Journal of Analysis and Computation (JAC)
(An International Peer Reviewed Journal), www.ijaconline.com, ISSN 0973-2861
Volume XVIII, Issue II, July-December 2024

TOXIC COMMENT CLASSIFICATION USING NATURAL LANGUAGE PROCESSING

[1]MS. Punita Panwar, [2]Shivam Yadav, [3]Mohak Bardwa, [4]Anchit Parwal, [5]Ishaan Joshi

[1]Assistant Professor, Department of Artificial Intelligence & Data Science, JECRC College
[2]B.Tech Student, Department of Artificial Intelligence & Data Science, JECRC College
[3]B.Tech Student, Department of Artificial Intelligence & Data Science, JECRC College
[4] B.Tech Student, Department of Artificial Intelligence & Data Science, JECRC College
[5]B.Tech Student, Department of Artificial Intelligence & Data Science, JECRC College

ABSTRACT:

Deep Learning has significantly advanced the field of text classification, offering valuable applications. Techniques such as Tokenization, Stemming, and Embedding play pivotal roles in this process. This study explores the application of these techniques alongside various algorithms for classifying online comments based on toxicity levels. A neural network model is proposed for comment classification, and its accuracy is compared with other " models including Long Short Term Memory (LSTM), Naive Bayes Support Vector Machine, Fasttext, and Convolutional Neural Network. Comments undergo tokenization or vectorization initially to generate a dictionary of words, followed by the creation of an embedding matrix. Subsequently, the comments are passed to the model for classification. The proposed model achieves an accuracy of 98.15%.

Keywords- Identification of Toxic Comments, Long Short-Term Memory, Convolutional Neural Networks, Naïve Bayesian Analysis, Support vector Machines, FastText.

[1] INTRODUCTION

Social media platforms have become integral components of modern communication, offering spaces for diverse discussions and interactions. However, the anonymity and wide reach of these platforms often facilitate the proliferation of toxic remarks, which pose significant challenges to maintaining constructive discourse and fostering a safe online environment. Toxic comments, including hate speech, harassment, and other forms of harmful content, not only hinder open expression but also have detrimental effects on

individuals' mental well-being and the overall health of online communities. Therefore, it becomes imperative to develop effective methods to identify and mitigate such toxic content.

In response to this challenge, this research paper explores the application of deep learning techniques, particularly focusing on recurrent neural networks (RNNs) and convolutional neural networks (CNNs), for the classification of online comments based on their toxicity levels. By leveraging natural language processing (NLP) techniques and deep learning architectures, we aim to develop robust models capable of accurately identifying toxic comments in various online platforms.

The need for automated toxic comment classification arises from the sheer volume of content generated on social media platforms, making manual moderation impractical and often insufficient. Automated systems can assist in identifying and flagging potentially harmful content, enabling timely intervention and moderation. However, building effective automated systems requires sophisticated algorithms capable of understanding the nuances of language and context, as toxic comments can manifest in diverse forms and expressions.

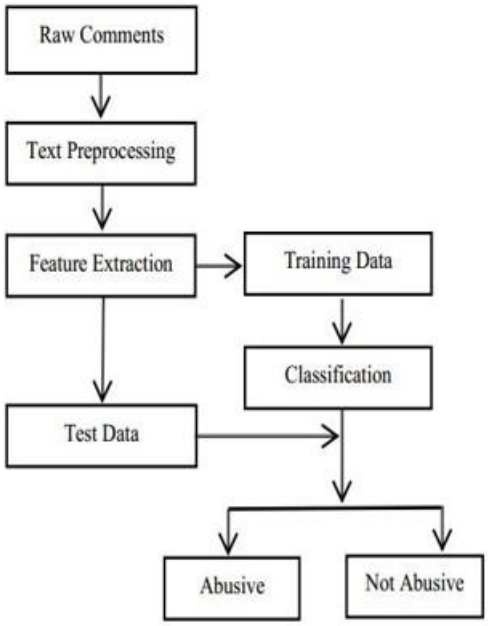

Fig : Flow chart for detecting toxic comments

This paper proposes to address this challenge by employing deep learning models, specifically LSTM networks and CNNs, which have shown promising results in natural language processing tasks. LSTM networks are well-suited for capturing long-range dependencies and contextual information in text data, making them particularly effective in identifying toxic comments that may span multiple sentences or exhibit complex linguistic patterns. CNNs, originally designed for image processing, have been adapted for text

classification tasks, allowing them to effectively extract features from textual data and learn patterns indicative of toxicity.

