International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Identification and Categorization of SMS using Deep Learning and Machine Learning Methods

Sinkon Nayak*[1], Manjusha Pandey[2], Siddharth Swarup Rautaray[3]

Submitted:12/03/2024 **Revised**: 27/04/2024 **Accepted**: 04/05/2024

Abstract: Text messages are short messages that can be used for personal as well as professional ways to share messages without involving the internet as a mode of communication. There are some essential text messages and some are nonessential. It is crucial to filter out the nonessential messages from the essential ones. Various machine learning and deep learning methods are used to categorize the text messages. This research work uses various machine learning and deep learning methods to categorize them. To extract the features from the text messages this study uses word embedding and contextual embedding techniques. Finally, the measurement of the performances is done with the help of performance matrices and confusion matrix parameters. For the word embedding-based feature selection method the Extra Tree and LSTM are more accurate i.e. 96.86% and 98.06%. And for the sentence embedding-based feature selection method the SVM and Bi-directional LSTM are more accurate i.e. 99.1% and 99.19%.

Keywords: *Text message Categorization, Machine Learning, Deep Learning, Word Embedding, Contextual Embedding*

1. Introduction

Text messages (TM) are the short form of messages sent from one handheld device to another. TMs are the simplest way to communicate with each other with the help of a digital handheld device[1]. The handheld device can be mobile phones, laptops, desktops, or any other congenial device. TM is used to communicate with friends, family, business, and societal reasons. TM can be used for personal as well as professional ways to share messages in short form. TM consists of letters, alphanumeric characters, or numbers. The TM can be used to communicate with two or more users. It is also used by businesses and organizations for official messaging purposes. This is referred to as digital or business messaging with the help of a set of channels. Through digital messaging, they can connect with their customers, employees, and students in the educational sector instantly. The main advantage of the TM is it does not require immediate response from the receiver but security is the main concern as the messages are not end to end encrypted. TM can be a simple text message which is called a Short Message Service(SMS). When a multimedia message is involved in the text message it is called a Multimedia Messaging Service(MMS) and when it contains digital pictures, video clips, and audio clips, as well as ideograms known as emoticons it is called Rich Communication Service(RCS). TMs are used wherever and whenever voice calling is not workable. TMs are a more brief description of a communication and that is why it is known as a short message as well[2]. There are other kinds of messaging services like push messages which are a kind of pop-up message, and instant messages which are done through the internet and in-app messaging. Due to the lack of encryption system in the TM, the message can be easily attached and interrupted. In this study, we have used various Machine Learning Techniques(MLTs) and Deep Learning Techniques(DLTs) to classify the simple text messages either essential or non-essential ones. Essentials are the ones that we send or receive from our friends, family, or colleagues and non-essential ones are the one that consists of advertisements or company messages.

2. Related Work

A novel method was proposed[3] for the identification of unwanted SMS detection with the help of CNN and LSTM. They have tested their proposed approach by analyzing their performances with the help of numerous ML models. The proposed architecture was based on hybridizing CNN and LSTM for English SMS data that is available publicly and Arabic SMS data that was collected by them from a local phone. Finally, they concluded that their proposed hybrid approach is performing better than the individual performance of the models. To find out the toxic messages and get the sentiment out of them, in [4] topic modeling and sentiment analysis are used. They perform analysis on one of the publicly available data, one data collected from their university students, and data augmentation methods. They analyze the performance with the help of three models SVM, LSTM, and adversarial domain adaptation. The analysis is performed on Russian text message data. Phishing attack identification on SMS data is done[5] with the help of numerous machine-

[1] *Research Scholar, Kalinga Institute of Industrial Technology, School of Computer Engineering, Bhubaneswar-751024, Odisha, India.*
[2,3] *Associate Professor Kalinga Institute of Industrial Technology, School of Computer Engineering, Bhubaneswar-751024, Odisha, India.*
** Corresponding Author Email: [1]sinkonnayak07@gmail.com*
[2]manjushafcs@kiit.ac.in, [3]siddharthfcs@kiit.ac.in

learning methods. The main goal of their study is to blacklist the unwanted URLS and customized code words used in the SMS for the efficient use of their proposed method.

