International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

Multilabel Toxic Comment Detection and Classification

[1st] Sharayu Lokhande
Department of Computer Engineering
Army Institute of Technology
Pune, India

[2nd] Ajay Kumar
Department of Computer Engineering
Army Institute of Technology
Pune, India

[3rd] Kumar Shivam
Department of Computer Engineering
Army Institute of Technology
Pune, India

[4th] Naresh Kumar
Department of Computer Engineering
line 3: Army Institute of Technology
Pune, India

[5th] Rahul Malhan
Department of Computer Engineering
Army Institute of Technology
Pune, India

***Abstract*— Toxic comments refers to hatred online comments classified as disrespectful or abusive towards individual or community. With a boom of internet, lot of users are brought to online social discussion platforms. These platforms are created to exchange ideas, learning new things and have meaningful conversations. But due to toxic comments many users are not able to put their points in online discussions. This degrades quality of discussion. In this paper we will check the toxicity of comment. And if the comment is toxic then classify the comments into different categories to examine the type of toxicity. We will utilize different machine learning and deep learning algorithms on our dataset and select the best algorithms based on our evaluation methodology. Moving forward we seek to attain high performance through our machine learning and deep learning models which will help in limiting the toxicity present on various discussion sites.**

Keywords*— *Toxic Comments, Natural Language Processing, Machine Learning, Deep Learning, Text Classification, Multilabel Classification

I. INTRODUCTION

There is increase in number of people using internet. Internet is main invention for 21st century. According to website , the number of internet users have increased from 1100 million in 2005 to 3969 million users in 2019 which is staggering 260% increase [1]. Hence, more people are using social networking and online discussion platforms.

There is also a huge shift the way internet is used. In the initial days of the internet, people used to communicate with each other through Email. But with a platform like social media, we see that people got a way to keep in touch with their long-lasting friends, meet new peoples having same interests and hobbies. We are now more connected than ever. Not only discussion of friends and people, but social media has also evolved to support business needs. With increase in services like gaming and live streaming, more velocity of comments is added to sites. Social media has broken down many of the communication barriers between different consumer groups as well as between individuals. Hence no doubt that social media sites such as Facebook, Twitter, Reddit, etc. have become billion-dollar companies.

Over these all years we have seen lot of instances where social media have played pivot role due to toxic comments and hatred. For example, Chief Minister of Uttar Pradesh State of India blamed social media like Facebook, Twitter, and YouTube for escalating tensions during communal conflict between Hindu and Muslim community in Muzzafarnagar, India in 2013 [2]. Kalamboli police on booked a man for abusing and threatening the police via a comment on a Facebook post [3]. Another example is of Riots that took place in DJ Halli, Bengaluru, India in 2020 over a provocative Facebook post against Islam that left 3 dead and many injured [4].

On January 6, 2021 US Capitol Riots took place by supporter of Donald Trump. Many extremists had posted on Social Networking sites posts such as “occupy the Capitol”, “bring revolution”, etc. before riots [5]. Hence, it is very important to detect such threats, hatred, toxicity on online discussion platforms and social networking sites. Because not doing so can cause violence, riots, prevent good debates, make internet an unsafe place and can affect people mentally.

Let us take an example of comment present in our dataset “Just shut up and stay shut. Don't edit anymore”, it can be easily identified that the phrases like “shut up”, “Don’t edit anymore”, etc. are negative and thus this comment is toxic. But it besides toxic we need to go through series of steps to classify comment using machine learning classification algorithms to verify type of toxicity of obtained results.

We will use different machine learning and deep learning models on our Data set which is made available by Conventional AI in Kaggle.com. In this paper we will use Logistic Regression and Support Vector Machine Models with TF-IDF Vectorizer, Long Short-Term Memory with Glove and Word2Vec Embedding. We have used all models on given dataset and compare their scores to find which one will be best.

The rest of the paper is arranged as follows. All the recent approaches being used for text classification and Natural Language Processing have been elaborated in

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

Section II. Section III contains the proposed methodology, different machine learning models and evaluation metric that is used. Section IV contains results and analysis. The paper ends with conclusion in Section V.