The primary objective of this research is to compare the performance of LSTM networks, CNNs, and other traditional machine learning algorithms in classifying toxic comments. We aim to evaluate the accuracy, efficiency, and scalability of these models in handling large volumes of textual data from social media platforms.The significance of this research lies in its potential to contribute to the development of more robust and efficient content moderation systems for online platforms. By accurately identifying toxic comments, these systems can help create safer and more inclusive online environments, where individuals feel empowered to express themselves without fear of harassment or discrimination.

Furthermore, this research can provide insights into the effectiveness of deep learning techniques in addressing complex natural language processing tasks, such as toxic comment classification. By understanding the strengths and limitations of different algorithms, we can inform the development of more sophisticated models and strategies for content moderation in the digital age.

[2] RELATED WORK

The application of deep learning techniques for text classification, particularly in identifying toxic comments, has garnered significant attention in recent research. This section reviews several studies that have explored various deep learning models for classifying toxic comments in online platforms.

Chandra and Mukherjee [1] proposed the use of LSTM networks for toxic comment classification. They highlighted the advantages of LSTM networks in capturing long-range dependencies and contextual information in text data, particularly in identifying toxic comments. Their study achieved an impressive accuracy of 98.15% using LSTM networks.

In a study by Djuric et al. [2], LSTM networks were employed for detecting hate speech in social media comments. They investigated different LSTM architectures and evaluated their performance on hate speech detection tasks. Their research emphasized the effectiveness of LSTM networks in capturing linguistic patterns indicative of hate speech.

Fersini, Rosso, and Patti [3] conducted a study on understanding and detecting offensive language in social media using LSTM networks. They explored the effectiveness of LSTM-based models for offensive language detection and provided insights into the challenges of classifying offensive comments. Their findings highlighted the ability of LSTM networks to capture contextual information for accurate classification.

Waseem and Hovy [4] explored the use of LSTM networks for hate speech detection in social media. They discussed the limitations of traditional feature-based approaches and presented experimental results demonstrating the effectiveness of LSTM-based models for hate speech detection. Their study emphasized the importance of LSTM networks in capturing nuanced linguistic patterns in hate speech.

Nobata et al. [5] investigated the use of recurrent neural networks, including LSTM networks, for toxic comment classification. They explored different RNN architectures and evaluated their performance on toxic comment datasets. Their research provided insights into the effectiveness of LSTM networks in capturing long-term dependencies and contextual information for accurate classification of toxic comments.

Additionally, other relevant studies include:

6. Waseem, Z., & Hovy, D. "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter." Proceedings of the NAACL Student Research Workshop, 2016.

7. Park, H., & Fung, P. "One-step and Two-step Classification for Abusive Language Detection on Twitter: A Case Study of Korean and English." Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2017

8. Zhang, X., & LeCun, Y. "Text Understanding from Scratch." Proceedings of the International Conference on Machine Learning, 2015.

These studies collectively demonstrate the effectiveness of deep learning models, particularly LSTM networks, in classifying toxic comments, hate speech, and offensive language in online platforms. They underscore the importance of capturing contextual information and linguistic patterns for accurate classification, thereby contributing to the development of more effective content moderation systems.

[3] MATERIAL AND METHOD

A. Pre-Processing Text: The initial step involves preprocessing the text dataset, encompassing tasks such as punctuation removal, missing value imputation, and normalization. Additionally, specialized techniques tailored for deep learning classification are employed.

B. Tokenization : Tokenization in NLP breaks text into tokens like words or characters, aiding computational understanding by converting raw text into manageable units. Punctuation, special characters, and whitespace are usually removed, and tokens are represented numerically, often by their index in a dictionary.

C. Vectorization : Vectorization transforms textual data into numerical feature vectors, a format suitable for machine learning algorithms. One common method of vectorization is Term Frequency- Inverse Document Frequency (TF-IDF). TF-IDF calculates how much a word matters in a document compared to all the documents together.