Short message topic modeling is used for the analysis of SMS sent by therapist and they tested their method of proposal by setting various hyper-parameters. Out of which Latent Feature Dirichlet Multinomial Mixture with a value setting $\alpha = 0.1$, $\beta = 0.01$, and $K = 8$ turns out to be the best one. They tested their proposed method initially on 28 SMS and then finally on 53 SMS. The topic is mostly related to energy recharge, locus of control, mutual respect, schedule activity, handling uncertainty, medium of communication, management of health and thoughts, hope, and readiness[6]. For the unwanted SMS classification[7] a multi-channel-based CNN method. The feature extraction is done by static and dynamic word embedding methods. They have taken the data of SMS from public data and also to test the efficiency of their proposed method they have collected some amount of data by data augmentation method. In[8] a GPT-3 based method has been proposed for the identification of unwanted SMS from public data. They have tested their approach with boosting, bagging, stacking, and voting classifiers by implementing SVM, KNN, CNN, and Light GBM.

UCI spam base dataset is used to classify junk mail and non-junk mail in Iqbal, K., Khan, M. S. To pick out the feature Point-Biserial correlation is taken into consideration and for performance estimation 10 fold cross-validation is considered. They analyzed 8 identical methods and came to the conclusion that Support Vector machines and Artificial neural networks provide greater results as compared to others. For the identification of junk and non-junk mail, they conducted their study on various tree-based and distance-based methods[9]. As email is the most used mode of communication for professionals unused junk mail causes a waste of time and resources. According to Mansoor, R. A. Z. A., et al. supervised methods are widely used to filter out junk mail from the essential ones out of which Naive Bayes and Support Vector Machine are the ones that provide better results in terms of accuracy. More focus is required on junk mail which is in the form of hyperlinks, images, and attachments[10].

A detailed survey analysis of the articles published from 2006 to 2016 is done by Mujtaba, G., et al. which includes mail categorization, data sets, feature space, classification techniques, and the criteria to evaluate their performance. They provide a detailed study of the most popular data sets, feature sets, and learning methods. They also mentioned the issues and problems faced in the email categorization. As it is a widely used mode as compared to social media, communicating applications, and mobile SMS, the management of it needs to be automated[11]. Through analysis of machine learning methods and the dataset used and criteria to evaluate performance for the same were discussed by Dada, E. G., et al.. They describe the learning methods used to date for mail identification by the service providers of emails. According to their study usage of deep learning and deep adversal learning models can be used in the identification task for better outcomes [12].

Machine learning methods were used by Alurkar, A. A., et al. to detect a sketch of an uninteresting phrase. Cc/Bcc, domain, and header are the other parameters that are considered as a feature for their study. They highlighted the issue with the traditional method of identification that uses only a set of keywords to filter out the unrequired mail and the failure of the traditional method is due to an increase in the usage and advanced methods used by attackers[13]. By taking three identical departments of one of the enterprise data Huang, J. W., et al. classify mail by using text mining and machine learning approaches to hinder the leakage of business resources. They proposed a methodology using bi-grams and paragraph vectors. Their proposed method is implemented in an enterprise for email management[14]. Karim, A., et al. discussed artificial intelligence techniques and machine learning used to decipher email phishing. The mail header, mail sender's and receiver's identity and their respective sources, the body that consists of from, to, date, and subject, and text and attachments are taken into consideration[15].

CSDMC2010 spam data and CACBANK dataset are used by Alamlahi, Y., & Muthana, A. to classify emails by using ANN. The first data set is publicly accessible and the second one is private data. They have proposed a framework for filtration by extracting the features from the data sets using PCA and finally applied a layer ANN network to do the filtration job[16]. To deal with numerous issues faced by junk mail Aski, A. S., Sourati, N. K. came up with a method to filter out the junk by using Multi-Layer Perceptron, Decision Tree, and Naive Bayes and evaluate their performance. The proposed approach claims to deal with web traffic, memory, computation power, speed wastage, and finally monetary loss due to junk mail[17]. Olatunji, S. O. came up with a Support Vector Machine approach to recognize junk and non-junk mail and shows that performance is best as compared to others. As the spammers have evolved with their spamming methods the traditional methods fail to recognize the unwanted emails. So to deal with it the author proposed a SVM based approach to recognize them[18].

