2022 Some Conference
Some date, Chittagong, Bangladesh

Toxicity Detection and Classification From Social Media Comments

Mohammad Minhaj Reza Hasan[*], Md. Nesarul Haque[†], Puja Chakraborty[*], Md. Hanif Seddiqui[*]
[*]Department of Computer Science and Engineering, University of Chittagong, Bangladesh
[†]Department of Computer Science and Engineering, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Banglad
: minhazreza8@gmail.com, mnhshisir@gmail.com, puja.cse@std.cu.ac.bd, hanif@cu.ac.bd

***Abstract*—As e-commerce and social media platforms grow in popularity, the number of microblog websites and user-generated material floods the internet. The proliferation of toxic comments does as well. As a result, detecting and classifying toxic comments has become a popular research topic. It's also a source of concern for communities that have been affected by toxic comments. Researchers and businesses are working to create effective approaches for dealing with toxic comments. Nonetheless, these efforts need continuous improvements. This research is largely concerned with two major objectives. Detecting toxicity and classifying toxicity. To achieve the goal, deep neural networks such as LSTM(Long Short-Term Memory), GRU(Gated Recurrent Unit), and CNN implementations have been examined, in addition to basic models like SVM and Multinomial NB. The thesis' purpose is to not only improve individual model's performance, but also to compare them in terms of natural language processing capabilities. The experimental results show that all four preliminary models performed nearly equally well in binary classification, whereas CNN performed better in multi-label classification with an accuracy of 87 percent.**

***Index Terms*—LSTM,GRU,Multi-label classification**

I. Introduction

Advances in IT technology and widespread virtualization have resulted in huge social media engagement, and there is little doubt that the internet is one of the most important hallmarks of the twenty-first century. Since 2004, according to (Zaheri, Leath, and Stroud 2020), social media has grown at an exponential rate. Meanwhile, social media is a platform to express personal viewpoints and exchange ideas in the context of making a positive contribution to the development of a safe environment in which everyone can exercise their rights as they see fit (Yang et al. 2019). "Twitter users generate 500 million tweets per day, and in 2019, they had a 14 percent year-over-year growth of daily usage," according to a report by (Birkland 2019).

But these huge number of comments,texts may conceal dangers such as sexual harassment, bogus news, and toxicity (Duggan 2014). Few remarks result in not only verbal toxicity, but also numerous personal attacks on reputable organizations and celebrities, bullying, and attacks on people's self-respect. It may have an impact on the individual's feelings, and some people may commit suicide as a result of these statements. These internet remarks may have adverse impact on one's job security, organization revenue, and so on. According to Wikimedia, 54 percent of people who have been bullied have acknowledged a reduction in their participation in social projects (Wulczyn, Thain, and Dixon 2017). Threatening, obscene, insulting, or identity-based hatred are all examples of toxic comments. For fear of being abused or harassed, some people refrain from expressing their opinions or seeking out different points of view, resulting in an unhealthy and unfair conversation. As a result, many platforms are regularly obliged to either limit user comments or shut down comment threads entirely in order to ensure fair discussion.

For that, automatic toxic comment detection and prediction in real time is critical in this regard, as it would allow for the prevention of a number of negative consequences for internet users.And the level of perfection required in these systems is quite high. As a result, the current research was motivated by the need for improved procedures and approaches to better identify and classify distinct sorts of online comments.

Many state-of-the-art methods have been proposed in recent years to build systems that help to identify and classify toxic comments. But the level of perfection required in these systems is quite high. Deep learning models were used alongside machine learning models in these systems.

(Saif et al. 2018), discuss the issue of identifying abusive content on the Internet. A solution to the challenge of toxic online comment classification, which was posted on the machine learning site Kaggle (www.Kaggle.com) in March of 2018, is presented. Four models for performing the job are proposed based on the study of the initial data: a logistic regression model and three neural network models - CNN, LSTM, and Conv + LSTM.
All models are found to solve the task successfully, however the combined model Conv + LSTM is the most effective, since it delivers the highest accuracy. The best model in this study has 2 LSTM layers and 4 Conv Layers, with a 0.9645 accuracy score.

