Web-based Chat Toxicity Classification in Counter-Strike: Global Offensive Using LSTM

NEAL PATRICK SAN ANDRES | CEASAR DELIZO | ANGELICA FORDELON

I.INTRODUCTION

VIDEO GAMES

- Casual games
- Competitive games

COUNTER-STRIKE: GLOBAL OFFENSIVE

first-person shooting video game developed by Valve that is based on the original Half-Life MOD of the game that was introduced 19 years ago

I.INTRODUCTION

STEAMCHARTS *An ongoing analysis of Steam's concurrent players.*

Counter-Strike: Global Offensive

Month	Avg. Players	Gain	% Gain	Peak Players
Last 30 Days	*598,378.7*	*+3,925.7*	*+0.66%*	*936,192*
July 2022	594,453.1	+22,230.6	+3.88%	928,329
June 2022	572,222.5	+7,260.8	+1.29%	906,670

This is the latest update on CSGO's concurrent players as of August 10, 2022.

For more updates, the chart is accessible at the steamcharts website https://steamcharts.com/app/730

Harassment by Game. Valorant and DOTA 2 players reported the most harassment for the second consecutive year, at 79% and 78%, respectively.

Source: Hate is No Game: Harassment and Positive Social Experiences in Online Games 2021

Game	2019	2020	2021
Valorant	N/A	80%	79%
DOTA 2	79%	80%	78%
Overwatch	75%	62%	75%
Call of Duty	N/A	75%	74%
Counter-Strike: Global Offensive	75%	75%	74%
PlayerUnknown's Battlegrounds	75%	67%	72%
Grand Theft Auto (GTA)	70%	76%	71%
Fortnite	70%	74%	68%
World of Warcraft	72%	66%	66%
Apex Legends	64%	N/A	66%
League of Legends	75%	73%	65%
Among Us	N/A	N/A	62%
Roblox	N/A	61%	61%
Clash Royale	N/A	65%	61%
Madden NFL	67%	66%	60%
Rocket League	66%	76%	59%
Minecraft	51%	53%	46%

ADL
FIGHTING HATE FOR GOOD

According to ADL's study in 2021, CSGO is ranked in the 5th place in 2021 of having 74% of their players experience harassment when playing games

Purpose and Objective of the Study

THE GENERAL OBJECTIVE OF THIS STUDY IS TO CLASSIFY THE TOXICITY IN COUNTER-STRIKE: GLOBAL OFFENSIVE . SPECIFICALLY, THE STUDY AIMS:

- *TO CREATE A STRUCTURED DATA COLLECTION PROCESS FROM THE CSGO MATCHES THAT CONTAINS NON-TOXIC OR TOXIC CHAT DATA THAT IS BEING GENERATED BY THE PLAYERS.*
- *TO DEVELOP A WEB APPLICATION THAT CAN CLASSIFY THE CONTENTS OF THE CHAT BASED ON ITS TOXICITY BY THE USE OF A MULTI-LABEL BINARY CLASSIFIER.*
- *TO FINE TUNE AND CONTINUALLY OPTIMIZE THE MODEL UP TO AND BEYOND LITERATURE RECORD ACCURACY WITHOUT USING PRE-TRAINED WORD EMBEDDINGS.*

Scope and Limitation

The study covers:

- ***Developing web applications that can classify the toxicity in the content of text chat in Counter-Strike: Global Offensive using Bi-directional LSTM.***
- ***Combination of dataset extracted from FaceIt and Kaggle will be used in the study.***
- ***Emphasizing the use of built-in word embedding from own training model using Keras.***
- ***The study will be using English language as a medium for the text classifier***
- ***Precision, Recall, and Accuracy performance metrics***

The study will not cover:

- ***Other games like Defense of the Ancients (DOTA) and League of Legends (LOL)***
- ***Team voice, team chats***

II. LITERATURE REVIEW

A. Faculty voice: Gaming and toxicity

B.New Unity study shows just how toxic online gaming can be

C.Toxicity in video games: Analyzing cause and effects of abusive online behavior