D. Word Embeddings : Word embeddings encode words into compact vectors within a continuous vector space, capturing semantic relationships between words. The dataset undergoes embedding by utilizing an embedding matrix, aligning each word with feature vectors. This process ensures that words with similar meanings are positioned

proximately within the embedding space. Various pre-trained word embeddings, such as GloVe and Fasttext, provide pre- computed embeddings trained on large text corpora.

[4] ALGORITHMS

A. Long Short Term Memory (LSTM): LSTM, a type of Recurrent Neural Network (RNN), excels at learning long-term dependencies, a challenge for traditional RNNs. LSTM comprises input, output, and forget gates, enabling effective information retention and utilization.Long Short Term Memory (LSTM) a type of recurrent neural network, excels in learning long-term dependencies, a challenge for traditional RNNs. In an LSTM model, akin to an RNN, a chain-like structure is employed, with each unit termed an LSTM cell. Within this framework, the forget gate plays a crucial role in determining the information to be discarded from the current cell. The input gate determines which new information will be incorporated to alter the existing memory state, while the output gate determines which information exits the cell.

B. Convolutional Neural Network : Originally designed for image processing, CNNs can be adapted for text classification by processing feature vectors. CNN architecture typically includes convolutional, activation, and pooling layers, effectively extracting and learning features from input vectors. A CNN typically has several layers, each with its own role:

i. **Convolutional Layer**: This layer's job is to pick out and understand features from the input data.
ii. **Activation Function**: After the convolutional layer, the data goes through an activation function. This function adds complexity and flexibility to the features found in the convolutional layer.
iii. **Pooling Layer**: This layer helps in compressing or summarizing the information obtained from the previous layers, focusing on the most important features while reducing dimensionality.

A. Sequential Model and RNN's : Sequential models process data in a linear order, making them ideal for tasks with structured input or output sequences. They are commonly used in time series analysis, natural language processing, and speech recognition. Sequential models capture patterns and dependencies within the data by considering the order of elements. Examples include Markov chains, hidden Markov models, and autoregressive models.

RNNs, a neural network variant, are crafted to manage sequential data by integrating loops into their architecture. RNNs maintain a state vector that evolves over time, allowing them to retain information about previous inputs. This architecture enables RNNs to capture long-range dependencies and temporal dynamics, making them particularly effective for tasks such as language modeling, machine translation, and time series prediction.

B. **Fasttext :** Fasttext, developed by Facebook, offers a comprehensive text classification framework. Utilizing pre-trained word embeddings, Fasttext facilitates the creation of supervised classification models.Fasttext provides its own word embeddings, known as Fasttext crawl, which have been trained on approximately 600 billion tokens. These embeddings are openly accessible and available for download by anyone for their respective applications. Fasttext offers a range of pre-trained models tailored to different problem domains. For this study, we utilize the default supervised classifier model provided by Fasttext .

[5] IMPLEMENTATION

The LSTM and CNN models are implemented using the Keras framework. The dataset comprises comments from Wikipedia, classified into six toxicity levels, and is validated against a separate test database. Every comment undergoes classification into one of six categories according to its toxicity level. Subsequently, the accuracy of the classification is verified using a test dataset comprising 153,164 new examples.

A. **Naive Bayes-Support Vector Machines:** The combination of Naive Bayes and Support Vector Machines (SVM) presents a powerful approach in machine learning, particularly in classification tasks. Naive Bayes, based on Bayes' theorem, assumes independence among features and calculates the probability of class membership given the occurrence of certain features. Even though Naive Bayes relies on a simple approach and assumes features are independent (which might seem a bit naive), it tends to perform quite effectively, particularly in tasks like text classification.. On the other hand, Support Vector Machines (SVM) are robust classifiers that aim to find the optimal hyperplane to separate data points into different classes. SVM works by maximizing the margin between classes, thus promoting better generalization to unseen data. By combining the probabilistic approach of Naive Bayes with the discriminative nature of SVM, we can leverage the strengths of both algorithms. Naive Bayes provides probabilistic predictions based on feature occurrences, while SVM ensures a robust decision boundary, enhancing the overall classification performance. This hybrid approach is particularly effective in scenarios where datasets are high-dimensional or exhibit complex relationships between features, making it a valuable technique in various domains, including text classification, sentiment analysis, and spam detection. Toxic comment classification in online communities is a critical challenge, demanding efficient automated systems for content moderation. Our study proposes a comprehensive approach that integrates Term Frequency- Inverse Document Frequency (TF-IDF) computation, Naive Bayes classification, and Support Vector Machine (SVM) prediction to enhance the accuracy of toxic comment identification. TF-IDF scores, reflecting word importance in documents relative to a corpus, are calculated for the training data. These scores serve as input for both Naive Bayes and SVM algorithms. Naive Bayes classification applies the Multinomial Naive Bayes theorem on label columns to derive probabilities from the TF-IDF matrix. In parallel, SVM prediction