(Abubakar, Tukur, and Usman, n.d.) discusses Long-Range Correlations as a Difficulty to Toxic Comment Classification. This is particularly problematic in the case of lengthier comments. The low accuracy of comment classification systems is another issue. They used LSTM because they deal with a lot of long comments. Epoch has been shown to improve the accuracy of long-term memory. Epoch has a beneficial impact on the speed and quality of the learning process, therefore it tends to improve the classifier's accuracy. They improved precision by 0.4068, recall by 0.2871, F1 by 0.2293, and inaccuracy by 0.4291, which is not a significant improvement in terms of accuracy and other scores.

(Li, Mao, and Liu 2019) implemented three models in the project that could detect toxic comments with high accuracy. Their baseline model was Naive Bayes, SVM, which scored 68.33 percent F1 and 87.57 percent EM. After that, two deep-learning models, LSTM and BERT, are thoroughly examined. They used weighted loss to overcome the problem of unbalanced data. The best performance of a single model, achieved an F1 score of 81.19 percent and an EM score of 95.54 percent. Finally, two ensemble approaches were employed to boost the F1 and EM scores to 84.28 percent and 95.14 percent, respectively.

The paper is structured as follows: section[II] explains the Dataset and feature engineering where we discussed about our dataset and feature extraction process. section[III] describes our proposed methodology. Experimental Results can be found In Section[IV] and Section[V] includes the conclusion and future work.

II. Dataset and Features

A. Dataset

One of the most difficult aspects of classification is obtaining adequately labeled data from which a representative training set for modeling may be taken. This conundrum is magnified for text-based or Natural Language Processing(NLP) problems (Djuric et al. 2015). Due to the subjective nature of inferring sentiment from textual communication, there is no choice but to rely on human labeling in the absence of a proper analytical labeling model (thus the requirement for the classification method). Although there are numerous big unlabeled text corpora accessible, human-labeled data is significantly most common (Radford et al. 2018).

This project's database was compiled on Kaggle, specifically in the "Jigsaw Unintended Bias for Toxicity Classification" competition. There are over 1804874 comments in the collection.

All samples present in these documents have been extracted from news sites comments, independently from the topic of the article. Figure 1 shows the distribution of toxicity.

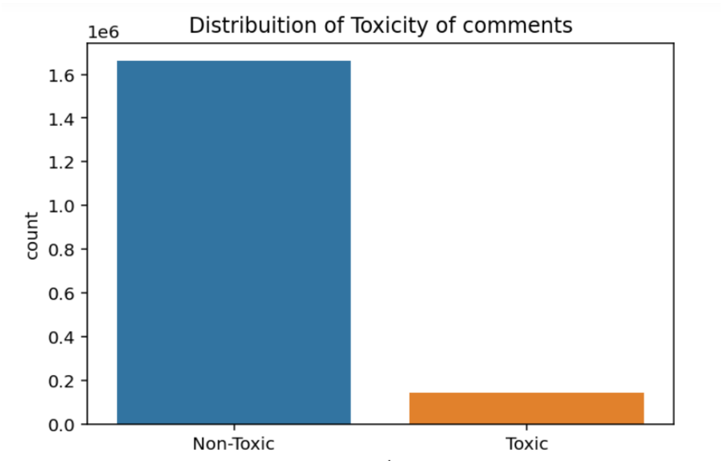


Fig. 1: Toxicity Distribution

Human raters identified and classified them into the following five categories:

- Severe Toxicity
- Threat
- Insult
- Identity-attack
- Obscene

There are less toxic comments than expected. We extracted 144,334 toxic comments from whole 1804874 different comments.Figure 2 depicts the distribution of toxic comments.

