DETOX

A Toxic Content Classifier

Group P07

Akanksha Mohan

Rishi Rai

Shravani Yellu

Tushar Dey

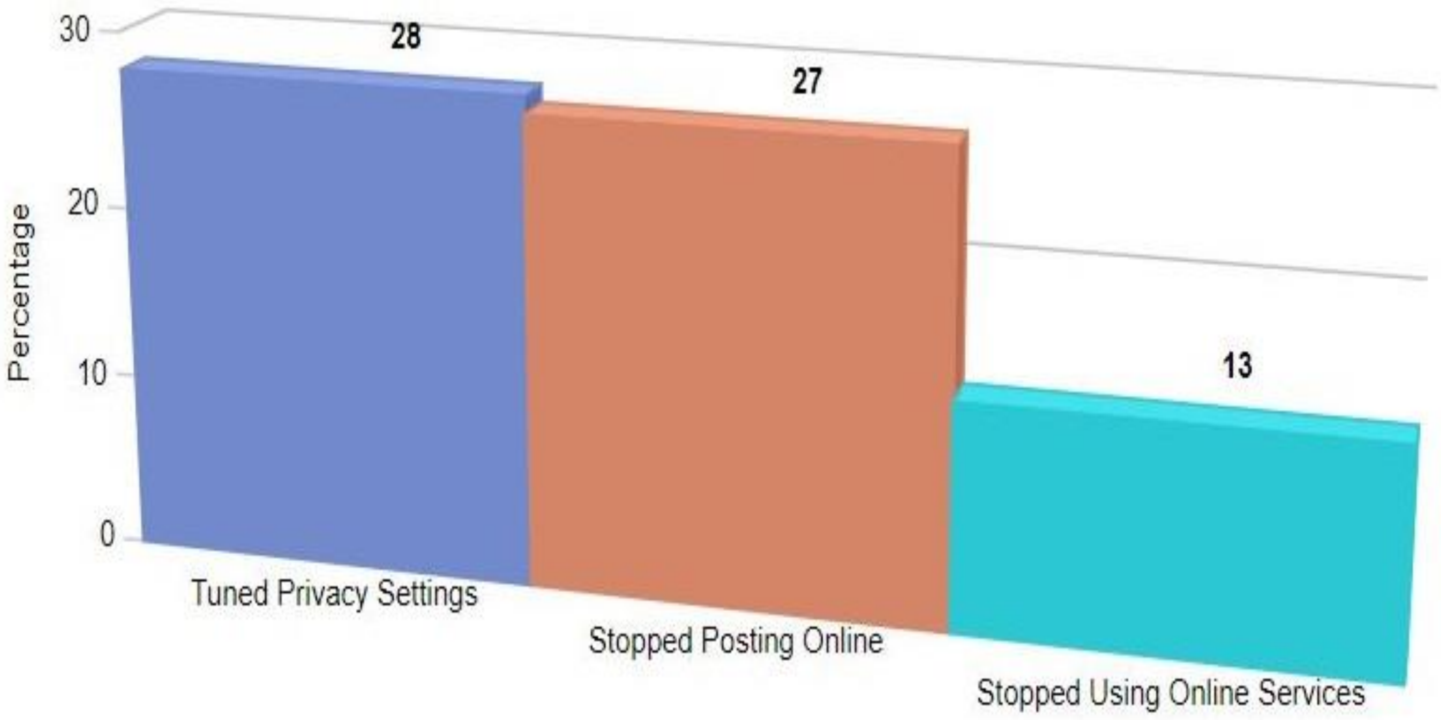
Motivation

"Words can inspire. And words can destroy. Choose yours well." - Robin Sharma

- **Four in ten** U.S. adults have been harassed online.

- Among those who've been harassed, about **18%** of U.S. adults said they have been the target of severe behaviors such as **physical threats and sexual harassment**.

- **23%** of online harassment targets say their most recent experience occurred in the **comments sections of a website**.



Problem Statement

Online interactions provides platforms for more diverse and constructive conversations. However, it is being plagued by bigoted people spewing racist and harmful believes.

