

# A Comparative Study on the Performance of Deep Learning Algorithms for Detecting the Sentiments Expressed in Modern Slangs



Vivank Sharma, Shobhit Srivastava, B. Valarmathi, and N. Srinivasa Gupta

**Abstract** Sentiment analysis is a text investigation technique that distinguishes extremity inside the text, regardless of whether an entire document, sentence, etc. Understanding individuals' feelings are fundamental for organizations since customers can communicate their considerations and emotions more transparently than any other time in recent memory. In this paper, the proposed model is the sentimental analysis on Twitter slangs, i.e., tweets that contain words that are not orthodox English words but are derived through the evolution of time. To do so, the proposed model will find the root words of the slangs using a snowball stemmer, vectorizing the root words, and then passing it through a neural network for building the model. Also, the tweets would pass through six levels of pre-processing to extract essential features. The tweets are then classified to be positive, neutral, or negative. Sentiment analysis of slangs used in 1,600,000 tweets is proposed using long short-term memory (LSTM) network, logistic regression (LR), and convolution neural network (CNN) algorithms for classification. Among these algorithms, the LSTM network gives the highest accuracy of 78.99%.

**Keywords** Snowball stemmer · Classifiers · Sentiment analysis · LSTM · CNN · Logistic regression · Twitter · Word2Vector

---

V. Sharma · S. Srivastava

Department of Information Technology, Vellore Institute of Technology, Vellore, Tamil Nadu, India

e-mail: [vivanksharma@ymail.com](mailto:vivanksharma@ymail.com)

S. Srivastava

e-mail: [shobhit.sri0108@gmail.com](mailto:shobhit.sri0108@gmail.com)

B. Valarmathi (✉)

Department of Software and Systems Engineering, School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India

e-mail: [valargovindan@gmail.com](mailto:valargovindan@gmail.com)

N. Srinivasa Gupta

Department of Manufacturing, School of Mechanical Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India

e-mail: [guptamalai@gmail.com](mailto:guptamalai@gmail.com)

## 1 Introduction

The Web has significantly changed how individuals express their perspectives and assessments. Sentiment analysis refers to the task of recognizing suppositions, positivity decisions, and other data identified with the emotions and frames of mind communicated in ordinary language writings. Some of the emotions are happy, sad, frustrated, angry, and so on.

Slang is the utilization of exceedingly casual words, shortened forms, and articulations that are dismissed when asked to be taken as a significant aspect of the traditional language. The flood of online communications, for example, electronic mail, text messages, and microblogs administration made use of Internet slang practically omnipresent. It has turned out to be critical to gauge feeling extremity of the opinion or sentiment arranged slang present on the Internet that shows up in the investigation or analysis. Opinion analysis attempts to choose the sort of opinion (affirmative or not) in a given content. Emotion or feeling-based classification has a couple of basic characteristics, including different tasks, highlights, strategies, and context. A new strategy is proposed to classify the sentiment expressed in the microblog slangs in this paper. At that point, the proposed model offers a technique to aid in deciding the extremity available in Slangs.

## 2 Related Works

Almost all existing sentiment analysis algorithms to date are designed in such a way, that it classifies any content into two binary class, i.e., either it is positive or else negative [1–3]. Few recently emerged algorithms were designed in such a way that it extends binary classification to multipoint rating system, i.e., rating inference or multi-category problem [4–6]. Recently few researchers have proposed models for analyzing the reviews of a product or service with respect to all its features [7, 8]. For slangs, a slang dictionary is used to calculate the TF-IDF score to decide the polarity [9]. They made an underlying stride at programmed identification and recognizable proof of slang from normal sentences utilizing deep learning techniques. They showed how phonetic highlights joined with deep learning algorithms offer interpretability. They found that the bidirectional LSTM with feature-based inputs and character-based convolutional embeddings utilizing multilayer perceptron yields the best performance in position recognizable proof, and the model with comparative components aside from with conditional random field has better execution in distinguishing regardless of whether a source sentence contains a slang term [10]. Proposed Twitter information to identify depression [11].