3. Dataset Description

The data set we have considered SMS data from the UCI repository which is the combination of SMS data from 4 sources as shown in Table 1. The compound data set of

SMS consists of 5574 messages and has 2 columns, a label for the class category and another for raw messages[19].

Table 1. Dataset Description

Source of Messages	Total Messages
The Grumbletext Web site	425
NUS SMS Corpus (NSC)	3375
Caroline Tag's Ph.D. Thesis	450
Spam Corpus v.0.1 Big	1324
Total	5574

4. Word Embedding

As machines do not understand the text it is necessary to convert the text data into numeric representation. The numeric representation is done in such a way that it represents the text in a unique way. To represent the text in numeric form a bunch of numbers is used and they are denoted as vectors. This vectorized representation of text is done in such a way that the similar numbers have a similar type of vectorized form of representation. In this segment, we have used the GloVe method for vectorize representation of text messages[6]. The GloVe is defined as global vectors for word representation that follow the matrix factorization method. To capture the context of the words it uses global statistics. This method solves the issue of out-of-vocabulary problems. GloVe calculates the semantic measurement based on the content of the text. The word2vec method performs better in larger training samples whereas the GloVe performs better in smaller training samples as well[20]. This is the reason we have used the GloVe word embedding technique for feature extraction.

5. Proposed Model using Word Embedding

Text message data required to do preprocessing on it. The first step of the preprocessing task is to check for null values and duplicate values is done. If any of them is present it is required to drop them as they do not contribute anything to the data set. The second step is to convert all the mail data into lower case and tokenization is done. In tokenization for text categorization, the plain text is converted into tokens which can be characters, numbers, or words. In the third step, stop word removal, special characters, and removal of punctuation are done as they are not significant for the categorization task. After preprocessing the data it is required to convert the textual data into a machine understandable format. This process is called feature engineering. The word embedding method is used in this study. Word embedding is used to solve the word semantics. It converts the word according to the meaning of the word into a vector form. It also captures the semantics between words that are neighboring to each other. In this segment, the GloVe method is used to encode words into vectorized format. These vectors then feed to the classification methods for categorization to carry out. To categorize the SMS 11 classification ML and DL models are used and finally, the performance with the help of various matrices is displayed in Figure 1.