D.Toxicity Detection in Multiplayer Online Games

E.Natural Language Processing for Games Studies Research

II. LITERATURE REVIEW

F. Framewise phoneme classification with bidirectional LSTM and other neural network architectures

G. Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model

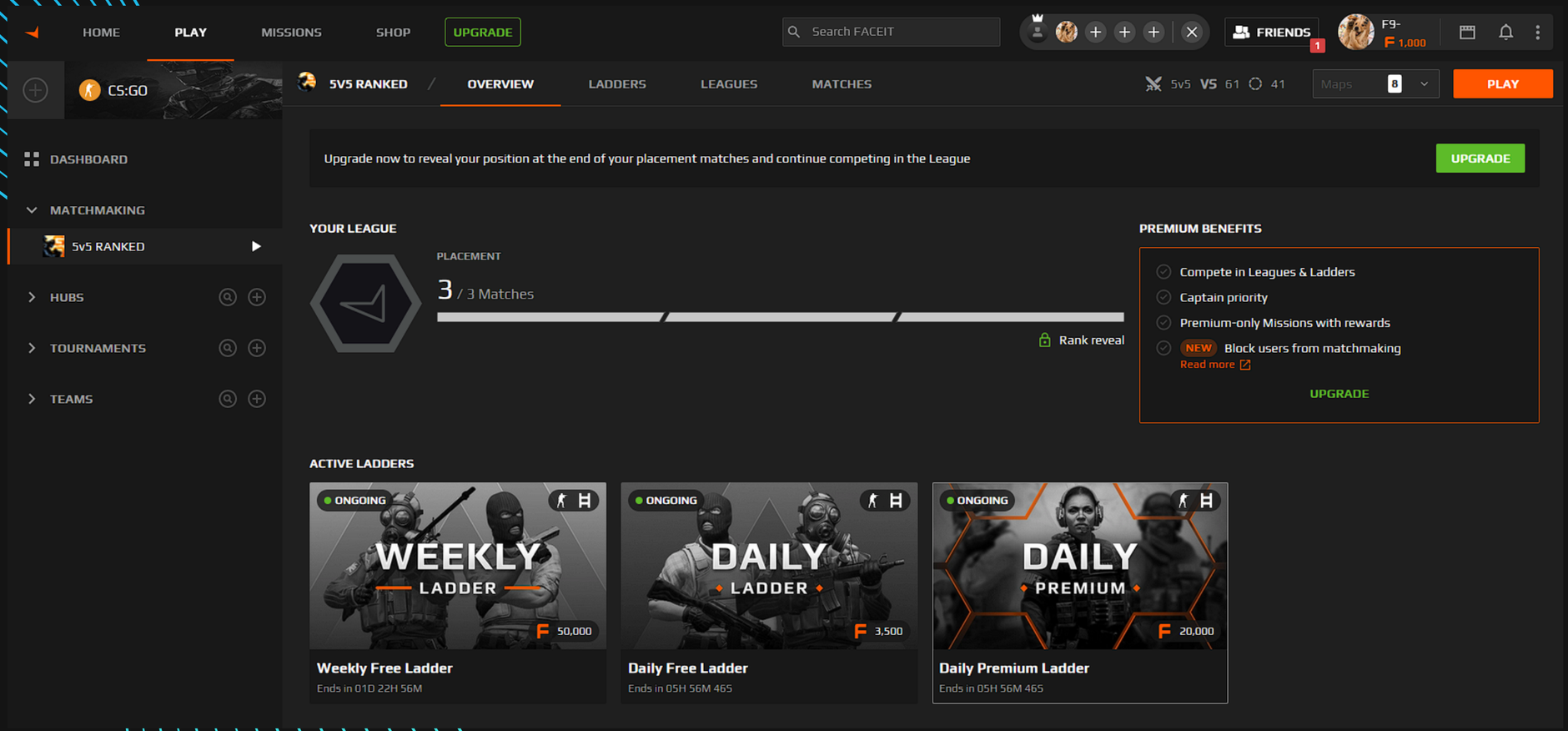
H. LSTM Neural Networks for Language Modeling