Better data pre-processing methods like changing over emojis to message structure, changing overstretched words to normal form, etc. were utilized to improve the identification accuracy. They extracted the highlights utilizing BOW, TF-IDF with n-grams, and Word2Vec procedures and used these highlights to the methods of

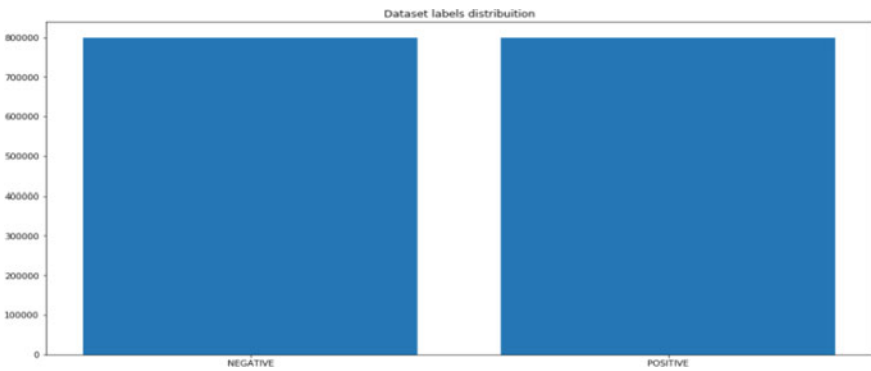
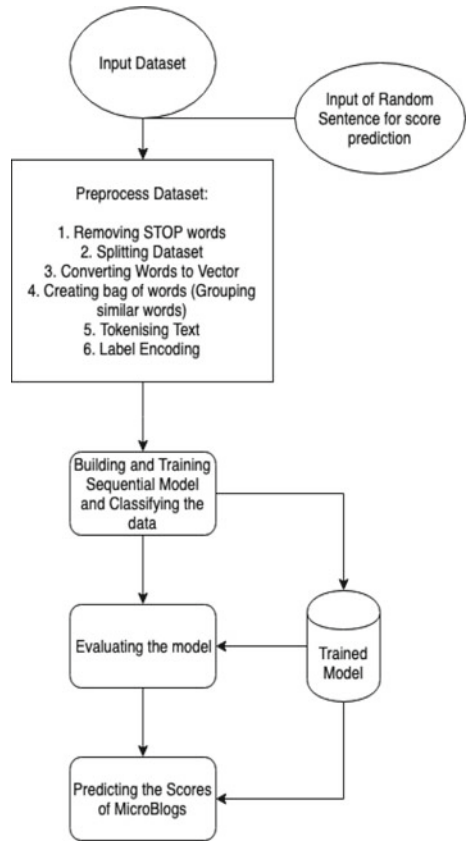
classification. Logistic regression classifier (TF-IDF with n-grams) provided a 81% of the most extreme accuracy. Proposed a novel method called Representative Term-Document Matrix (RTDM) [12]. The given text document was transformed into a vector consisting of eight terms like bad, very bad, disgusting, never recommended, good, very good excellent, and recommended. A classification method is described using the Mahalanobis Distance (MD) [13]. The classifier name was Mahalanobis Distance Classifier (MDC). For 25,000 movie reviews, MDC achieved a 70.8% of accuracy. The hybrid classifier (MDC + MLP) performed with 98.8% of accuracy for 25,000 movie reviews. Deep learning technique is utilized to figure out sentiment analysis issues, for example, sentiment polarity [14]. Models utilizing term frequency-inverse document frequency (TF-IDF) and word embedding were applied to various datasets like Sentiment140, IMDB Movie Reviews, Cornell Movie Reviews, etc. CNN, DNN, and RNN were used in this article.