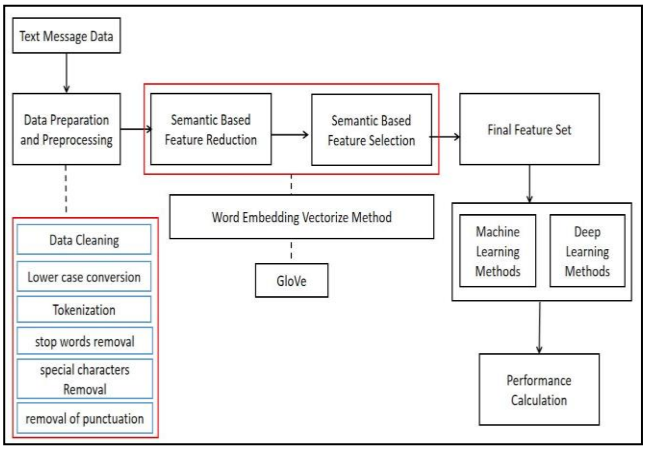

Fig. 1. Proposed Methodology using Word Embedding

6. Contextual Embedding

To capture the similarity between words BERT is used. BERT refers to Bidirectional Encoder Representations from Transformers. In the word embedding technique the words have fixed embedding which is not feasible in some cases where the meaning of a word changes depending upon the context of a sentence. In some cases, the same word can have different meanings. To solve this issue of word embedding we need a better technique that can able to capture the meaning of a word depending upon the context of a word. The technique can look at the sentences and based on that it generates the vector representation for a word so that the issue with word embedding can be resolved. BERT can do that. It can generate the vector or numeric representation of a word or sentence depending on the context of the sentence. BERT has the ability to generate embedding for the entire sentence. BERT generates a vector of size 768[21]. BERT is a transformer-based architecture and has 2 versions. The first version is BERT BASE which uses 12 encoder layers and the second version is BERT LARGE which uses 24 encoder layers[21]. It uses special tokens called CLS at the beginning of a sentence and SEP at the end of a sentence. BERT is trained by Google on 2500 million words in Wikipedia and 800 million on different books. Using a masked language model and next-sentence prediction approaches BERT is trained[21]. By multiplying the embedding matrix with the input text each of the transformer layers generates a sequence of word embeddings and applies a series of operations to compute new contextualized embeddings. The next transformer layer takes these contextualized embeddings as input and this process repeats until the vector representation of the text is not generated. Figure 2 is the illustration of the contextual embedding workflow.

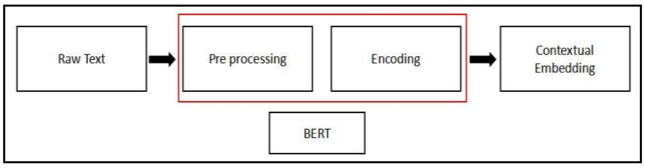

Fig. 2. Contextual Embedding

7. Proposed Model using Contextual Embedding

Text message for one to one communication after getting the data it is required to do preprocessing on it. BERT is trained by Google on 2500 million words in Wikipedia and 800 million on different books but the training is not done in a domain specific way it is an essential task to fine-tune the model as embedding is required to do contextualized further analysis. BERT has its own preprocessing packages to preprocessing to make the text compatible with it. Tokenization is done by using WordPiece tokenization that breaks words into smaller sub-words if required. This allows to build model vocabulary more efficiently[21]. Adding special tokens like CLS and SEP as discussed in the previous segment. The text is padded with special tokens or shortened to a fixed length because BERT needs fixed length inputs. To mark the separate different sentences segment ID is used that denotes the sentence origin. MASK token is used to learn to predict missing words based on the context. After preprocessing the model, fine tuning is needed. Fine tuning is done by using a masked language model and next sentence prediction[21]. By considering the adjacent words BERT makes a prediction of the masked word in both directions for a concealed word through the masked language model and the next sentence prediction is used to find out the interrelation between two sentences to predict the upcoming sentence. This task is done simultaneously and finally, BERT generates contextual embedding for SMS data. These embedding vectors then feed to the classification methods for categorization to carry out. To categorize the SMS 11 classification ML and DL models are used and finally, the performance with the help of various matrices is displayed in Figure 3.

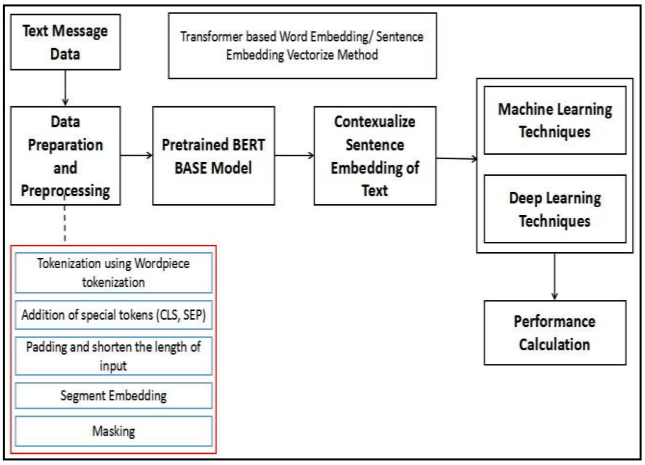
Text Message Data
Transformer based Word Embedding/ Sentence Embedding Vectorize Method
Data Preparation and Preprocessing
Pretrained BERT BASE Model
Contexualize Sentence Embedding of Text
Machine Learning Techniques
Deep Learning Techniques
Performance Calculation
Tokenization using Wordpiece tokenization
Addition of special tokens (CLS, SEP)
Padding and shorten the length of input
Segment Embedding
Masking

Fig. 3. Proposed Methodology using Contextual Embedding

8. Results

The compounded SMS data is used in this segment for the categorization of SMS. GloVe is used to convert the text SMS data into embedding vector form. The vectors are fed to the MLTs and DLTs for categorization. Finally, the performance is calculated by using various matrices. Table 2 displays the performance of the proposed framework w.r.t MLTs. Table 3 is the illustration of the performance of compound data w.r.t DLTs.