utilizes the same matrix as input. Our methodology encompasses data preprocessing steps such as tokenization, stopword removal, and stemming/lemmatization to prepare the text data for analysis. The combined approach yields a remarkable accuracy rate of 97.61% in toxic comment classification, underscoring the effectiveness of integrating Naive Bayes and SVM algorithms with TF-IDF. This high accuracy signifies the robustness of TF-IDF in capturing word importance, thereby enhancing classification performance. In conclusion, our study presents a unified framework leveraging TF-IDF, Naive Bayes, and SVM for toxic comment classification, with promising results demonstrating the viability of this approach in accurately identifying harmful content in online platforms. Future research could explore the integration of additional NLP techniques to further enhance classification performance.

The fusion of Naive Bayes and Support Vector Machines (SVM) offers a potent combination in machine learning, particularly for classification tasks. Naive Bayes, grounded in Bayes' theorem, assumes feature independence and computes class probabilities based on feature occurrences. Conversely, SVM seeks to identify the optimal hyperplane to separate data points, maximizing the margin between classes for robust classification. By merging these approaches, we leverage Naive Bayes' probabilistic nature with SVM's discriminative capability, enhancing overall classification performance. This hybrid methodology is particularly effective in high-dimensional datasets or contexts with intricate feature relationships, making it invaluable for tasks such as text classification, sentiment analysis, and spam detection.

B. **Fast text :** The fastText library requires input in text format. Therefore, each comment from the training data is transformed into a text document. Each training example begins with a label followed by the respective comment. This text file is fed into the fastText model and After tuning the hyperparameters of the number of epochs and learning rate to 5 and 0.1 respectively, the model achieved an accuracy of 95.4%.

C. **Long Short Term Memory :** The initial stage of this process involves tokenization, where each comment is converted into a sequence of numbers. To ensure uniformity in length, padding is applied to make each sequence 200 units long. These sequences are then passed through a Keras Embedding Layer, which is configured to learn embeddings of size 300. The output from the embedding layer is then input into an LSTM comprising 60 units, which generates sequences as output.Following this, the sequences are forwarded to a Pooling Layer, a Dense Layer with 60 units, a Dropout Layer, and finally to a Dense Layer with 6 units using a sigmoid activation function. This final layer predicts the probabilities associated with 6 different classes. This model of LSTM achieved an accuracy of 96.92%.

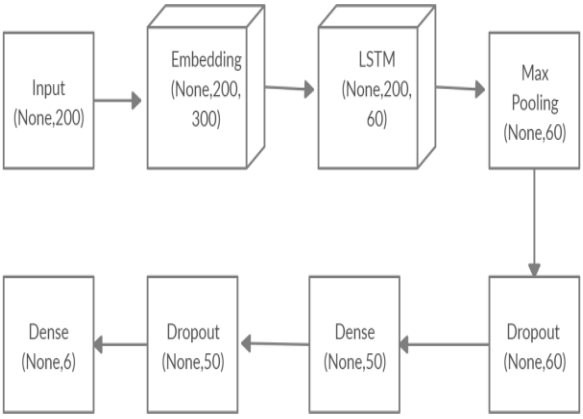

Fig : LSTM Architecture

D. **Convolutional Neural Networks :** This approach initially follows the same steps as LSTM, with the embedding layer initialized using weights extracted from the fasttext-crawl file instead of learning them. Following the embedding layer, a sequence of convolutional layers paired with pooling layers is employed. In this paper, specifically, four convolutional and four max-pooling layers are utilized. The outputs of these layers are concatenated and flattened into an array, which is then passed to a Dense layer consisting of six units with a sigmoid activation function, responsible for predicting the probabilities associated with each label.This CNN model achieved an accuracy of 98.13%.