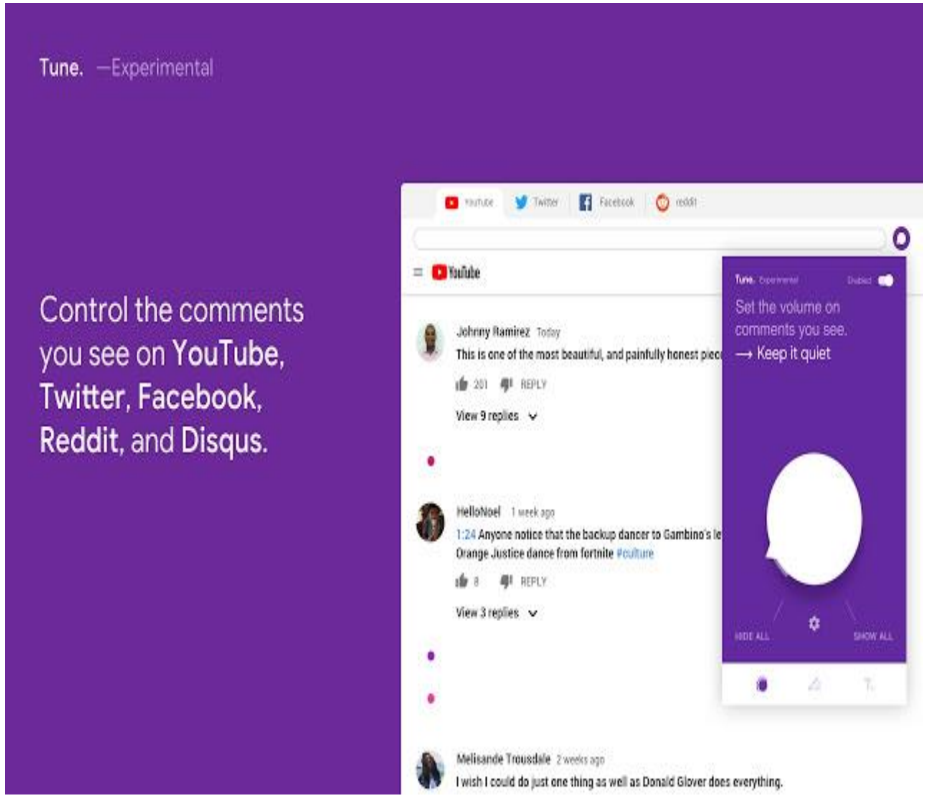
Challenge

Monitoring the conversation through human resource is immensely costly due to the large user base on the social media platforms.

Potential Resolution

Training machine learning models to be able to identify the toxic comments which are soiling social media interactions in order to create a safe online experience.

Monitoring Social Interactions



Previous Related Work

- Traditionally classification approaches followed the classical **two stage** scheme of **extraction of (handcrafted) features**, followed by **classification** using a **traditional machine learning model**.[1][2][3][4]
- Prominent paper by Kim et. al.[5] implemented **CNN** for sentence classification, which **improved** upon **the state of art** for multiple **NLP task.**
- Other deep learning architecture like **LSTM**[6][7] has shown to preform exceptionally well for the text classification task.
- Learned **vector representation**[8][9] for words has been extensively used with these deep architecture, improving their performance.

Data Set

- The **Toxic Comment Classification** consists of 159572 Wikipedia comments which were labelled by human raters. The comments are classified as **toxic**, severe toxic, obscene, threat or insult. The data is **skewed** as less than **10%** of the comments belongs to toxic class.
- We divide data in to **80%** for training set and **20%** for test set.

Here are the classes in the dataset, as well as random comments from each class:

I will *** your family, you ba**** **** you, block me, you ****!	**TOXIC**
How they achieved the speed? You ignored him too	NON-TOXIC

Research Questions

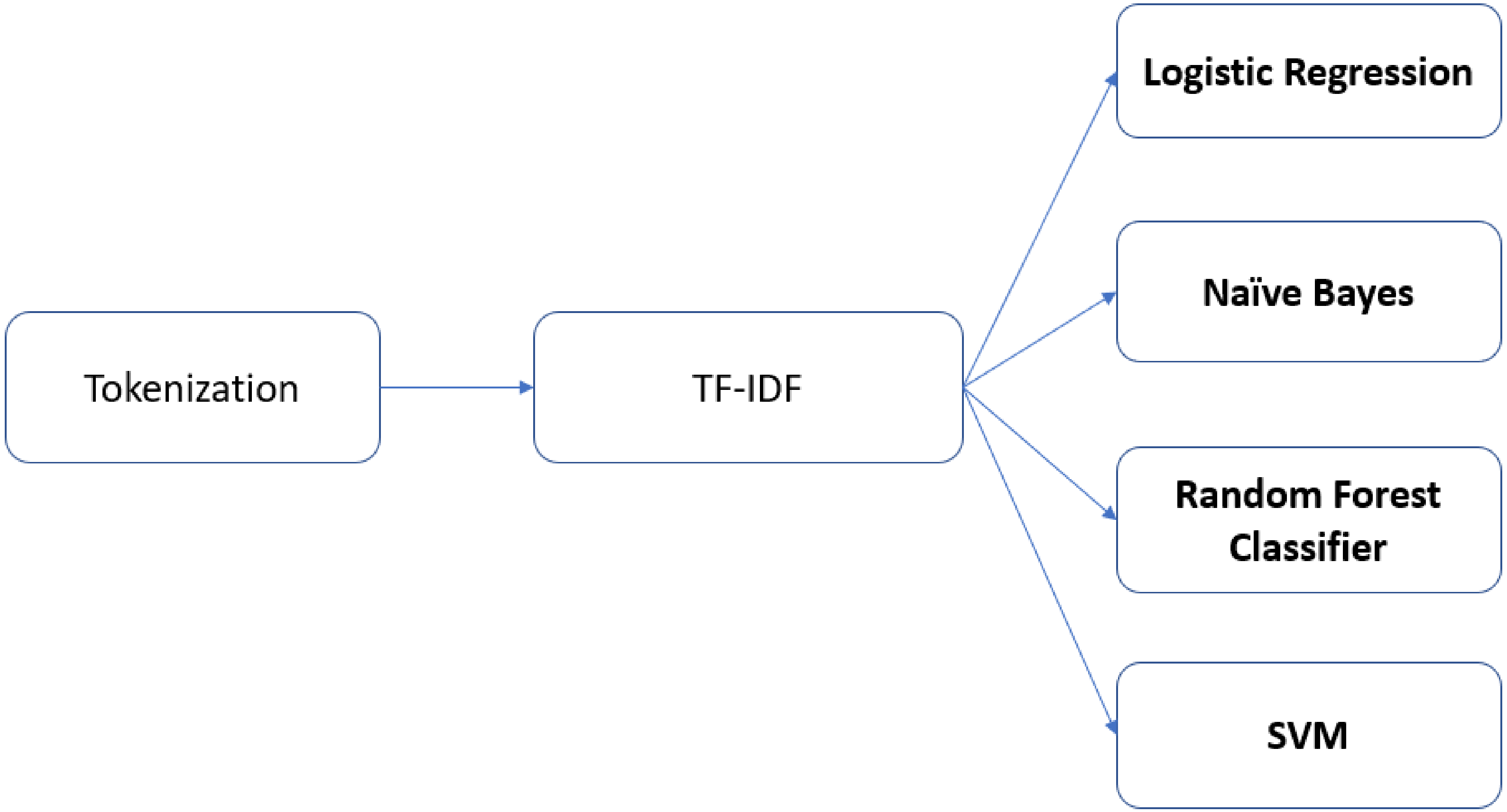
- **Can we architect various deep learning models for our problem so that we can achieve high accuracy on classification?**
- **Can we adequately address the issue of skewness in our data, while comparing various models' robustness?**
- **Can we gather some insight on our data from the results of the models?**

