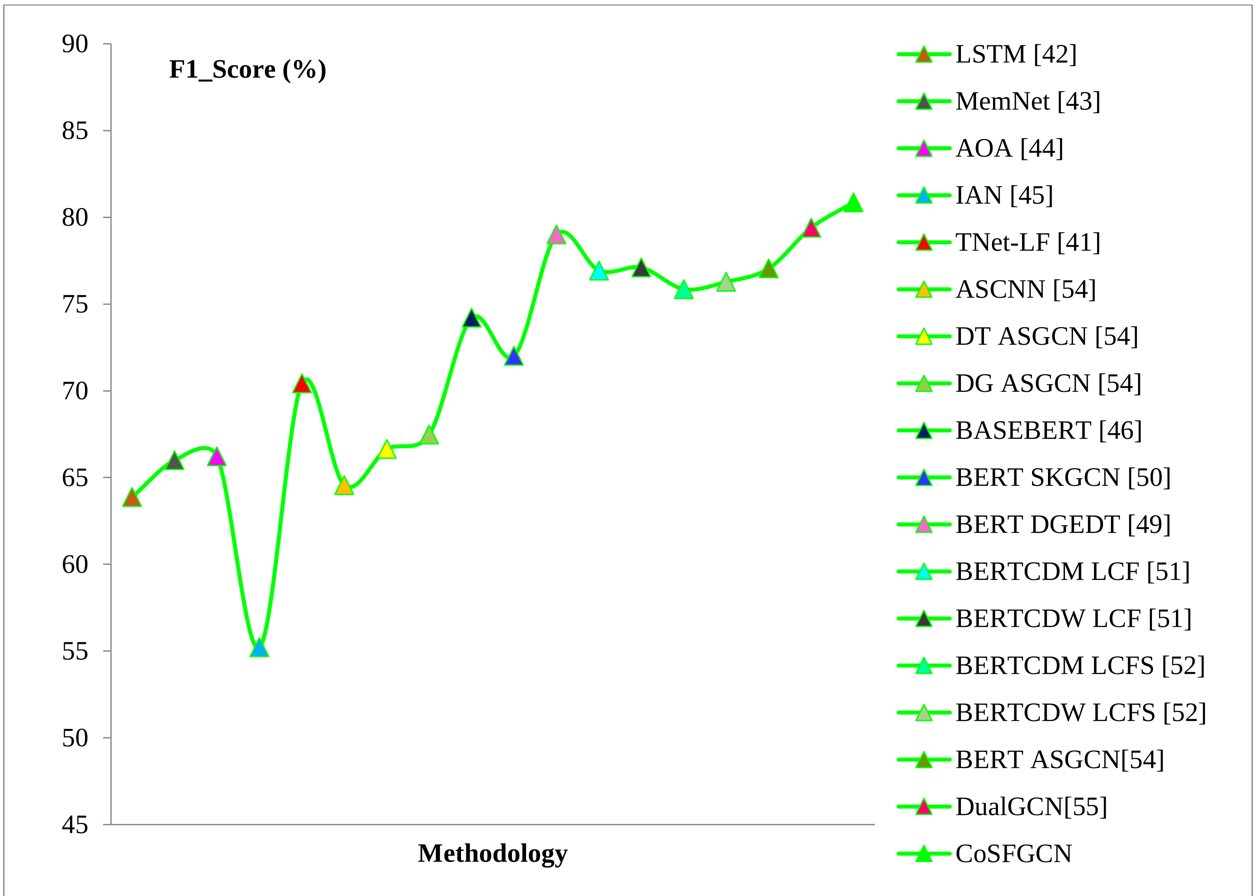


Fig. 9 F1-score for REST_16 dataset analysis

The above Figures clearly show the better performance of the proposed method for TWITTER_15, LAP_14, REST_14, REST_15 and REST_16. The proposed CoSFGCN gives better results in terms of F1-Score, +1.15% better than the DualGCN [55] approach for TWITTER_15 dataset, similarly in terms of F1-score the proposed model attains +1.86% more than BERT SDGCN [48] for LAP_14 dataset +0.74% more than BERT RGAT [47] for REST_14, +2.11% better than DualGCN [55] for REST_15 dataset and +1.46% higher than DualGCN [55] for REST_16 dataset.