II. RELATED WORK

There is lot of information being delivered every time on social media sites. There is increase in hate speech that both may promotes violence towards Muslims and Arabs after following extremist violence events [6]. Because of this negativity, particular community might feel insecurity while utilizing social platforms. There was survey conducted asking American adult about problem of online harassment or bullying. Roughly four-in-ten Americans have personally experienced online harassment. 62% percent of participant in study considered it as major problem whereas 33% considered as minor problem. A total of 95% called it problem and 35% agreed for online companies to build better policies and tools for their platforms [7]. Due to negativity, civilized conversations via social media are not present since hateful remarks are limiting individual to communicate and to have contradicting feelings [8].

There were endeavours by people to build the online wellbeing by moderating websites through crowd-sourcing schemes and remark criticizing, much of the time these procedures neglect to recognize the toxicity [9]. Along these lines, we need to track down a potential method that can recognize the online toxicity of client content successfully [10].

As Computer understands binary information and in real world we have information in different structures for example pictures or text. So, we need to change over the information of real world into binary for legitimate processing through the computer. In this paper, they have utilized this changed over information and apply Machine learning strategies to arrange online remarks [11].

Nguyen and Nguyen [12] made model consisting of 2 components – Deep Learning Classifier and Tweet Processor. Tweet Processor is used for applying semantic rules and preprocessing on datasets to capture important information. They used character-level embeddings to increase information for word-level embedding. They then used DeepCNN for character level embeddings. After that a Bidirectional Long Short-Term Memory network (BiLSTM) produces a sentence-wide feature representation from the global fixed size feature and word-level embedding. Their model produce an accuracy of 86.63% on Standford Twitter Sentiment Corpus.

Liang-Chih Yu et al. [13] proposes a word vector refinement model. This refinement can be applied to any of already trained word vectors. Based on semicolon lexicon, their model gives high rank to sentimentally similar neighbour and vice versa. Their experimental results show that their proposed method can perform better for various neural networks. It also have better performance for both sentimental embedding and conventional word embeddings.

A method was introduced by Wulczyn et al. [14] develop method to analyze personal attacks. They generated over 63M machine labeled and 100k human labeled comments. They found that attacks on Wikipedia are not limited to a set of users.

Hossein Hosseini et al. [15] apply the attack on the Perspective toxic comment detection website. This website gives toxic score to any phrase. They tried to modify a toxic phrase having same meaning so that model will give it very low toxic scores. This existence is harmful for toxic detection system.

In another methodology, Convolutional Neural Networks (CNN) was utilized in text characterization over online substance [16], with no information on syntactic or semantic language.

Y. Chen et al. [17] propose the Lexical Syntactic Feature (LSF) architecture to distinguish offensive content and recognize likely offensive clients in web-based media. Their experiments shows that LSF algorithms for sentence and user offensiveness outperformed traditional learning-based methods. Their LSF can adapt to various writing styles of English language and can tolerate informal and misspelled content.

Jigsaw and Google's Counter Abuse Technology team introduce project named Perspective. It uses machine learning models to identify abusive comments. The models score a phrase based on the perceived impact the text may have in a conversation and have capability to classifying comments.

Navoneel Chakrabarty [18] utilizes machine learning model on Jigsaw Toxic Comment Detection dataset to label toxicity of comments and produce the Mean Validation Accuracy, so obtained, is 98.08%.

III. PROPOSED METHODOLOGY

In this paper, we take a dataset provide in Kaggle website provided by Conventional AI. It is collection of large number of comments in Wikipedia website and labeled as - toxic, severe toxic, threat, identity hate, obscene and insult. The advantage of this type of data is that these comments represent a true sample of the content present on the social media sites. We first ran analysis and visualization on this data which we have discussed in Section III-B. For our Machine Learning model, we have removed outliers and noise which is present is data. We initially tested the performance of classical models namely, Support Vector Machine and Logistic Regression on this task with TF-IDF Vectorizer. We then applied pretrained embeddings, namely GloVe and Word2Vec in our model and performed the classification. We compared the performance of all the models using the mean AUC ROC score, Hamming and Log loss as the performance metric.

A. Type of Classification

As discussed above our dataset have 6 categories i.e., threat, insult, toxic, severe toxic, obscene, or identity hate. Hence our problem can belong to multiclass or multlabel classification problem.