Handling Skewness

To combat with the issue of this highly skewed data we did the following :

- Shifting our focus from **Accuracy to Recall** for the minority class
- **Over Sampling** for the minority class

Traditional Models



Word Embeddings

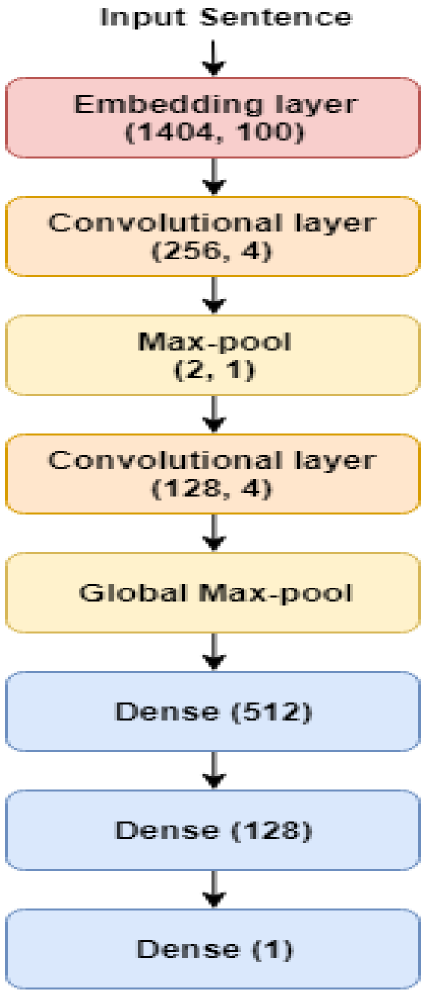
- In all of our deep learning models we attached an embedding layer which is a neural network architecture that computes the vector representation for each word in a sentence. This embedding layer learns the vector representation along with the network to produce task specific vectors [8] [9].
- Other than this we used GLOVE, which uses global matrix factorization and local context window methods to obtain linear substructures of the word vector space [10].

Model	Word vectorization method used
Traditional model	TF-IDF vectorization
Multilayer perceptron model	TF-IDF vectorization, Learned embedding
Convolutional Neural Network	Learned embedding, Glove with Tuning
LSTM	Learned embedding, Glove with Tuning

CNN Architecture

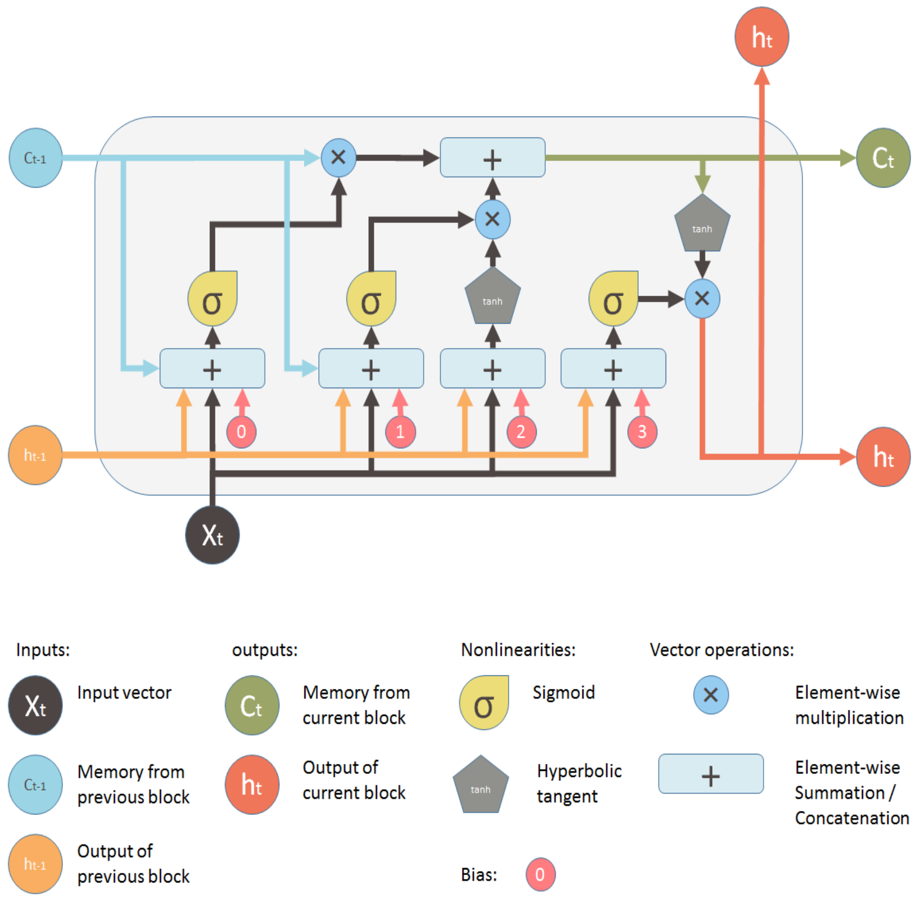
Training parameters:

- Cost function: **Binary Cross-Entropy**
- Optimizer : **ADADELTA**
- Batch size: **512**
- Epochs : **30**



Motivations For Using LSTM

- RNN models are powerful tools as memory component is presented in each node[6].
- Major Problem - Vanishing Gradient or exploding gradient problem.
- LSTM(Long Short-term Memory) is one variation of RNN which solves it[7].
- 4 Components- Cell, Input gate, Output gate and Forget gate.