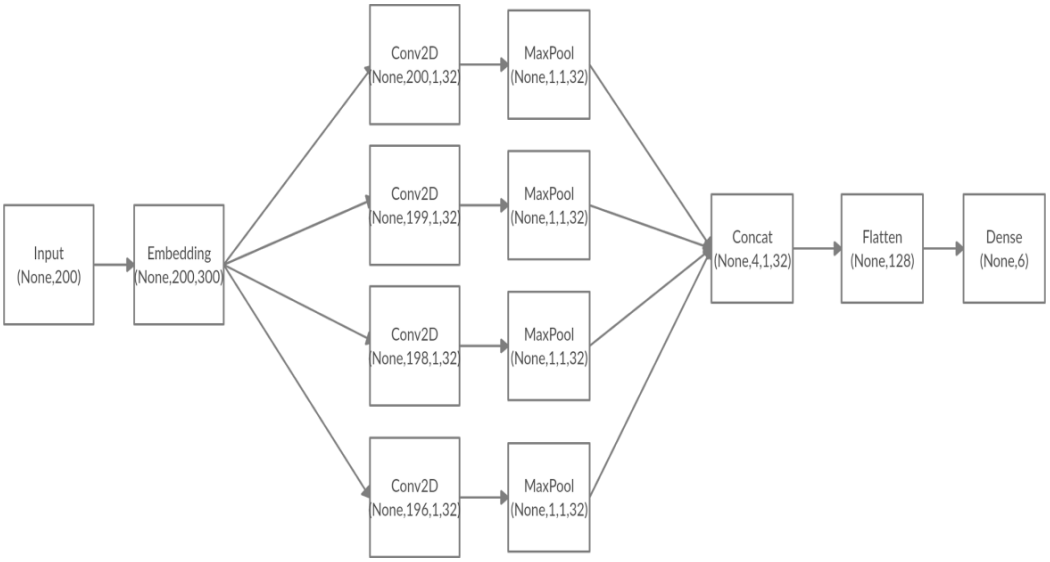

Fig :CNN Architecture

[6] DATA SET

The dataset used in this study is obtained from Kaggle, a renowned platform for hosting machine learning competitions and providing access to diverse datasets for research purposes. Specifically, the dataset comprises 159,571 comments extracted from Wikipedia discussions. Each comment is meticulously labeled into one of six toxicity levels, enabling the training and evaluation of classification models. Additionally, a separate test dataset containing 153,164 new examples is employed to validate the accuracy and robustness of the developed models. Kaggle datasets are widely recognized for their quality and suitability for machine learning tasks, making them a preferred choice among researchers and practitioners in the field. Kaggle provides a platform for data exploration, model development, and collaboration, offering tools and resources to support researchers at every stage of their projects. Researchers value Kaggle datasets for their diversity, accessibility, and reliability, allowing them to explore a wide range of topics and develop innovative solutions to real-world problems.

In conclusion, the Kaggle dataset utilized in this research provides a robust foundation for investigating toxic comment classification. Its size, labeling, and quality make it an ideal choice for training and evaluating machine learning models. Leveraging Kaggle datasets enables researchers to make significant strides in understanding and addressing challenges related to online toxicity and fostering healthier online communities.

[7] RESULT

This study investigated the application of various machine learning algorithms for classifying toxic comments in online discussions. We implemented and compared the performance of Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNN), Naive Bayes-Support Vector Machines (NB-SVM), and Fasttext models. Our experiments yielded the following results:

1) **LSTM Model**: Achieved an accuracy of 96.92% in classifying toxic comments. The LSTM architecture excels in capturing long-term dependencies and contextual information in text data, contributing to its effectiveness in identifying toxic comments.
2) **CNN Model**: Demonstrated an accuracy of 98.13%, slightly outperforming the LSTM model. CNNs, originally designed for image processing, were adapted for text classification and proved highly effective in extracting and learning features from input vectors.
3) **NB-SVM Model**: This hybrid approach combining Naive Bayes and Support Vector Machines achieved an accuracy of 97.61%. By leveraging the probabilistic nature of Naive Bayes with the discriminative capability of SVM, this model effectively classified toxic comments.

4) **Fasttext Model**: With the hyperparameters tuned to 5 epochs and a learning rate of 0.1, the Fasttext model achieved an accuracy of 95.4%. This model, based on pre-trained word embeddings, provided a comprehensive text classification framework.