As we can see the above description this problem is a multiclass classification as well as multilabel classification problem.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

- **Multiclass Classification:** In this classification we put each example into one of the several possible categories unlike binary classification problem where each example belongs to only one of two possible categories.
- **Multilabel Classification:** This type of classification involves examples such that each example can belong to multiple categories and not necessarily only one category. A multi label classification problem can be viewed as a generalized version of the multiclass classification problem in which there is no restriction over how many classes a training example can belong to.

As in our dataset, our data can belong to zero to any number of categories, hence our problem is Multilabel Classification problem.

B. Data Visualization

Data Visualization helps us get visual insights about the dataset, helping us to find patterns and features of data for selecting right machine learning algorithms and evaluation metrices.

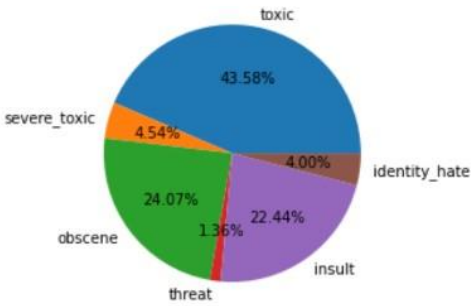

Figure 1 Hate tags present in dataset (Pie Chart)

We plotted a pie chart in Figure 1, showing the relative amount of hate tags present in the dataset and found a massive class imbalance. “Toxic” tags were the highest, whereas tags labeled with “threat” had the lowest count.

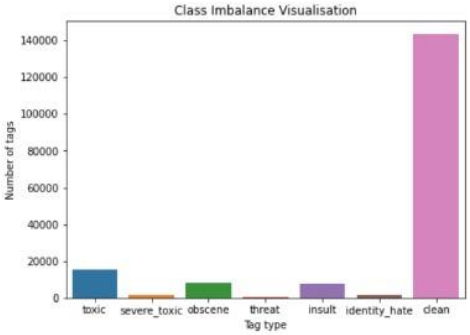

Figure 2 Hate tags present in dataset (Bar graph)

Also, plotting a bar graph Figure 2 after adding a column for “clean” tags, we found that most of comments are clean.

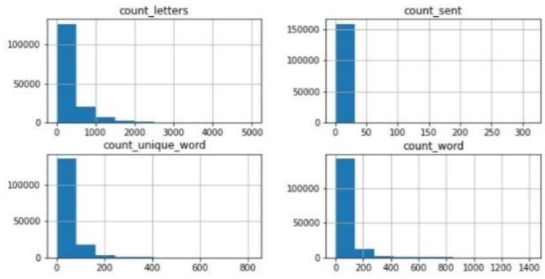

Figure 3(a) Number of Comments vs Count of letters, (b) Number of Comments vs Count of sentence, (c) Number of Comments vs Count of unique words, (d) Number of Comments vs Count of words

Figure **3** shows that lot of comments have less count of words, sentences, or letters.

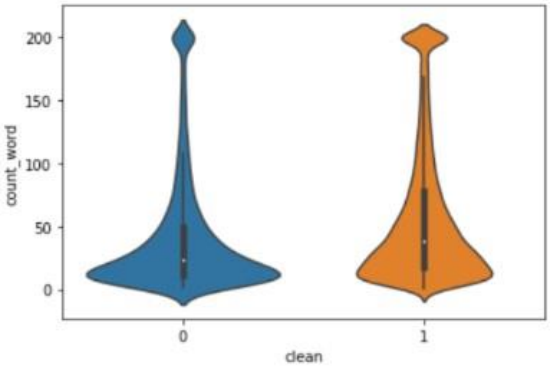

Figure 4 Count of word vs Clean Comment (0-toxic, 1-clean) Violin Plot

Figure 4 shows that there is slight correlation between count of word and comment’s toxicity.

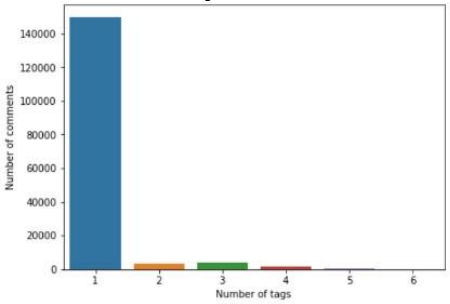

Figure 5 Bar Chart showing Number of comments vs Number of tags (type of toxic comment tags)

From Figure 5, it is quite evident that most of toxic comment have only one label, but there are some tags with two, three and four labels hence problem is multilabel classification problem.