Table 2. Performance of MLTs

MLTs	Accuracy(%)	Error(%)	Precision(%)	Recall (%)	F1-Score
Logistic Regression	90.22	9.78	89.86	100	94.65
Naive Bayes	86.63	13.37	86.63	100	92.83
Support Vector Machine	93.27	6.73	93.12	99.6	96.34
KNN	94.79	5.21	96.7	97.3	97
Decision Tree	95.7	4.3	96.83	98.24	97.53
Random Forest	96.59	3.41	97.25	98.9	98.1
AdaBoost	95.24	4.75	96.34	98.24	97.3

Bagging	96.68	3.32	97.6	99.17	98.11
Extra Tree	96.86	3.14	97.16	99.27	98.2
Gradient Boosting	95.96	4.04	96.84	98.55	97.7
Soft Voting	96.6	3.4	97.34	98.75	98.04

Table 3. Performance of DLTs

DLTs	Accuracy(%)	Error(%)	Precision(%)	Recall (%)	F1-Score
LSTM	98.06	1.94	96.99	88.96	92.80
Bi-directional LSTM	93.23	6.77	69.03	93.79	79.53

We have represented the performance in a graphical form of MLTs in Figure 4. And Figure 5 is the graphical representation of DLTs.

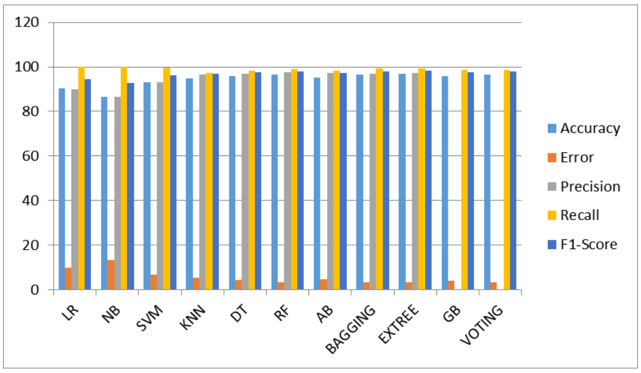

Fig. 4. Performance of MLTs

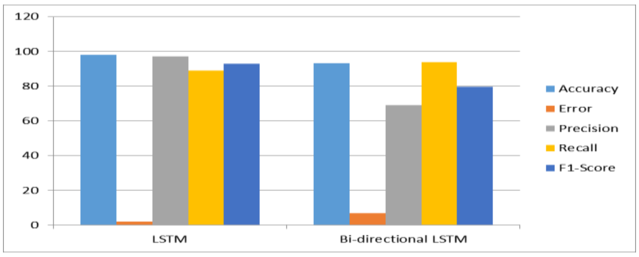

Fig. 5. Performance of DLTs

To add to the comprehensiveness of the study being conducted we also have done a performance analysis of MLTs and DLTs w.r.t True positive, False positive, False negative and True negative in Table 4 and Table 5 respectively.

Table 4. Performance of MLTs

MLTs	True Positive	False Negative	False Positive	True Negative
Logistic Regression	966	0	109	40
Naive Bayes	966	0	149	0
Support Vector Machine	962	4	71	78
KNN	940	26	32	117
Decision Tree	949	17	31	118
Random Forest	955	11	27	112
AdaBoost	949	17	36	113
Bagging	958	8	29	120
Extra Tree	959	7	28	121
Gradient Boosting	952	14	31	118
Soft Voting	954	12	26	123

Table 5. Performance of DLTs

DLTs	Truly Positive	Falsely Negative	Falsely Positive	Truly Negative
LSTM	885	4	16	129
Bi-directional LSTM	828	61	9	136

We have represented the performance in a graphical form of MLTs in Figure 6. And Figure 7 is the graphical representation of DLTs.