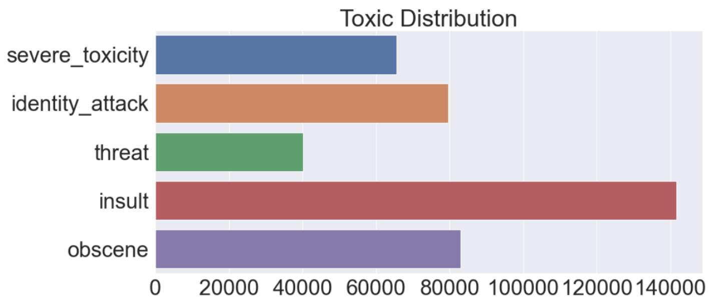
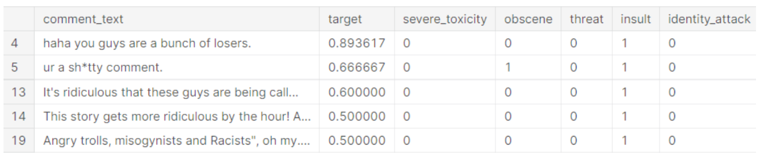


Fig. 2: Distribution of Toxic Comments

A csv document prepared to be charged in our model will have the following structure as shown in Figure 3:



	comment_text	target	severe_toxicity	obscene	threat	insult	identity_attack
4	haha you guys are a bunch of losers.	0.893617	0	0	0	1	0
5	ur a sh*tty comment.	0.666667	0	1	0	1	0
13	It's ridiculous that these guys are being call...	0.600000	0	0	0	1	0
14	This story gets more ridiculous by the hour! A...	0.500000	0	0	0	1	0
19	Angry trolls, misogynists and Racists", oh my....	0.500000	0	0	0	1	0

Fig. 3: Example of Database Format

B. Data Preprocessing

Following the dataset acquisition, the next critical step is to identify characteristics that will be utilized as input to the model for classification purposes. Finding the right features for any data mining job is really important. This procedure frequently entails a great deal of trial and error in order to determine which patterns in the data are truly indicative of each class. Finding features is typically thought of as an iterative process, and in our case, it was an experimental process.

Data was first pre-processed in order to extract features. Multiple morphological techniques, such as tokenization, stop-word removal, and stemming, were used to do the pre-processing. Tokenization is the process of breaking down sentences into individual tokens, each of which is a single word unit. Stop-words contain conjunctions, pronouns, assisting verbs, and other terms like is, am, are, he, she, and it that are abundant in the dataset but do not provide highly relevant information. The whole dataset is considerably reduced after such words are removed.

Furthermore, stemming reduces the quantity of data by mapping the word into the base word, which is not always a legitimate word.

Figure 4 shows the steps of pre-processing our data.

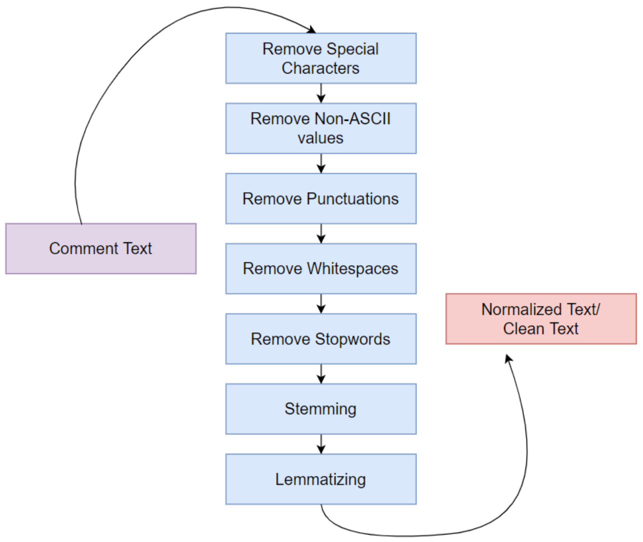


Fig. 4: Steps of Pre-processing

C. *Feature Engineering*

After data pre-processing, we experimented with different N-grams length. 1 gram, 2 gram, and 3 gram samples are researched and analyzed for feature analysis in this study. After further investigation, it was shown that 1-grams are superior.