C. Data Preprocessing

We have used both Machine learning and Deep learning models over our dataset. We have used hence used TF-IDF Vectorizer with machine learning models – Logistic Regression and SVM and GloVe and Word2Vec Embedding with LSTM. Raw text cannot directly be input to any Machine learning algorithms. Some of most popular technique for converting text into numerical features are Bag of Words, TF-IDF Word2Vec and GloVe.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

1. **Term Frequency Inverse Document Frequency (TF-IDF) Vectorizer**
   This vectorizer normalize the text. It reduces the weight of tokens that are occurring multiple times in documents. It is made in such a way that its value increases if the frequency of term is more in a document and value decreases if term occurs in multiple documents. It consists of 2 parameters – Term Frequency (TF) and inverse document frequency (IDF).
   - Term Frequency–$TF(i,d)$: This calculates the frequency of a term $'i'$ in a document 'd'. This is like the Count Vectorizer method of encoding text.
   - Inverse Document Frequency-$IDF(t)$: IDF gives the inverse of document frequency. df_t is count of documents that contains term t. Hence, it calculates number of documents in which term appears and take inverse and log that. Later 1 is denominator added to avoid zero-division.
     $$idf(t) = \log\left(\frac{1+|D|}{1+\ df_t}\right) + 1$$
     where, df_t denotes number of documents containing term t and $|D|$ contains total number of documents.
2. **Glove Word Embedding**
   Word embeddings are used to represent words in structured format. Because most of the data in internet is not structured, word embeddings techniques are a useful tool to transform data into more structured format so that useful information can be extracted.
   In Bag of words models feature extraction can be done but they fail to capture any semantic or contextual information in texts. One-hot encoded vectors may lead to a highly sparse structure which causes the model to overfit. To overcome these shortcomings of the above approach, word embeddings are used.
   Word embeddings represent words in the form of vectors in pre-defined dense vector space. These vectors contain meaningful semantic information about the words. The idea in this approach is words with similar semantic information are like each other. Hence, the similar words will be in proximity within the high dimensional vector space. Hence, we can significantly reduce the vector size in contrast to the one-encoding technique.
   Pretrained word embeddings are obtained by unsupervised training of a model on a large corpus. As they are trained on a large corpus, they capture the semantic information of most of the words. These pretrained embeddings are provided by different companies and organizations for open use.
   GloVe stands for Global Vectors. It is provided by Stanford as an open-source project. In this approach, a word co-occurrence matrix is constructed. This helps in capturing the semantic information. The co-occurrence matrix stores information about the frequency that appear in some context. Thus, it takes both local statistics and global statistics into account to obtain the embeddings.
3. **Word2Vec**
   It is one of the earliest pretrained embedding. It has 2 flavors. First is Skip-Gram Model where the algorithm tries to predict the context or surrounding words in which the word would have been used. It learns by predicting the surrounding words given a current word. Second is Continuous Bag of Words (CBOW) model where the algorithm tries to predict the word if a context is given. In this way, the word embeddings vectors are generated.

D. Machine Learning Models

1. **Logistic Regression**
   Logistic Regression is the fitting regression analysis to use when the reliant variable has a binary answer. Like any remaining kinds of regression frameworks, Logistic Regression is likewise a predictive regression framework. Logistic Regression is utilized to assess the connection between one reliant variable and one or more non-reliant variable. It gives discrete yields going somewhere in the range of 0 and 1. Logistic Regression utilizes a more complex cost function; this cost function is known as the 'Sigmoid function' or otherwise called the 'logistic function'.
   $$f(x) = \frac{1}{1+\ e^{-x}}$$
   In our case, we have used logistic regression for prediction in each class i.e., toxic, severe toxic, threat, identity hate, obscene and insult and mean score.