### 3 Proposed Method

Current algorithms for sentimental analysis work great on formal English literature, but it fails when it comes to slangs. As slangs have no definite list and have no exact meaning, it is all based on contextual and scenario. So, the proposed model, a new algorithm, is developed which not only focuses on keywords but also takes into consideration every single word and tries to overcome slang by making custom datasets and using stemmers to find its original keyword and meaning. Hence, it makes our model different in the form and capability that it can also detect and analyze informal English, which is mostly used in microblogs. For example, for love, one can write many variants which are not in the English dictionary like “Luv,” “Lub,” etc. and the same one word can be even used to describe sentences also. “Shoulda” is replaced by should have. Similarly “lol” is replaced by “laugh out loud.” So, to overcome this, the proposed model is taking more than 1 million Twitter data into account and training our model over these slangs and hence predicting it. Our model is achieving accuracy on the range of 70–80% on different datasets of different microblogs. The proposed model analyzes the microblogs by calculating a score by considering every word and slang and a sentiment score is calculated on a scale of 0-1 to assign sentiment label. This can aid an organization to better comprehend the social sentiment of their service or their product by analyzing online conversations.

The first step is to preprocess the dataset. The second step is building and training the sequential model. The third step is to evaluate the model. The next step is to predict the score of Microblogs. The proposed model is summarized in the flowchart given in Fig. 1. The sentiment 140 dataset can be found in the link <https://gofile.io/d/YS7U1r>. It consists of 1,600,000 tweets that were obtained through the Twitter API. 1,280,000 tweets are used for the training and 320,000 tweets are used for testing. In the dataset, the class label “0” represents a negative tweet, whereas “4” represents a positive tweet. This can be used to detect sentiments [15]. The dataset label distribution which is depicted in Fig. 2 displays that both the positive and

**Fig. 1** Proposed work flow chart



**Fig. 2** Dataset distribution

negative labels are almost equal to the preprocessed data by removing the stop words. Content may contain stop words like “the,” “is,” “are.” Stop words can be separated from the material to be prepared. There is no complete rundown of stop words in natural language processing inquire about; anyway, the NLTK module holds a group of stop words. All the stop words present in the dataset are removed. Then, the proposed model has used snowball stemmer for reducing inflected words in the dataset to their word stem, base, or root form. A stemmer is a calculation that works on the guideline of perceiving “stem” words implanted as such. These are useful for lexical purposes, for instance, in online lexicons, for heuristics in a record the board, or anyplace else that semantic apparatuses can help make a request.

Stemmers get the consideration of a center or stem word inside a more drawn out term. For instance, a stemming calculation may take a gander at a name like “planning,” and accurately perceive that the root word or stem word is “plan.” This can be a useful component of something that parses crude content for investigation, either for a site or some other venture. Snowball stemmer is famous for its compatibility with slangs. The next step is to convert all the text to vectors. These vectors are used by deep learning models over millions of words. The Word2Vec model helps in considering the context for further processing.

Word2Vec is a gathering of models which infers relations between a word and its relevant words. The two significant models inside Word2Vec are skip gram and continuous bag of words (CBOW).

The proposed model takes an inside word and a window of setting (neighbor) words and the model endeavor to foresee setting terms out to some window estimate for each middle name in skip gram model, Along these lines, our model will characterize a likelihood appropriation, for example, the likelihood of a word showing up in the setting given an inside word, and the next step will pick our vector portrayals to expand the possibility.

CBOW is just the inverse of the skip gram model. Try to forecast core-word by adding, vectors of neighboring words.

Skip gram as well as CBOW model output is shown in Fig. 3.

An example of a data pre-processing model is given in Fig. 4.

After this, tokenize the text, and then label encodes it for the further training process. Now, built the sequential model and it consists of four layers. Figure 5 shows the sequential model summary.

In the proposed model, the sigmoid activation function is used for the dense hidden layer. A sigmoid function exists between (0–1) and hence is the best model for predicting probability. The formula for the sigmoidal function is shown in Eq. (1).

```
similarity between 'alice' and 'wonderland' - CBOW : 0.9994316
similarity between 'alice' and 'machines' - CBOW : 0.99209344
similarity between 'alice' and 'wonderland' - Skip Gram : 0.9007044
similarity between 'alice' and 'machines' - Skip Gram : 0.86795944
```