By applying case-folding and stemming to the original dataset, the dataset was normalized. TFIDF scores were then calculated for each token. We chose the top 10,000 features with the highest TF-IDF scores. As a result of these characteristics, a comment was represented as a ten thousand-dimensional word-vector.As a result, these word vectors were eventually utilized to train and evaluate machine learning algorithms against each comment. We had a vocabulary V built with about 10,000 different words. We had a corpus of comments C containing 144,334 comments, each of which is represented as a string of words in order.

Stanford's GloVe (Pennington, Socher, and Manning 2014) and Facebook's fastText were the two embeddings employed. Both of these embeddings were built using neural networks and online text for training.

III. Methodology

A. *Models for Binary Classification*

The next stage is to build the machine learning model after feature extraction and engineering. It's vital to note that all of the proposed models have been designed to maximize performance while also allowing for the extraction of information about the network's decisions. We used following Four models for detecting toxicity from our data:

- Logistics Regression
- Multinomial Naive Bayes
- XGB Classifier
- SVM Classifier

Our target values are 0 or 1 because our goal is to detect toxicity. After extracting characteristics from each comment using the Tf-idf vectorizer, we developed the aforementioned classifier for detecting toxicity. The goal was to use the model using training data to apply all four models for each output column, and then make a prediction for test data. We then saved the results and performed a comparative analysis among them.

B. *Models for Multi-label Classification*

Figure 5 depicts a detailed overview of the system used to conduct the research. Data collection (gathering data from various sources), preprocessing (preparing the data for training), word embedding (using a pre-stored word-word embedding matrix), model training (training several models on the data), and performance analysis are the five components of the system.

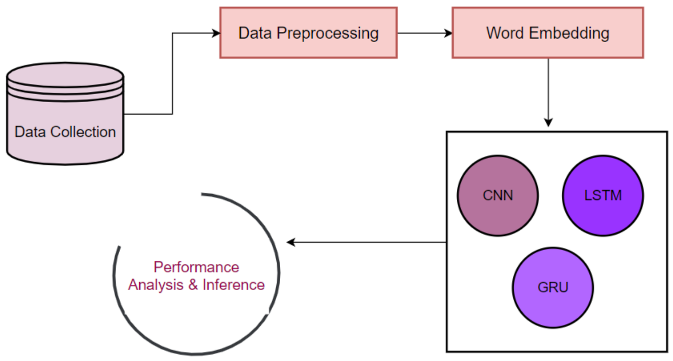


Fig. 5: Overview of the System

Given the need for a layer to achieve good NLP performance, the options considered for this project were GRU, LSTM, and CNN.

1) Convolution Neural Network: Convolutional Neural Networks are multistage trainable Neural Network topologies designed to solve classification problems. Our one-dimensional convolutional neural network was implemented using Keras with a TensorFlow backend (Abadi et al. 2016). The following Figure 6 is the general architecture of the networks we built.

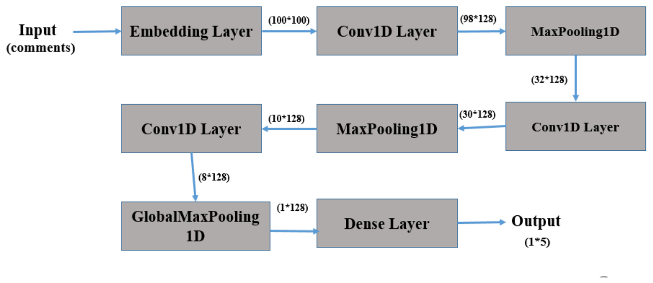


Fig. 6: Architecture of Our CNN

The comment was first loaded into the GloVe 100 dimension vectors' pretrained word embeddings. It's worth noting that this is where 90 percent of the model's complexity is hidden.

These word embeddings were then loaded into a three-layer CNN.Three one-dimensional convolutional layers were interleaved with max pooling layers to create our final model. MaxPooling reduces dimension, speeds up run time, and prevents overfitting. Because the vanishing gradient was computationally expensive, we applied ReLU activation over sigmoid between the layers to reduce execution time. Dropout into the dense layer is used in our final max pooling convolutional layer to regularize and further prevent overfitting (Srivastava et al. 2014).