2. **Support Vector Machine**
   Support Vector Machines use the concept of support vectors and hyperplanes for classification tasks. They can be used for both regression and classification tasks but are more popular for classification. They can be used for both linear and non-linear data. It works by constructing an optimal hyperplane in an N-dimensional space i.e., a boundary separating the data points, such that the margins between the support vectors is maximized. Support vectors refers to the data points which are closest to the hyperplane and are useful for training tasks, and the margin refers to the distance between the two parallel lines passing through the closest support vectors on either side of the hyperplane.

   In case of non-linearly separable data, it transforms the original data into higher dimensions for classification task. It is also known that SVM perform excellent for higher dimensional data because complexity of SVM does not depend on dimensionality of data used but on number of support vectors. This also helps it to be memory efficient.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

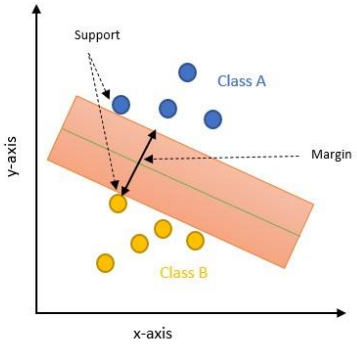

Figure 6 Support Vector Machine

In our case, we have used Support Vector Machine with Binary Relevance and Classifier Chains for predictions.

In Binary Relevance method, we transform the problem into separate single-class classification problems, each of the problems having a single label. We then apply the Support Vector Machine on each problem separately to get the result. After that, the results of each problems can be combined to get all the labels for a comment. Simple it is it comes with drawbacks. It ignores any correlation between the labels. Hence it will give poor results if there is correlation between labels.

In Classifier Chain method, we transform the problem into separate single-label classification problems, such that if ith classifier is trained on input variable(s) X , then $(i+1)th$ classifier is trained on input variable X and output produced by ith classifier. Hence, this technique considers the correlation between the labels, since for every new classifier, the predictions of the previous classifiers are considered, i.e., for a given target variable, it also considers the correlation between previous target variables.

3. Long Short Term Memory

Artificial neural network is a layered design of connected neurons, enlivened by natural neural network. It is not one algorithm yet mixes of different algorithm which permits us to do complex procedure on data.

Recurrent Neural Networks (RNN) is a class of neural networks customized to manage worldly information. The neurons of RNN have a cell state/memory, and input is handled by this interior state, which is accomplished with the assistance of loops within the neural networks. There is repeating module of '$tanh$' layers in RNNs that permit them to hold data but not for too long. This is the reason we need LSTM models. LSTM are special recurrent neural network that can capture long term dependencies. The cell state is regulated using gates which determine the amount of information that will flow through them. It has cell state c_t along with hidden states which stores information. This information can travel through cell state without any change hence, preserving long term dependencies.

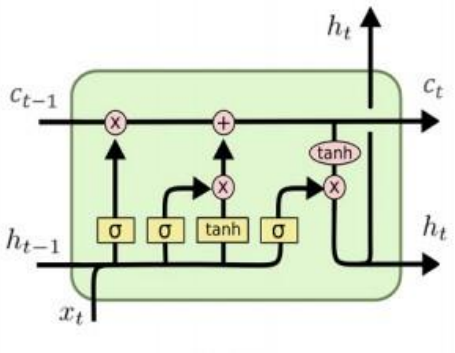

Figure 7 LSTM [19]

LSTM unit takes current input, previous hidden state, previous cell state as the input and results in the new cell state and hidden state. A LSTM unit consists of Input Gate, Forget Gate, Output Gate, Candidate Layer, Output Layer. All the gate uses Sigmoid function as activation function.

Forget Gate Layer decides the information to be stored in cell state using h_{t-1}, x_t and sigmoid layer. The decision of new information to be stored is done by input gate layer and $tanh$ layer. $tanh$ creates a vector of candidates $\widetilde{C_t}$ that can be new information. This happens after input gate decides on which values to update. Output is decided on basis of cell state. Sigmoid layer decides which part of cell state to output. $tanh$ changes value of cell state in -1 and 1 and multiply it by the output of the sigmoid gate.

RNN and LSTM give output based on current data and past data that have already passed through it. One directional LSTM does not take in account data further in sequence while predicting. Bi-directional LSTM trains two independent LSTMs in opposite directions and connect the both the hidden layers to the same output. One LSTM trains in forward direction and other in backward direction. Using the two hidden states combined we can use information from both past and future. In prediction of next word problem, Unidirectional LSTM can only see "The girl went to …" but in Bidirectional LSTM, forward LSTM sees "The girl went to …" and a backward LSTM sees "… and then there was sandstorm". This information provided by Backward LSTM can help to understand what next word is.