**Fig. 3** Example of data processed with skip gram as well as CBOW model

```
w2v_model.most_similar("love")

/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:1: DeprecationWarning: Call to deprecated 'most_similar' (Method will be removed in 4.0.0, use self.wv.most_similar() instead).
  """Entry point for launching an IPython kernel.
2019-02-17 08:50:44,280 : INFO : precomputing L2-norms of word weight vectors

[('luv', 0.5840025544166565),
 ('loves', 0.5525496602058411),
 ('loved', 0.5403332710266113),
 ('adore', 0.5374413728713989),
 ('looove', 0.5025960206985474),
 ('amazing', 0.494439959520062),
 ('loooove', 0.47303086519241333),
 ('awesome', 0.46765822172164917),
 ('loveee', 0.456807404756546),
 ('lovee', 0.45561471581459045)]
```

Fig. 4 Data pre-processing grouping similar words

```
-----
Layer (type)                Output Shape                Param #
-----
embedding_1 (Embedding)     (None, 300, 300)          87125700
dropout_1 (Dropout)         (None, 300, 300)          0
lstm_1 (LSTM)                (None, 100)                160400
dense_1 (Dense)              (None, 1)                  101
-----
Total params: 87,286,201
Trainable params: 160,501
Non-trainable params: 87,125,700
-----
```

Fig. 5 Sequential model summary

$$\sigma(x) = 1/(1 + \exp(-x)) \tag{1}$$

Now, finally train the proposed model on the dataset, which consists of 1,600,000 tweets that were obtained through the Twitter API. In the dataset, the class label “0” represents negative tweets, whereas “4” represents positive tweets. This can be used to detect sentiments [15]. So, that it can further be used for predicting scores of the content to be analyzed.

### 4 Result and Discussion

On evaluating the test dataset, the proposed model gets an accuracy of 78.99% (LSTM Network), which is excellent compared to earlier achieved accuracy of 56.58–76.69% considering slangs.

Dang et al. [14] used sentiment140 dataset. Accuracy of DNN, CNN, and RNN algorithms with TF-IDF was 76.50%, 76.69, and 56.58. Among these algorithms, CNN was performing well. In the proposed model, the LSTM network, logistic regression, and CNN algorithms are used for processing. Among these algorithms, LSTM network was performing well. LSTM network gets an accuracy of 78.99%. Figure 6 shows the training, as well as the validation accuracy graphs.

Figure 7 shows the training and validation loss graph.

The confusion matrix is shown in Fig. 8. Hence from the confusion matrix, true positive, false positive, true negative, and false negative values are shown as true positive is 0.77, false positive is 0.19, true negative is 0.81, and false negative is 0.23.

Fig. 6 Training and validation accuracy graph

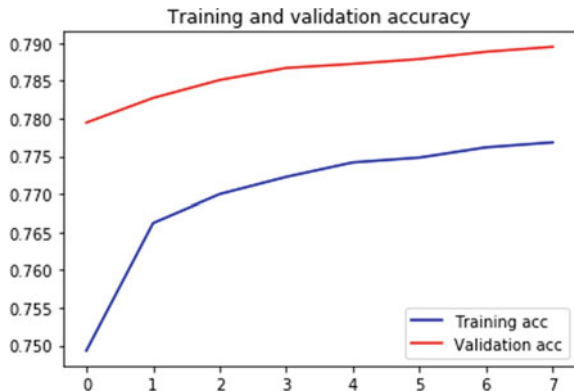


Fig. 7 Graph for training and validation loss



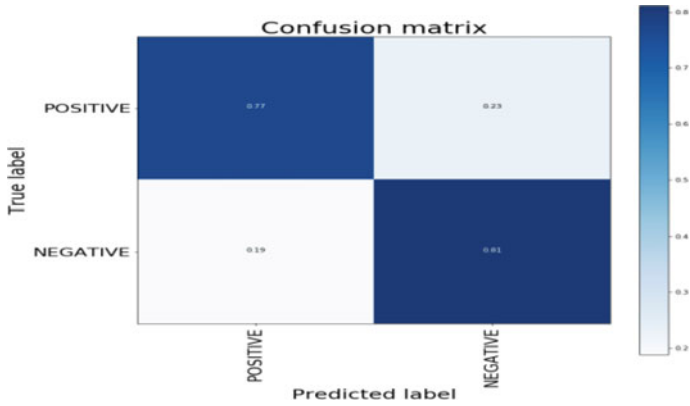


Fig. 8 Confusion matrix

This shows that our model is predicting scores and classifying sentences containing multiple slangs with very fewer errors. The proposed model is performing quite well-considering slangs in the dataset [16].