5. Conclusion

The proposed CoSFGCN: Co-Sensitive Fusion Graph Convolution Network model utilises both syntax and semantics of the words in a sentence, and the sentiment classification accuracy is improved than the existing algorithms. Incorporation of the sensitive-based dependency graph model improves semantic learning, which provides an effective polarity classification model. The graph hidden node selection for further graph layer is fine-tuned by the attention mechanism. The fusion of the Syntactic and Sensitive Specific Semantic graph model improves the performance of review analysis from various social media sources. The proposed CoSFGCN model achieves an 84.676% average accuracy, including all the datasets and 78.736% in terms of F1-Score. The future, along with the text rating concept, can be included additionally to improve the sentiment classification performance.

References

[1] Kim Schouten, and Flavius Frasincar, "Survey on Aspect-Level Sentiment Analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 813-830, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[2] Richard Socher et al., "Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank," *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1631-1642, 2013. [Google Scholar] [Publisher Link]

[3] Yoon Kim, "Convolutional Neural Networks for Sentence Classification," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[4] Rie Johnson, and Tong Zhang, "Effective Use of Word Order for Text Categorization with Convolutional Neural Networks," *Proceedings of Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, pp. 103–112, 2015. [Google Scholar] [Publisher Link]

[5] Xi Ouyang et al., "Sentiment Analysis Using Convolutional Neural Network," *Proceedings of IEEE International Conference on Computer and Information Technology, Ubiquitous Computing and Communications, Dependable, Autonomic and Secure Computing, Pervasive Intelligence and Computing*, pp. 2359-2364, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[6] Quoc Le, and Tomas Mikolov, "Distributed Representations of Sentences and Documents," *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, no. 2, pp. 1188-1196, 2014. [Google Scholar] [Publisher Link]

[7] Duyu Tang, Bing Qin, and Ting Liu "Learning Semantic Representations of Users and Products for Document Level Sentiment Classification," *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 1014–1023, 2015. [Google Scholar] [Publisher Link]

[8] Tao Chen et al., "Learning User and Product Distributed Representations using a Sequence Model for Sentiment Analysis," *IEEE Computational Intelligence Magazine*, vol. 11, no. 3, pp. 34-44, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[9] Callen Rain, "*Sentiment Analysis in Amazon Reviews Using Probabilistic Machine Learning*," Swarthmore College, vol. 42, 2013. [Google Scholar] [Publisher Link]

[10] Y. Xu, X. Wu, and Q. Wang, "Sentiment Analysis of Yelps Ratings based on Text Reviews," *17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, vol. 17, no. 1, 2015. [Google Scholar] [Publisher Link]

[11] Elli Maria Soledad, and Yi-Fan Wang, "Amazon Reviews, Business Analytics with Sentiment Analysis," *Perceived Derived Attributes of Online Customer Reviews*, 2016. [Google Scholar]

[12] Ronan Collobert et al., "Natural Language Processing (Almost) from Scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011. [Google Scholar] [Publisher Link]