For our loss, we used the categorical cross-entropy function, generally known as softmax loss (Gomez 2018).

2) LSTM: The goal of our LSTM model is to incorporate certain enhancements as well as the cell activation extraction. The samples are reduced in dimensionality from 3D (words, embedding, samples) in LSTM to 2D (highest embedding value, samples) in MaxPooling using 18 LSTM cells and a posterior layer of GlobalAveragePooling1D.Figure 7 shows the architecture of our LSTM System.

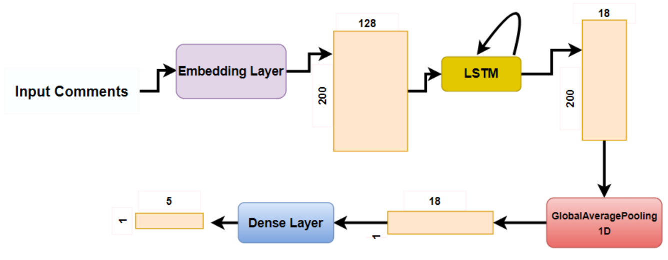


Fig. 7: Architecture of Our LSTM Model

To complete the network, a layer of length 5 is required, which corresponds to the toxicity categories.

3) GRU: The GRU model was intended to compare the performance and efficiency of a similar approach to the LSTM model. To make a fair comparison with LSTM, the GRU model does not include cell activations and additionally includes 18 GRU cells and GlobalAveragePooling. The GRU model architechture of our system are shown in Figure 8.

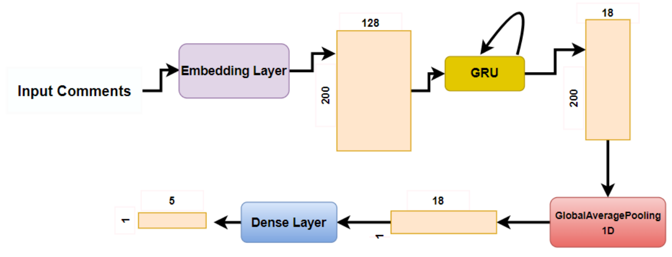


Fig. 8: Architecture of Our GRU Model

A layer of length 5 is required to complete the network, which corresponds to the toxicity categories. As a result, finishing the graph with several "Dense" normal layers is highly recommended.

IV. Experimental Result

A. Hyper-Parameter Settings

It is very important to specify the hyper-parameters of the models which are integral parts of the machine learning models. In this section, we present, the settings of the hyper parameters used in the machine learning models that were used during our experiments.

Models	Hyper parameter setting
CNN	Activation = relu, loss function = binary_crossentropy, optimizer = rmsprop, epochs = 10, batch_size = 1, verbose = 2, dense = 2
LSTM	Activation = sigmoid, lstm units=18, loss function= binary_crossentropy, return_sequences = true, optimizer = adam, epochs = 10, batch_size = 128, verbose=2, dense = 1
GRU	Activation = sigmoid, lstm units=18, loss function= binary_crossentropy, return_sequences = true, optimizer = adam, epochs = 10, batch_size = 128, verbose=2, dense = 1

TABLE I. Hyper Parameter

B. Dataset Distribution

The data was obtained via kaggle, as previously stated. After the preprocessing, the clean dataset was divided into two parts: training and testing. There were 1.8 million comments, but only 0.14 million were toxic comments. We partitioned the entire dataset into an 80 percent training set and a 20 percent test set for toxicity detection. The 1,44,000 toxic comments were separated into 90 percent as train set and 10 percent as test set for multi-label classification.

C. Evaluation Measures

In this section, we'll talk about our system's evaluation metrics, such as accuracy, recall, precision, and f1-score. The following are their descriptions:

Accuracy: The fraction of perfectly classified occurrences can be used to define accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where, TP stands for "True Positive", FP stands for "False Positive", TN stands for "True Negative" and FN stands for "False Negative".