Now, let us predict some text and see how well our model is doing. Figures 9, 10, 11 and 12 show some examples of the score predicted by our model.

Hence, our model is giving scores from 0 to 1 up to 8 decimal places which can be further extended to rating inference from 0 to 10, where 0 being worst and ten being the best and can be applied on microblogs for analyzing and mining contents. Though our model performing great, there are still some future works that can be done like

```
predict("I love the music")  
  
{'label': 'POSITIVE',  
 'score': 0.963658332824707,  
 'elapsed_time': 0.4383208751678467}
```

Fig. 9 Predicted score of “I love music”

```
predict("is upset that he can't update his Facebook by texting it... and might cry as a result  
School today also. Blah!")  
  
{'label': 'NEGATIVE',  
 'score': 0.03207983821630478,  
 'elapsed_time': 0.2546839714050293}
```

Fig. 10 Predicted score of “is upset that he can’t update his Facebook by texting it ... and might cry as a result School today also. Blah!”



```
predict("I love the music")

{'label': 'POSITIVE',
 'score': 0.963658332824707,
 'elapsed_time': 0.4383208751678467}
```

**Fig. 11** Predicted score of “I love the music”

```
predict("I hate the rain")

{'label': 'NEGATIVE',
 'score': 0.015783820301294327,
 'elapsed_time': 0.2579636573791504}
```

**Fig. 12** Predicted score of “I hate the rain”

**Table 1** Comparison of the accuracies of various classifiers of the proposed method

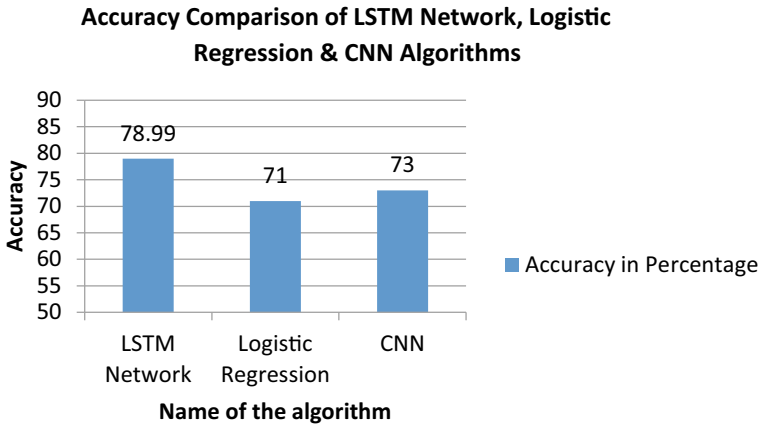
S. No.	Name of the classifier	Accuracy (%)
1	LSTM network	78.99
2	Logistic regression	71
3	CNN	73

extending its from microblog to excellent contents like books, documentary, etc. and further improving it for multiple grams content.

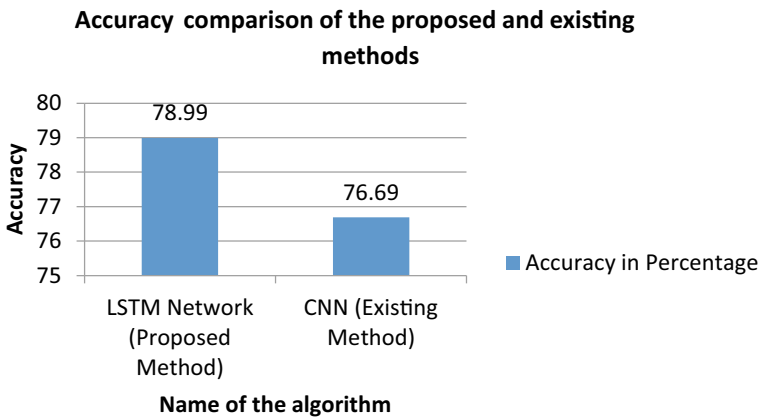
Perform a sentimental analysis of Twitter slangs using LSTM Network, logistic regression, and CNN. Table 1 shows a comparative performance of the different techniques or classifiers used in the proposed method. Figure 13 shows accuracy and comparison of the LSTM network, logistic regression and CNN algorithms. Among these algorithms, the LSTM network gives the highest accuracy of 78.99% and it is shown in Fig. 13. The accuracy comparison of the proposed and existing methods is shown in Fig. 14.