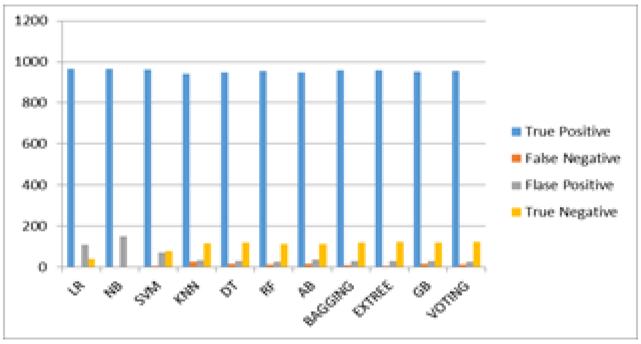

Fig. 6. Performance of MLTs

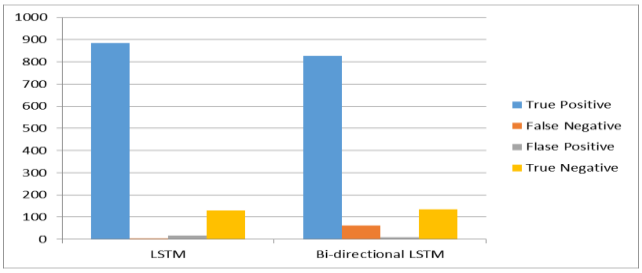

Fig. 7. Performance of DLTs

The compounded SMS data is used in this segment for the categorization of SMS. BERT is used to convert the text SMS data into contextualized vector form. The vectors are fed to the MLTs and Deep Learning Techniques(DLTs) for categorization. Finally, the performance is calculated by using various matrices. Table 6 displays the performance of the proposed framework w.r.t MLTs. Table 7 is the illustration of the performance of compound data w.r.t DLTs.

Table 6. Performance of MLTs

MLTs	Accuracy(%)	Error(%)	Precision (%)	Recall (%)
Logistic Regression	98.48	1.52	90.60	97.83
Naive Bayes	88.25	11.75	28.86	63.24
Support Vector Machine	99.10	0.9	95.30	97.93
KNN	96.68	3.32	86.58	88.36
Decision Tree	94.98	5.02	82.55	80.39
Random Forest	97.40	2.60	81.21	99.18
AdaBoost	98.30	1.70	90.60	96.43
Gradient Boosting	97.31	2.69	81.88	97.60
XGBoost	98.57	1.43	90.60	98.54

Table 7. Performance of DLTs

DLTs	Accuracy(%)	Error(%)	Precision (%)	Recall (%)
LSTM	96.77	3.23	97.32	81.92
Bi-directional LSTM	99.19	0.81	97.32	96.67

Figure 8 and Figure 9 are the graphical representation of performance of MLTs and DLTs for SMS data with respect to accuracy, precision, recall and error rate respectively.

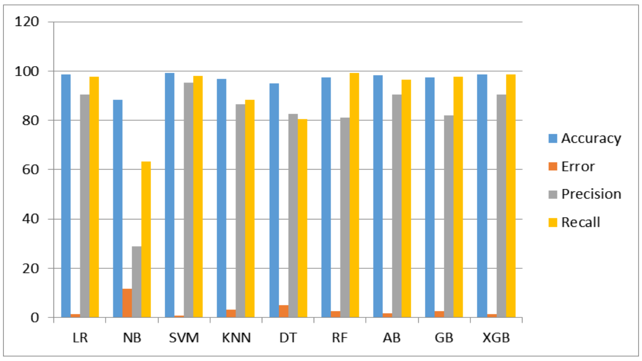

Fig. 8. Performance of MLTs

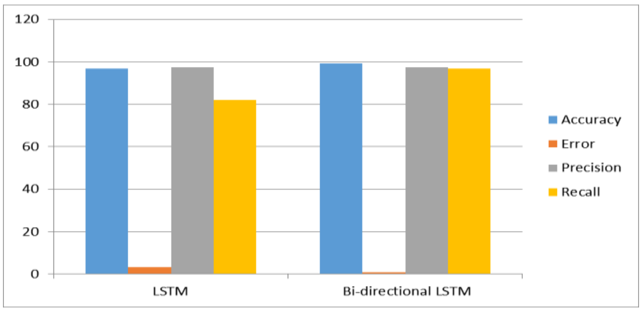

Fig. 9. Performance of DLTs

Table 8 and Table 9 is the illustration of performance of compound SMS data of MLTs and DLTs w.r.t True Positive, True Negative, False Positive, False Negative value respectively.