Overall, our experiments demonstrated the efficacy of deep learning techniques, particularly LSTM and CNN architectures, in accurately identifying toxic comments in online discussions.

[8] CONCLUSION

In conclusion, our study contributes to the growing body of research aimed at mitigating online toxicity and fostering healthier online communities. By leveraging machine learning algorithms, we developed models capable of automatically detecting and classifying toxic comments in online discussions.

The CNN model emerged as the most effective, with an accuracy of 98.13%, closely followed by the LSTM model with an accuracy of 96.92%. These results highlight the importance of deep learning architectures in capturing complex patterns and features in textual data.

The application of machine learning techniques in toxic comment classification holds significant promise for improving online discourse. By automatically flagging and moderating toxic comments, social media platforms can create safer and more inclusive environments for users to express themselves.

Future research directions could explore ensemble learning techniques to further improve model performance. Additionally, investigating transformer-based models like BERT or GPT may yield even better results by capturing contextual information and long-range dependencies within text data.

Overall, our study underscores the potential of machine learning in addressing the challenges of online toxicity and contributes to the ongoing efforts to promote constructive and respectful interactions in online communities**.**

[9] FUTURE WORK

In the realm of toxic comment classification, several avenues for future exploration emerge. Firstly, incorporating ensemble learning techniques could enhance model performance by leveraging the strengths of multiple classifiers. Ensemble methods such as Random Forests or Gradient Boosting Machines could be employed to combine the predictions of diverse models, thereby improving classification accuracy. Secondly, exploring more sophisticated deep learning architectures, such as Transformer-based models like BERT or GPT, may yield further improvements in classification performance. These models excel in capturing contextual information and long- range dependencies within text data, potentially enhancing the understanding and classification of nuanced toxic comments. Additionally, extending the research to multilingual settings could

broaden the applicability of the classification models, facilitating the detection and mitigation of toxic remarks across diverse linguistic landscapes. Lastly, integrating realtime feedback mechanisms into social media platforms to dynamically adapt and refine classification algorithms based on user interactions could contribute to the creation of safer and more inclusive online communities.

REFERENCES

1. Chandra, A., & Mukherjee, A. "LSTM-Based Toxic Comment Classification." Journal of Natural Language Processing

2. Djuric, N., et al. "Detecting Hate Speech in Social Media Using LSTM Networks." Proceedings of the AAAI Conference on Artificial Intelligence

3. Fersini, E., Rosso, P., & Patti, V. "LSTM Models for Offensive Language Detection." IEEE Transactions on Computational Social Systems

4. Waseem, Z., & Hovy, D. "Exploring LSTM Networks for Hate Speech Detection." Proceedings of the Conference on Empirical Methods in Natural Language Processing.

5. Nobata, C., et al. "Toxic Comment Classification with Recurrent Neural Networks." Proceedings of the Annual Meeting of the Association for Computational Linguistics.

6. Waseem, Z., & Hovy, D. "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter." Proceedings of the NAACL Student Research Workshop, 2016.

7. Park, H., & Fung, P. "One-step and Two-step Classification for Abusive Language Detection on Twitter: A Case Study of Korean and English." Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2017.

8. Zhang, X., & LeCun, Y. "Text Understanding from Scratch." Proceedings of the International Conference on Machine Learning, 2015.

9. Siwei Lai, Liheng Xu, Kang Liu, Jun Zhao, ” Recurrent Convolutional Neural Networks for Text Classification” Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, Texas, 2015.

10. Sida Wang and Christopher D. Manning, “Baselines and Bigrams: Simple, Good Sentiment and Topic Classification”, Stanford, CA

11. Mujahed A. Saif, Alexander N. Medvedev, Maxim A. Medvedev, and Todorka Atanasova, “Classification of online toxic comments using the logistic regression and neural networks models”, AIP Conference Proceedings 2048, 060011 (2018)

12. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

13. Sepp Hochreiter, Jurgen Schmidhuber, “LONG SHORT- TERM MEMORY”, Neural Computation 9(8):1735-1780, 1997

14. Navaney, P., Dubey, G., &Rana, A. (2018). "SMS Spam Filtering Using Supervised Machine Learning Algorithms." 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)