[13] Maite Taboada et al., "Lexicon Based Methods for Sentiment Analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[14] Anna Jurek, Maurice D. Mulvenna, and Yaxin Bi, "Improved Lexicon-Based Sentiment Analysis for Social Media Analytics," *Security Informatics*, vol. 4, no. 1, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[15] Muhammad Zubair Asghar et al., "Lexicon Enhanced Sentiment Analysis Framework Using Rule-Based Classification Scheme," *PLoS ONE*, vol. 12, no. 2, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[16] Anil Bandhakavi et al., "Lexicon-based Feature Extraction for Emotion Text Classification," *Pattern Recognition Letters*, vol. 93, pp. 133–142, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[17] Chedia Dhaoui, Cynthia M. Webster, and Lay Peng Tan, "Social Media Sentiment Analysis: Lexicon versus Machine Learning," *Journal of Consumer Marketing*, vol. 34, no. 6, pp. 480–488, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[18] Christopher S.G. Khoo, and Sathik Basha Johnkhan, "Lexicon-Based Sentiment Analysis: Comparative Evaluation of Six Sentiment Lexicons," *Journal of Information Science*, vol. 44, no. 4, pp. 491–511, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[19] Shunxiang Zhang et al., "Sentiment Analysis of Chinese Micro-Blog Text based on Extended Sentiment Dictionary," *Future Generation Computer Systems*, vol. 81, pp. 395–403, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[20] Hamidreza Keshavarz, and Mohammad Saniee Abadeh, "ALGA: Adaptive Lexicon Learning Using Genetic Algorithm for Sentiment Analysis of Microblogs," *Knowledge-Based Systems*, vol. 122, pp. 1–16, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[21] Shi Feng et al., "A Word-Emoticon Mutual Reinforcement Ranking Model for Building Sentiment Lexicon from Massive Collection of Microblogs," *World Wide Web*, vol. 18, no. 4, pp. 949–967, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[22] Zhao Jianqiang, Gui Xiaolin, and Zhang Xuejun, "Deep Convolution Neural Networks for Twitter Sentiment Analysis," *IEEE Access*, vol. 6, pp. 23253–23260, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[23] Dongmin Hyun et al., "Target-Aware Convolutional Neural Network for Target-Level Sentiment Analysis," *Information Sciences*, vol. 491, pp. 166–178, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[24] Yukun Ma et al., "Sentic LSTM: A Hybrid Network for Targeted Aspect-Based Sentiment Analysis," *Cognitive Computation*, vol. 10, no. 4, pp. 639–650, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[25] Huiling Chen et al., "Fine-Grained Sentiment Analysis of Chinese Reviews Using LSTM Network," *Journal of Engineering Science and Technology Review*, vol. 11, no. 1, pp. 174–179, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[26] Shiping Wen et al., "Memristive LSTM Network for Sentiment Analysis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 3, pp. 1794-1804, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[27] Fazeel Abid et al., "Sentiment Analysis through Recurrent Variants Latterly on Convolutional Neural Network of Twitter," *Future Generation Computer Systems*, vol. 95, pp. 292–308, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[28] Tao Chen et al., "Improving Sentiment Analysis via Sentence Type Classification using BiLSTM-CRF and CNN," *Expert Systems with Applications*, vol. 72, pp. 221–230, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[29] Fei Hu et al., "Emphasizing Essential Words for Sentiment Classification based on Recurrent Neural Networks," *Journal of Computer Science and Technology*, vol. 32, no. 4, pp. 785–795, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[30] Peter W. Battaglia et al., "Relational Inductive Biases, Deep Learning, and Graph Networks," a*rXiv*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[31] Hong Yun Cai et al., "A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1616–1637, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[32] Thomas N. Kipf, and Max Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *ICLR*, 2017. [Google Scholar] [Publisher Link]

[33] Liang Yao, Chengsheng Mao, and Yuan Luo, "Graph Convolutional Networks for Text Classification," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 3, no. 1, pp. 7370-7377, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[34] C.J. Hutto, and Eric Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," *Eighth International AAAI Conference on Weblogs and Social Media*, vol. 8, no. 1, pp. 216-225, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[35] Jeffrey Pennington et al., "Glove: Global Vectors for Word Representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532-1543, 2014. [Google Scholar] [Publisher Link]