## 5 Conclusion and Future Scope

The proposed work utilized Twitter data (sentiment140 dataset) to identify sentiment. After pre-processing the data, develop an algorithm that not only focuses on keywords but also takes into consideration every single word and tries to overcome slang by making a custom dataset and using stemmers to find its original keyword and meaning. Next, trained and tested the model by using various algorithms like long short-term memory network, logistic regression, and convolution neural network algorithms. Among these algorithms, the long short-term memory network gives the



**Fig. 13** Accuracy comparison of the LSTM network, logistic regression and CNN algorithms



**Fig. 14** Accuracy comparison of the proposed and existing methods

highest accuracy of 78.99%. In the future, to increase the accuracy of the proposed model, use bidirectional encoder representations from transformers and embeddings from language models algorithms for classification.

## References

1. Dave K, Lawrence S, Pennock DM (2003) Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: WWW '03: proceedings of the 12th international conference on World Wide Web, pp 519–528

2. Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? Sentiment classification using machine learning techniques. In: EMNLP '02: Proceedings of the ACL-02 conference on empirical methods in natural language processing. Association for computational linguistics, pp 79–86
3. Turney PD (2002) Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: ACL '02: Proceedings of the 40th annual meeting on association for computational linguistics, pp 417–424
4. Goldberg AB, Zhu X (2006) Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In: Text Graphs '06: proceedings of text graphs: the first workshop on graph based methods for natural language processing on the first workshop on graph based methods for natural language processing, association for computational linguistics, pp 45–52
5. Ki Leung CW, Fai Chan SC, Lai Chung F (2006) Integrating collaborative filtering and sentiment analysis: a rating inference approach. In: ECAI 2006 workshop on recommender systems, pp 62–66
6. Pang B, Lee L (2005) Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: ACL'05: proceedings of the 43rd annual meeting on association for computational linguistics, pp 115–124
7. Gamon M (2004) Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In: COLING '04: proceedings of the 20th international conference on computational linguistics, pp 841
8. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: KDD '04: proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 168–177
9. Manuel K, Varma K, Radha I, Krishna P (2010) Analyzing internet slang for sentiment mining. In: IEEE, 2010 second vaagdevi international conference on information technology for real world problems
10. Sun Z, Xu Y (2019) Slang detection and identification. In: Proceedings of the 23rd conference on computational natural language learning, pp 881–889
11. Mounika M, Srinivasa Gupta N, Valarmathi B (2020) Detection of depression related posts in tweets using classification methods—a comparative analysis. In: Proceeding of the international conference on computer networks, big data and IoT (ICCBI-2019), vol 49, pp 620–630
12. Srinivasa Gupta N, Valarmathi B, Sanju J (2012) Sentiment analysis using representative terms—a grouping approach for binary classification of documents. *J Theor Appl Inf Technol* 44:161–165
13. Valarmathi B, Srinivasa Gupta N, Palanisamy V (2016) Mahalanobis distance—the ultimate measure for sentiment analysis. *Int Arab J Inf Technol* 13:252–257
14. NhanCach D, María NM-G, Fernando De la P (2020) Sentiment analysis based on deep learning: a comparative study. *Electron Open Access J* 9:1–29
15. Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, vol 1
16. <https://towardsdatascience.com/sentiment-analysis-of-transliterated-texts-in-hindi-and-marathi-languages-c29eab19fae2>