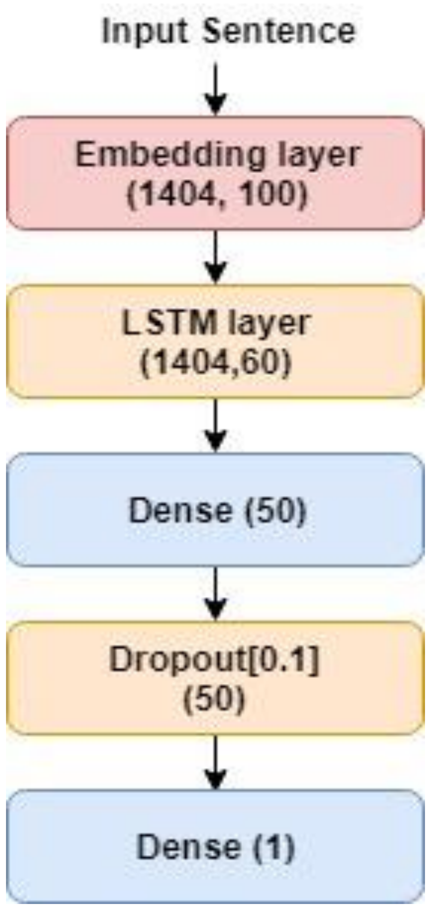


LSTM MODEL

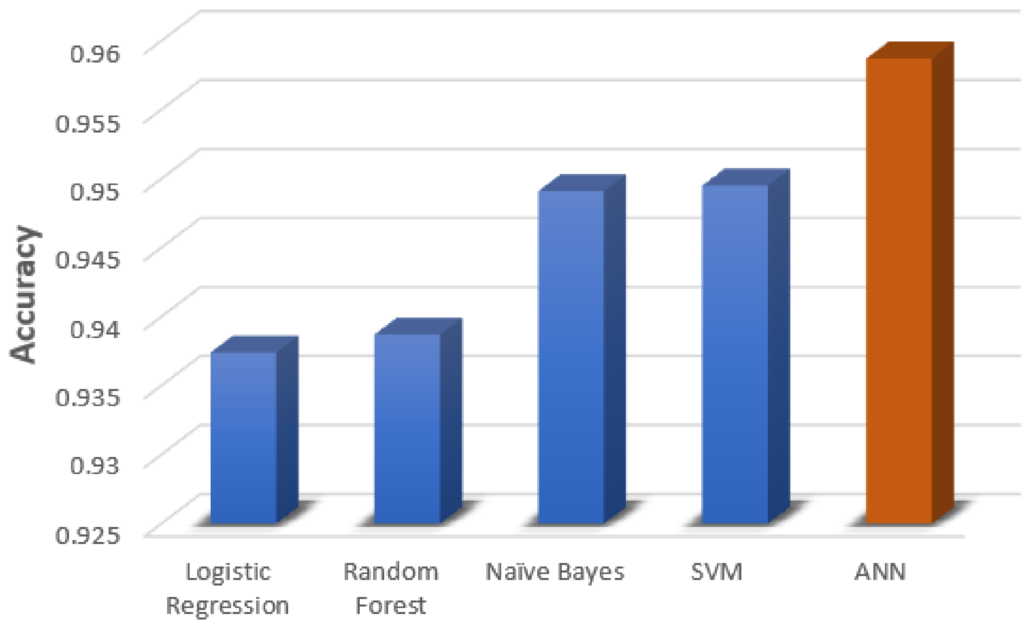
LSTM Architecture

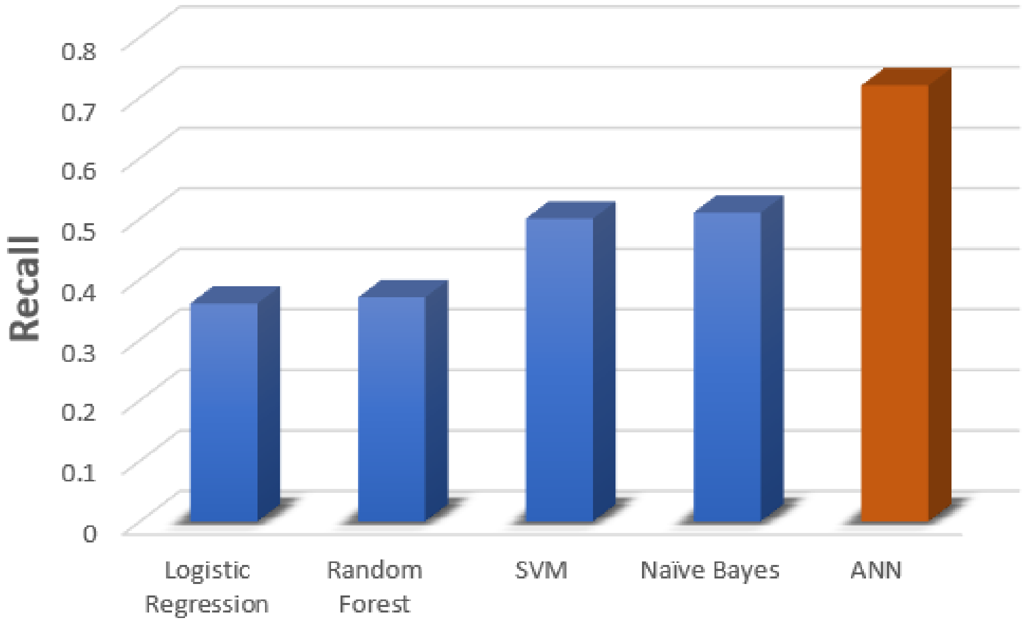
Training parameters:

- Cost function: **Binary Cross-Entropy**
- Optimizer : **ADADELTA**
- Batch size: **512**
- Epochs : **10**

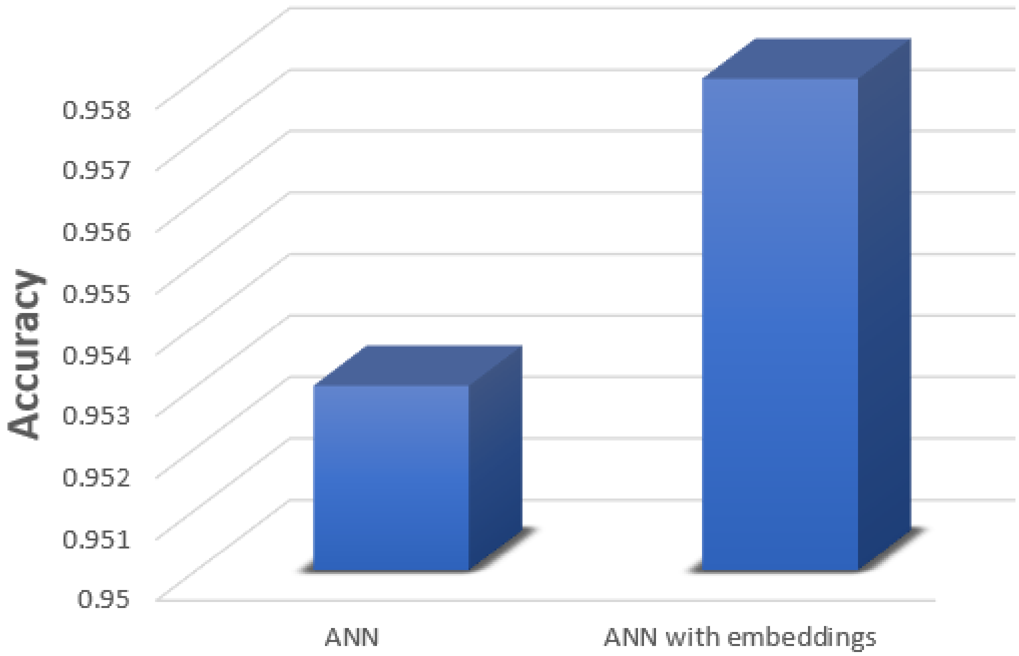


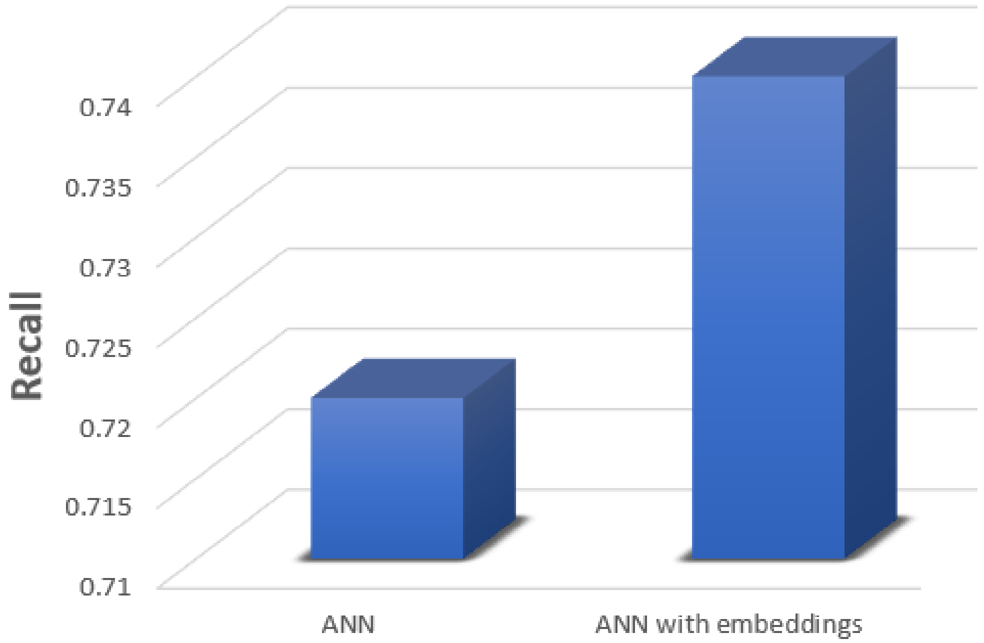
Traditional VS ANN





Bag of words VS learned embeddings

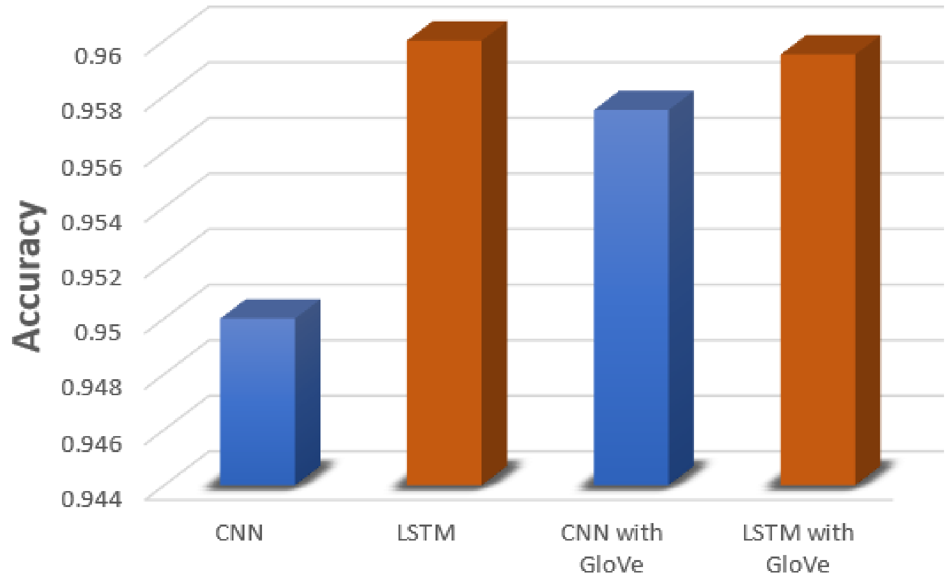


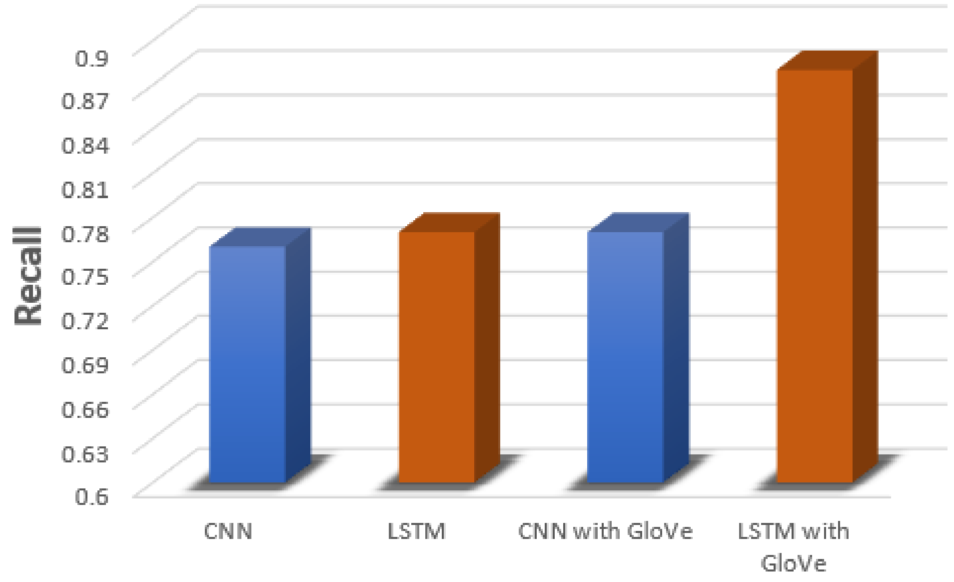


Bag of words VS learned embeddings

Sentence	ANN1 Confidence	ANN 2 confidence
He had killed many people in the past and would continue to kill if we do not do something about it	**0.921**	**0.288**
In a time were rape and harassment are widespread, we need to stand up against the bullies.	**0.918**	**0.464**
The word **** is very inappropriate please refrain from using. It leaves a bad impression.	**0.999**	**0.992**
I will kill your family, nice good wonderful.	**0.762**	**0.954**