[36] Yaqing Wang et al., "EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 849–857, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[37] Alex Graves, and Jurgen Schmidhuber, "Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[38] Li Dong et al., "Adaptive Recursive Neural Network for Target-Dependent Twitter Sentiment Classification," *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. 2, pp. 49–54, 2014. [Google Scholar] [Publisher Link]

[39] Maria Pontiki et al., "Semeval-2014 Task 4: Aspect-based Sentiment Analysis," *Proceedings of the 8th International Workshop on Semantic Evaluation*, pp. 27–35, 2014. [CrossRef] [Publisher Link]

[40] Maria Pontiki et al., "Semeval 2016 Task 5: Aspect-based Sentiment Analysis," *Proceedings of SemEval-2016*, pp. 19-30, 2016. [Google Scholar] [Publisher Link]

[41] Maria Pontiki et al., "Semeval-2015 Task 12: Aspect Based Sentiment Analysis," *Proceedings of the 9th International Workshop on Semantic Evaluation*, pp. 486–495, 2015. [Google Scholar] [Publisher Link]

[42] Xin Li et al., "Transformation Networks for Target-Oriented Sentiment Classification," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 946– 956, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[43] Duyu Tang et al., "Effective LSTMS for Target-Dependent Sentiment Classification," *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 3298–3307, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[44] Duyu Tang, Bing Qin, and Ting Liu, "Aspect Level Sentiment Classification with Deep Memory Network," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 214–224, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[45] Binxuan Huang, Yanglan Ou, and Kathleen M Carley, “Aspect Level Sentiment Classification with Attention-Over-Attention Neural Networks,” *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behaviour Representation in Modeling and Simulation*, pp. 197–206, 2018. [CrossRef] [Google Scholar] [Publisher Link]
[46] Dehong Ma et al., “Interactive Attention Networks For Aspect-Level Sentiment Classification,” *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 4068–4074, 2017. [Google Scholar] [Publisher Link]
[47] Jacob Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 4171–4186, 2019. [CrossRef] [Google Scholar] [Publisher Link]
[48] Kai Wang et al., “Relational Graph Attention Network for Aspect-based Sentiment Analysis,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3229– 3238, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[49] Pinlong Zhao, Linlin Hou, and Ou Wu, “Modeling Sentiment Dependencies with Graph Convolutional Networks for Aspect-Level Sentiment Classification,” *Knowledge-Based Systems*, vol. 193, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[50] Hao Tang et al., “Dependency Graph Enhanced Dual-Transformer Structure for Aspect-Based Sentiment Classification,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6578–6588, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[51] Jie Zhou et al., “SKGCN: Modeling Syntax and Knowledge via Graph Convolutional Network for Aspect-Level Sentiment Classification,” *Knowledge-Based Systems*, vol. 205, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[52] Biqing Zeng et al., “LCF: A Local Context Focus Mechanism for Aspect-Based Sentiment Classification,” *Applied Sciences*, vol. 9, no. 16, pp. 1-22, 2019. [CrossRef] [Google Scholar] [Publisher Link]
[53] Minh Hieu Phan, and Philip O. Ogunbona, “Modelling Context and Syntactical Features for Aspect-Based Sentiment Analysis,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3211–3220, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[54] Hu Xu et al., “BERT Post Training for Review Reading Comprehension and Aspect-based Sentiment Analysis,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 2324–2335. [CrossRef] [Google Scholar] [Publisher Link]
[55] Chen Zhang, Qiuchi Li, and Dawei Song, “Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional Networks,” *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4560–4570, 2019. [CrossRef] [Google Scholar] [Publisher Link]
[56] Ruifan Li et al., “Dual Graph Convolutional Networks for Aspect-based Sentiment Analysis,” *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 6319-6329, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[57] D. Dhinakaran et al., “Leveraging Semi-Supervised Graph Learning for Enhanced Diabetic Retinopathy Detection,” *SSRG International Journal of Electronics and Communication Engineering*, vol. 10, no. 8, pp. 9-21, 2023. [CrossRef] [Google Scholar] [Publisher Link]