Table 8. Performance of MLTs

MLTs	True Positive	False Negative	False Positive	True Negative
Logistic Regression	963	14	3	135
Naive Bayes	941	106	25	43
Support Vector Machine	963	7	3	142
KNN	949	20	17	129
Decision Tree	943	27	23	122
Random Forest	965	29	1	120
AdaBoost	961	14	5	135
Gradient Boosting	963	27	3	122
XGBoost	964	14	2	135

Table 9. Performance of DLTs

DLTs	True Positive	False Negative	False Positive	True Negative
LSTM	934	4	32	145
Bi-directional LSTM	965	4	5	141

Figure 10 and Figure 11 is the graphical representation of performance of MLTs and DLTs for SMS data with respect to True Positive, True Negative, False Positive, False Negative value.

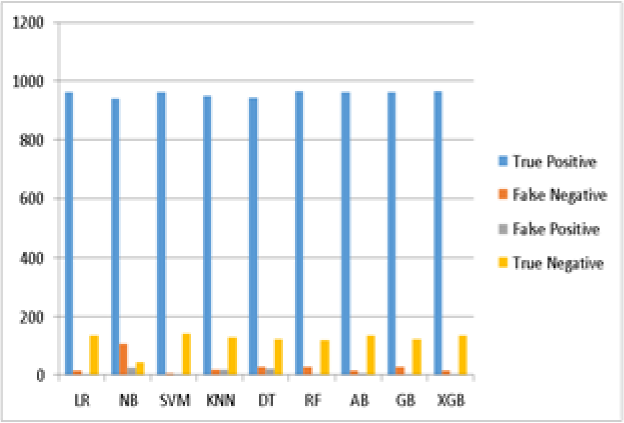

Fig. 10. Performance of MLTs

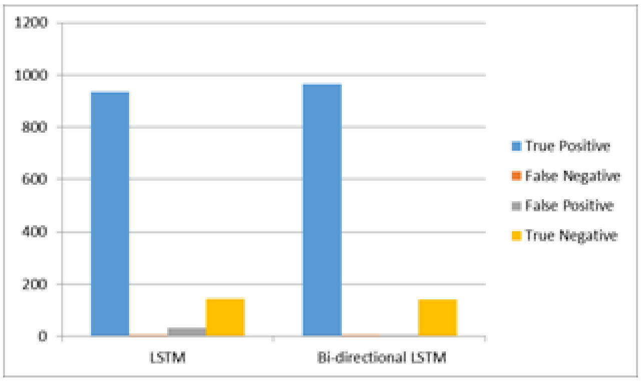

Fig. 11. Performance of DLTs

9. Conclusion

In this study, we have performed analysis on text message data and applied nine MLTs and two DLTs for categorization of text message. The result indicate that the performance of Extra Tree is more accurate among MLTs and among DLTs LSTM performs the best. The accuracy for Extra Tree and LSTM is 96.86 and 98.06. The true positive value predicted for Extra Tree and Bagging are nearly same. And for LSTM the predicted true positive value is 885 and true negative value is 129. And in the similar manner we also performed analysis using BERT for feature extraction. The result shows that the SVM performance is more accurate as compared with other MLTs and for DLTs Bi-directional LSTM performs best. And if we compare the eleven methods applied for text message analysis then the Bi-directional LSTM performance is the most accurate one. As the true positive value affects the accuracy, the true positive value is better for SVM and Bi-directional LSTM. And among these two models Bi-directional model has more true positive value predicted accurately. The accuracy for SVM and Bi-directional LSTM is 99.1% and 99.19% for BERT.