For our case, we have used Word2Vec and Glove word embedding available in Kaggle with 300 dimensions and then train a Bidirectional LSTM with 4 epochs.

E. Evaluation Metric

There are quite a lot of evaluation metric for machine learning models. The problem involves highly unbalanced dataset. So, accuracy is not a well-suited performance measure. With only 10% of the training data belonging to the positive class (hate tags), it is trivial to achieve 90% accuracy

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

by a naive model which simply labels every input as clean. Precision-Recall or F1 score seem like the next obvious choice however they have their own share of limitations including selection of threshold value and relative importance to be given to precision vs recall. Hence, we finally settled on the ROC curve and AUC score which give a very accurate picture of the performance of a discriminative model, Hamming loss and Log loss.

Receiver Operating Characteristic is curve that plot True Positive Rate (TPR) vs False Positive Rate (FPR). Aim of any model is to have high True Positive Rate while keeping False Positive Rate as low as possible.

$$TPR = \frac{TP}{TP + FN}$$

where, TP (True Positive) is number of samples that are true and predicted as true and FN (False Negative) is number of samples that are false and predicted as true.

$$FPR = \frac{FP}{FP + TN}$$

where, FP (False Positive) is number of samples that are false and predicted as false and TN (True Negative) is number of samples that are true and predicted as false.

AUC denotes the complete Area Under the ROC curve for the given domain. Its value can range from 0 to 1.A better model will have more area under the ROC curve with a perfect model having AUC=1 and a model which always predicts incorrectly having AUC score=0. In this sense AUC can be understood as the average of performance measures of the classifier across all thresholds. AUC is scale and threshold invariant.

Hamming Loss is fraction number of labels that are incorrectly predicted to total number of labels.

$$Hamming\ Loss = \frac{1}{NL} \sum_{l=1}^{L} \sum_{i=1}^{N} Y_{i,l} \oplus X_{i,l}$$

where, $\oplus$ is exclusive-or, $Y_{i,l}$ is predicted value and $X_{i,l}$ is the actual value for the ith comment on lth label value, NL is the total number of labels.

Log Loss takes in account probability of models. It is defined as following:

$$Log\ Loss = \frac{1}{N} \sum_{l=1}^{N} \sum_{i=1}^{M} Y_{li} \log (p_{li})$$

where M is the number of labels, N is the number of samples, Y_{li} is a binary indicator of the correct classification and is model probability.

IV. RESULT AND ANALYSIS

After applying 3 different machine learning models to our dataset, we got the result in form of ROC AUC, Accuracy, Hamming Loss and Log Loss.

TABLE I. HAMMING LOSS , LOG LOSS AND ROC AUC SCORES FOR MACHINE LEARNING MODELS

Model	ROC AUC Score	Hamming Loss	Log Loss
Logistic Regression	0.74628214 81181874	0.02929652 901518230	1.01187927 52266792
SVM Binary Relevance	0.68860740 82694079	0.02672533 266643742	1.62367158 70057155
SVM Classifier Chains	0.70052230 66528395	0.02836912 688736753	1.51399782 69884409
LSTM (Word2Vec Embedding)	0.96646487 43249146	0.02778819 802640491	0.28978367 47210841
LSTM (Glove Embedding)	0.96666456 30149224	0.02574582 929548699	0.29125949 122474454

After analysis results, we can say that LSTM with Glove embedding performs the best because it has highest ROC AUC score and lowest Hamming loss and one of the lowest Log loss among all models which means that there is very less multilabel are accurately measured. LSTM with Word2Vec embedding has also performed comparable to LSTM with glove embedding. We also observe that Classifier SVM performs better than Binary Relevance SVM which was expected. Both hamming loss and log loss in all our models are lower than the algorithms presented in [20]. It was expected for deep learning model LSTM to have best result over all the algorithms.