Precision: Precision calculates the percentage of correct items among a set of items.

$$Precision = \frac{TP}{TP + FP}$$

Recall: The rate of accurate items selected is calculated via recall.

$$Recall = \frac{TP}{TP + TN}$$

F1-score: F1-score is considered as the weighted average of precision and recall.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

D. Results

We used 4 machine learning models for toxicity detection and 3 deep learning models for multi-label toxicity classification. Based on the Evaluation metrics value, the result of the 4 preliminary models is given below:

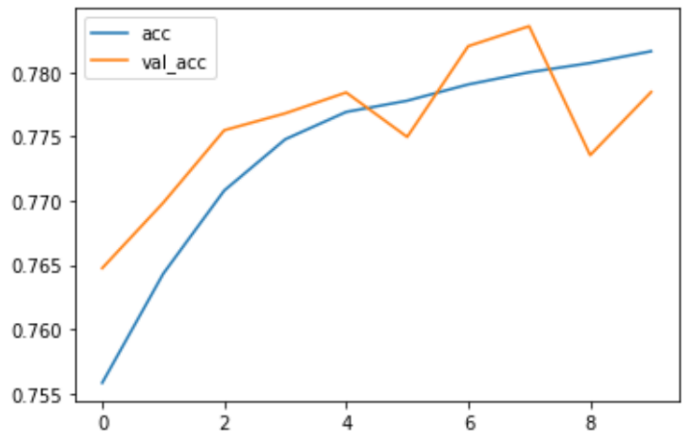
Model	F1 score	Accuracy	Precision	Recall
Log Regression	0.56	0.94	0.44	0.78
XGB Classifier	0.52	0.93	0.38	0.80
SVM Classifier	0.57	0.94	0.45	0.77
Multinomial NB	0.02	0.92	0.01	0.96

TABLE II. Preliminary models Result for Toxicity Detection

In terms of confusion matrix and accuracy, our SVM model marginally beats the other three models, as shown in the table II.

We experimented with Three Deep learning models for multi-label classification of toxic comments. Our dataset had over 144,000 toxic comments with toxicity label.The results of these models are presented below:

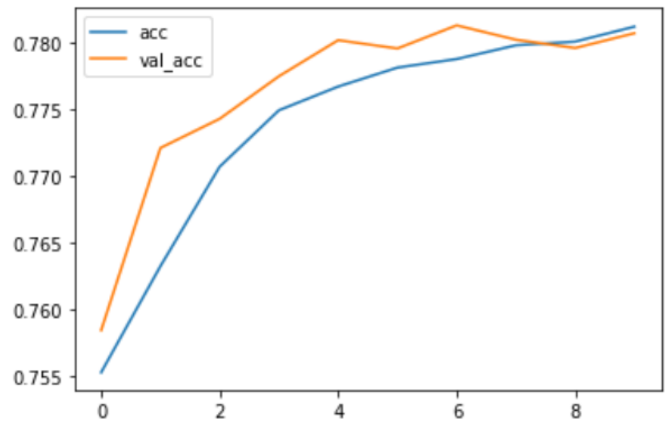
1) LSTM(Long Short Term Memory): Figure 9 shows Training and validating accuracies evolving with epoch.our LSTM model showed 78 percent validation accuracy.



```
Training Accuracy is   : 0.7816432118415833
Validation Accuracy is: 0.7835937142372131
```

Fig. 9: LSTM Model Accuracy Graph

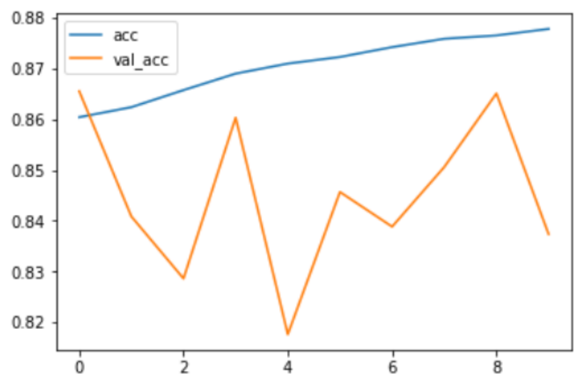
2) GRU(Gated Recurrent Unit): Our GRU model shows approximately same training and validation accuracy as LSTM. We can see it From Figure 10.



```
Training Accuracy is   : 0.7811755537986755
Validation Accuracy is: 0.7812727093696594
```