Author contributions

Sinkon Nayak: conceptualized, analyzed and wrote the original draft. **Manjusha Pandey** and **Siddharth Swarup Rautaray:** aided in the conceptualization, supervision, validation, reviewing and editing of the final manuscript. All authors read and approved the final version of manuscript.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Ghourabi, A., Mahmood, M. A., & Alzubi, Q. M. (2020). A hybrid CNN-LSTM model for SMS spam detection in arabic and english messages. Future Internet, 12(9), 156.

[2] Bogoradnikova, D., Makhnytkina, O., Matveev, A., Zakharova, A., & Akulov, A. (2021, May).

Multilingual sentiment analysis and toxicity detection for text messages in russian. In 2021 29th Conference of Open Innovations Association (FRUCT) (pp. 55-64). IEEE.

[3] Dogra, V., Verma, S., Chatterjee, P., Shafi, J., Choi, J., & Ijaz, M. F. (2022). A complete process of text classification system using state-of-the-art NLP models. Computational Intelligence and Neuroscience, 2022.

[4] Adhikari, S. (2020, March). Nlp based machine learning approaches for text summarization. In 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC) (pp. 535-538). IEEE.

[5] Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., & Heck, L. (2016). Contextual lstm (clstm) models for large scale nlp tasks. arXiv preprint arXiv:1602.06291.

[6] Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], 9(1), 381-386.

[7] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning--based text classification: a comprehensive review. ACM computing surveys (CSUR), 54(3), 1-40.

[8] Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., ... & He, L. (2022). A survey on text classification: From traditional to deep learning. ACM Transactions on Intelligent Systems and Technology (TIST), 13(2), 1-41.

[9] Roberts, Daniel P., et al. Precision agriculture and geospatial techniques for sustainable disease control. Indian Phytopathology, pp. 1-19. 2021.

[10] Nunes, Simão AS, et al. “Cities go smart!”: A system dynamics-based approach to smart city conceptualization. Journal of Cleaner Production. pp. 127683. 2021

[11] Agbozo, Ebenezer, and Kamen Spassov. Establishing efficient governance through data-driven e-government. Proceedings of the 11th International Conference on Theory and Practice of Electronic Governance. 2018.

[12] AlSayegh, Ahmed, Chowdhury Hossan, and Bret Slade. Radical improvement of e-government services in Dubai. International Journal of Services Technology and Management 25.1, pp. 53-67. 2019.

[13] Kumar, Shiv, et al. Advance e-governance system. International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS). IEEE, 2017.

[14] Al-Dmour, H., Saad, N., Basheer Amin, E., Al-Dmour, R., & Al-Dmour, A. (2023). The influence of the practices of big data analytics applications on bank performance: filed study. VINE Journal of Information and Knowledge Management Systems, 53(1), 119-141.

[15] Vasa, J., Yadav, H., Patel, B., & Patel, R. (2023). Architecture, Applications and Data Analytics Tools for Smart Cities: A Technical Perspective. In Sentiment Analysis and Deep Learning: Proceedings of ICSADL 2022 (pp. 859-873). Singapore: Springer Nature Singapore.

[16] Samuel, P., Reshmy, A. K., Rajesh, S., Kanipriya, M., & Karthika, R. A. (2023). AI-Based Big Data Algorithms and Machine Learning Techniques for Managing Data in E-Governance. In AI, IoT, and Blockchain Breakthroughs in E-Governance (pp. 19-35). IGI Global.

[17] Bibri, S. E., Krogstie, J., Kaboli, A., & Alahi, A. (2024). Smarter eco-cities and their leading-edge artificial intelligence of things solutions for environmental sustainability: A comprehensive systematic review. Environmental Science and Ecotechnology, 19, 100330.

[18] Gubareva, R., & Lopes, R. P. (2024). Literature Review on the Smart City Resources Analysis with Big Data Methodologies. SN Computer Science, 5(1), 152.

[19] Reference Link-https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

[20] Choudhary, K., & Beniwal, R. (2021, November). Xplore Word Embedding Using CBOW Model and Skip-Gram Model. In 2021 7th International Conference on Signal Processing and Communication (ICSC) (pp. 267-270). IEEE.

[21] Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/