V. CONCLUSION

We compared the performance of the model based on the mean ROC AUC scores, hamming and log loss. Hamming and Log loss of classical models are more than deep learning LSTM model. We also found out that the classifier chain method performed slightly better than binary relevance in this task. LSTM model outperformed other models in this task. We can further test the performance of state of the art models like Transformers on this task. BERT (Bidirectional Encoder Representations from Transformers) caused stir in Machine Learning community by presenting state-of-the-art results in wide NLP tasks. BERT can be used for this problem in future. We can also experiment with more sophisticated models like GRUs. We can also combine machine learning models together for this problem. We can ensemble the results obtained from the various models in a majority vote fashion.

VI. REFERENCES

[1] J. Johnson, "statista," 27 Jan 2021. [Online]. Available: https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/. [Accessed 15 Mar 2021].

[2] B. S., "The Role of Social Media in Mobilizing People for Riots and Revolutions," in *Social Media in Politics, Public Administration and Information Technology, vol 13. https://doi.org/10.1007/978-3-319-04666-2_19,* Springer, Cham , 2014.

[3] R. Assainar, "The Hindu," 04 May 2020. [Online]. Available: https://www.thehindu.com/news/cities/mumbai/kalamboli-man-abuses-police-on-fb-booked/article31496732.ece.

[4] A. Bharadwaj, "The Hindu," 12 August 2020. [Online]. Available: https://www.thehindu.com/news/national/at-least-three-killed-in-police-firing-as-riots-break-out-over-fb-post-in-bengaluru/article32331790.ece.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 05, May-2021

[5] K. Dilanian, "NBC News," 8 March 2021. [Online]. Available: https://www.nbcnews.com/politics/justice-department/fbi-official-told-congress-bureau-can-t-monitor-americans-social-n1259769.

[6] C. C. J. B. K. R. V. Alexandra Olteanu, "The Effect of Extremist Violence on Hateful Speech Online," in *Twelfth International AAAI Conference* , 2018.

[7] M. Duggan, "Pew Research," July 2017. [Online]. Available: https://www.pewresearch.org/internet/wp-content/uploads/sites/9/2017/07/PI_2017.07.11_Online-Harassment_FINAL.pdf.

[8] J. F. T. P. A. R. A. J. K. Marilyn Walker, "A Corpus for Research on Deliberation and Debate," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, 2012.

[9] P. S. H. T. S. R. P. S. S. K. M. P. G. A. M. Binny Mathew, "Thou shalt not hate: Countering Online Hate Speech," in *ICWSM 2019*, 2019.

[10] J. T. A. T. Y. M. Y. C. Chikashi Nobata, "Abusive Language Detection in Online User Content," in *WWW '16: Proceedings of the 25th International Conference on World Wide Web*, 2016.

[11] S. K. V. T. M. IKONOMAKIS, "Text Classification Using Machine Learning Techniques," *WSEAS TRANSACTIONS on COMPUTERS*, vol. 4, no. 8, 2005.

[12] M.-L. N. Huy Nguyen, "A Deep Neural Architecture for Sentence-level Sentiment Classification in Twitter Social Networking," in *PACLING Conference* , 2017.

[13] J. W. K. R. L. X. Z. Liang-Chih Yu, "Refining Word Embeddings for Sentiment Analysis," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[14] N. T. L. D. Ellery Wulczyn, "Ex Machina: Personal Attacks Seen at Scale," in *WWW '17: Proceedings of the 26th International Conference on World Wide Web*, 2017.

[15] S. K. B. Z. R. P. Hossein Hosseini, "Deceiving Google's Perspective API Built for Detecting Toxic Comments," in *Computers and Society (cs.CY); Social and Information Networks (cs.SI)*, 2017.

[16] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[17] Y. Chen, Y. Zhou, S. Zhu and H. Xu, "Detecting Offensive Language in Social Media to Protect Adolescent Online Safety," in *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, Amsterdam, Netherlands, 2012.

[18] N. Chakrabarty, "A Machine Learning Approach to Comment," in *INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE IN PATTERN RECOGNITION (CIPR 2019)*, 2019.

[19] colah, "Colah Blog," 27 August 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed 18 03 2021].

[20] H. K. J. H. G. S. Rahul, "Classification of Online Toxic Comments Using Machine Learning Algorithms," in *Proceedings of the International Conference on Intelligent Computing and Control Systems* , 2020.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)