Fig. 10: GRU Model Accuracy Graph

3) CNN(Convolutional Neural Network): Figure 11 depicts that, CNN has performed pretty well with validation accuracy of 86 percent.



```
Training Accuracy is  : 0.8777745962142944
Validation Accuracy is: 0.8655211925506592
```

Fig. 11: CNN Model Accuracy Graph

CNN outperforms the other two neural network models in terms of accuracy among these three models. A comparison of these three models is shown below:

Model	Training Accuracy	Validation Accuracy
CNN	0.878	0.866
LSTM	0.782	0.784
GRU	0.7811	0.7812

TABLE III. Results of Toxicity Classification for different models

V. Conclusion and Future work

We have discussed different methods for identifying and classifying toxicity in this paper. We were able to show that all of the models can extract relevant information from text input. With the exception of SVM, which outperformed other models by a little margin, all preliminary models performed well in terms of toxicity identification. In contrast, CNN beats the other two models we evaluated.So, in a way, the potential methods for classifying toxicity levels were investigated in this study.

The most challenging goal of this thesis is to obtain useful information from neural network cells, and that is what makes it personal and distinctive. One of the primary topics explored during the research is the ability of convolutional neural networks to perform effectively in natural language processing issues. Because it is often used in video and image processing, this type of network did not appear to be a good fit for this project at first. This trait has been investigated and proven, providing insights on the similarities between image and sentence representation.

Our precision isn't up to par. In future, We need to figure out how to improve our model so that it can be used in a real-world system. The dataset can be increased even more by adding new classes to the existing model. When there are a huge number of classes to classify, it will be fascinating to see how the model performs.

References

Abadi, Martın, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. "Tensorflow: A system for large-scale machine learning." In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16),* 265–283.

Abubakar, Muhammad, Aminu Tukur, and Usman Bukar Usman. n.d. "AN IMPROVED MULTI-LABELED LSTM TOXIC COMMENT CLASSIFICATION."

Birkland, Thomas A. 2019. *An introduction to the policy process: Theories, concepts, and models of public policy making.* Routledge.

Djuric, Nemanja, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. "Hate speech detection with comment embeddings." In *Proceedings of the 24th international conference on world wide web,* 29–30.

Duggan, Maeve. 2014. "Online Harassment. Pew Research Center: Internet." *Science and Tech. Available online: http://assets. pewresearch. org/wp-content/uploads/sites/14/2014/10/PI_OnlineHarassment_* 72815.

Gomez, Raul. 2018. "Understanding categorical cross-entropy loss, binary cross-entropy loss, softmax loss, logistic loss, focal loss and all those confusing names." *URL: https://gombru. github. io/2018/05/23/cross entropy loss/(visited on 29/03/2019).*

Li, Hao, Weiquan Mao, and Hanyuan Liu. 2019. "Toxic Comment Detection and Classification." In *CS299 Machine Learning.* Standford University.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning. 2014. "Glove: Global vectors for word representation." In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP),* 1532–1543.

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. "Improving language understanding by generative pre-training."

Saif, Mujahed A, Alexander N Medvedev, Maxim A Medvedev, and Todorka Atanasova. 2018. "Classification of online toxic comments using the logistic regression and neural networks models." In *AIP conference proceedings,* 2048:060011. 1. AIP Publishing LLC.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15 (1): 1929–1958.

Wulczyn, Ellery, Nithum Thain, and Lucas Dixon. 2017. “Ex machina: Personal attacks seen at scale.” In *Proceedings of the 26th international conference on world wide web,* 1391–1399.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. “Xlnet: Generalized autoregressive pretraining for language understanding.” *Advances in neural information processing systems* 32.

Zaheri, Sara, Jeff Leath, and David Stroud. 2020. “Toxic comment classification.” *SMU Data Science Review* 3 (1): 13.