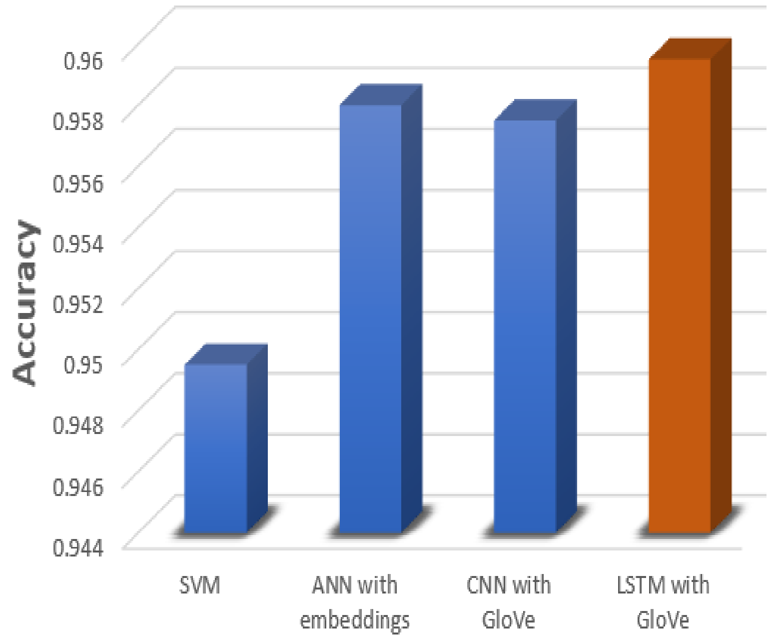
CNN & LSTM – GLOVE VS Learned Embedding

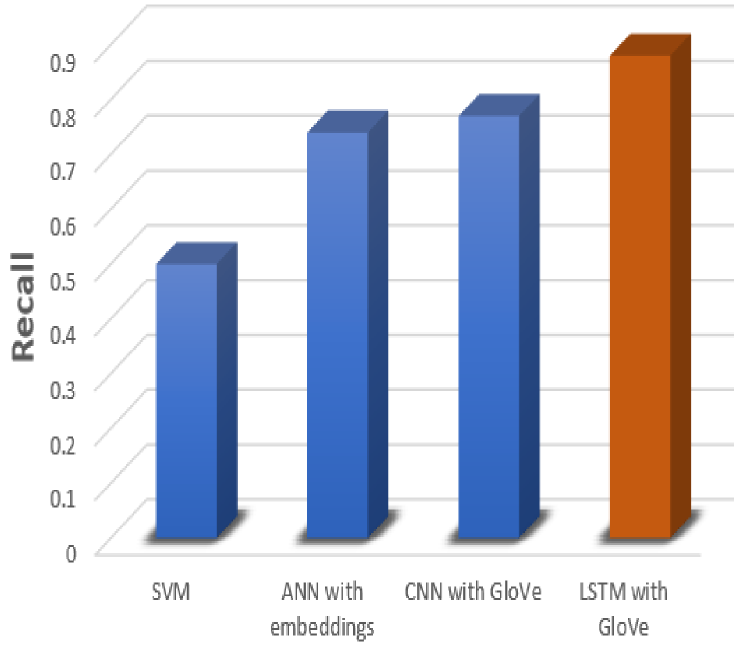


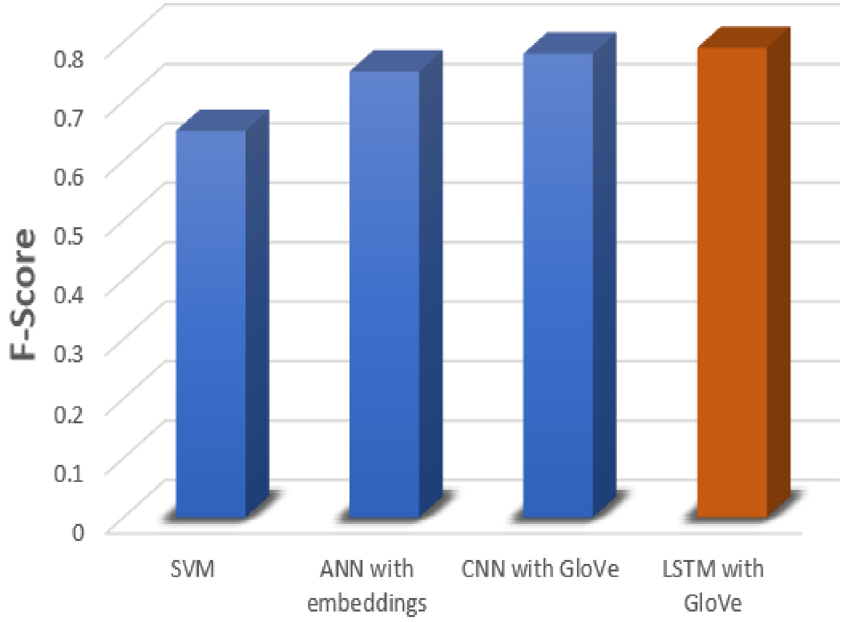


Models	CNN	CNN With GloVe	LSTM	LSTM With GloVe
F-Score	0.742	0.783	0.791	0.798

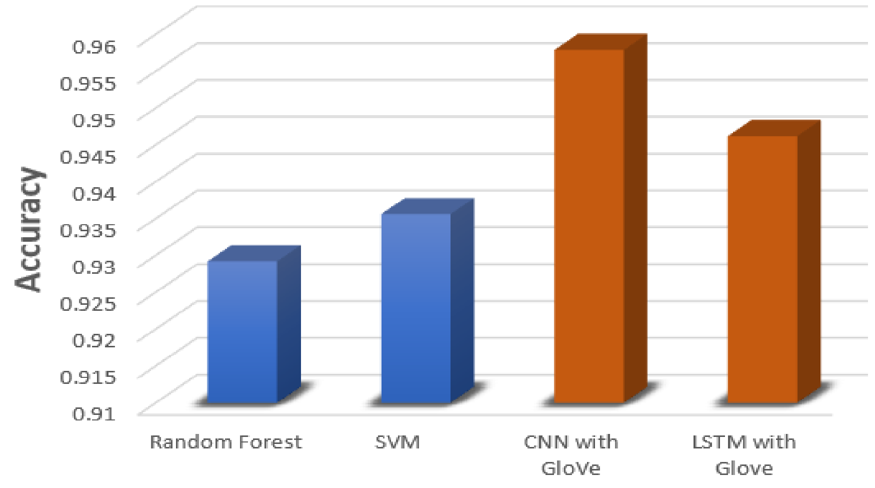
Traditional VS ANN VS CNN VS LSTM

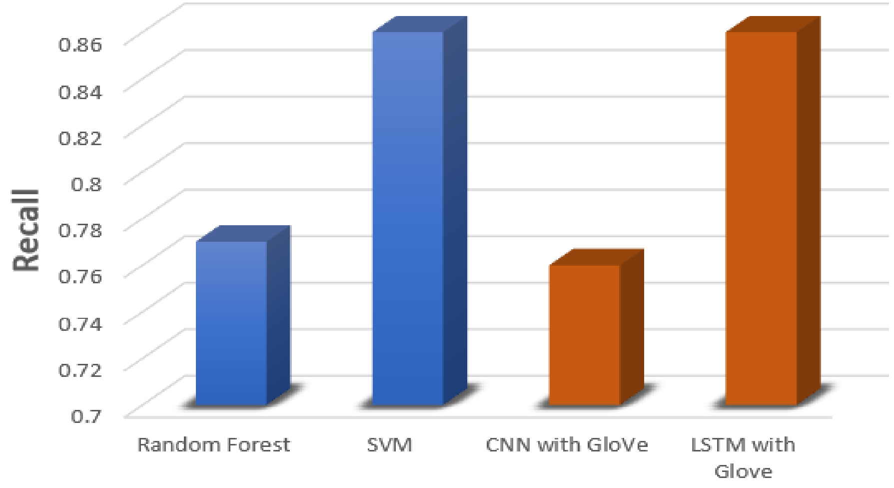


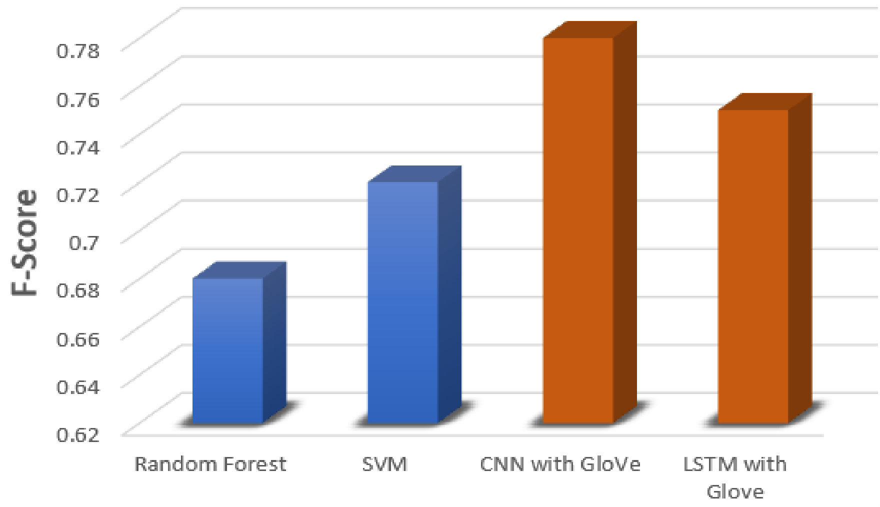


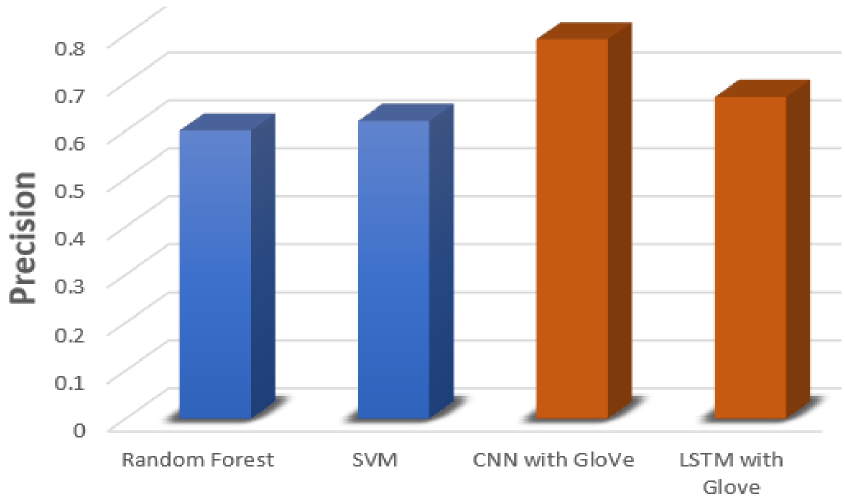


Over Sampling Minority Classes









Conclusion

- We designed and implemented CNN and LSTM model for our classification problem.
- Deep learning models **performs best** when the words are represented using **GLOVE embedding** and is further **tuned** for our dataset.
- **Deep learning models** are more **robust against skewed data** and since our data is highly skewed, oversampling along with deep learning architecture performs the best.

References

[1] Joachims, Thorsten. "Text categorization with support vector machines: Learning with many relevant features." European conference on machine learning. Springer, Berlin, Heidelberg, 1998.

[2] Das, Bijoyan, and Sarit Chakraborty. "An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation." arXiv preprint arXiv:1806.06407 (2018).

[3] Kim, Sang-Bum, et al. "Some effective techniques for naive bayes text classification." IEEE transactions on knowledge and data engineering 18.11 (2006): 1457-1466

[4] Ifrim, Georgiana, Gökhan Bakir, and Gerhard Weikum. "Fast logistic regression for text categorization with variable-length n-grams." Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008

[5] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

[6] Lee, Ji Young, and Franck Dernoncourt. "Sequential short-text classification with recurrent and convolutional neural networks." arXiv preprint arXiv:1603.03827 (2016).

[7] Yin, Wenpeng, et al. "Comparative study of cnn and rnn for natural language processing." arXiv preprint arXiv:1702.01923 (2017).

[8] Collobert, Ronan, et al. "Natural language processing (almost) from scratch." Journal of machine learning research 12.Aug (2011): 2493-2537.

[9] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[10] B. Kulis and T. Darrell, “Learning to hash with binary reconstructive embeddings,” *Nips*, pp. 1–9, 2009.

[11] Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014

Thank You

Questions?