I. Couples Behavior Modeling and Annotation Using Low-Resource LSTM Language Models

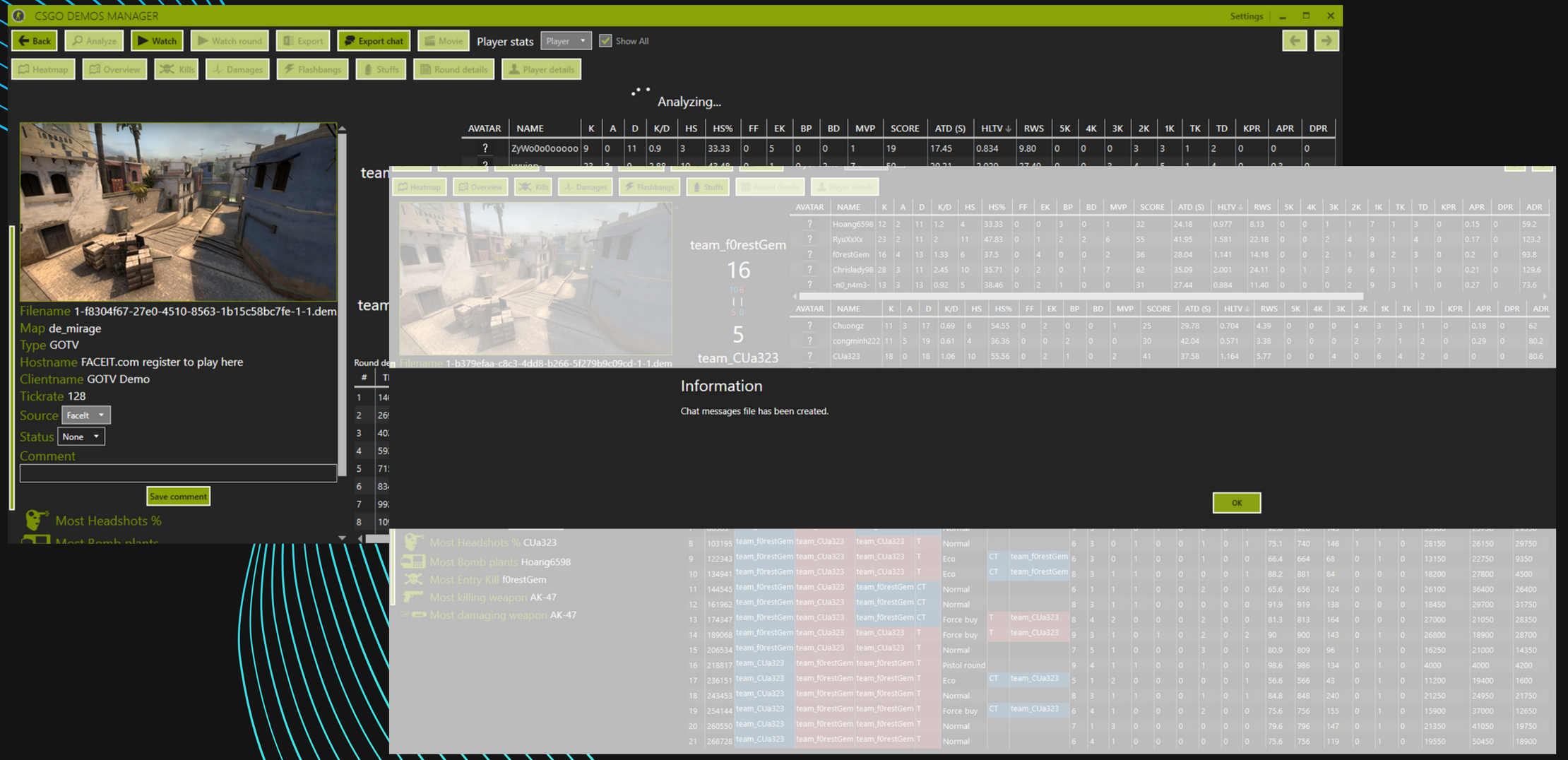
J. An Automated Toxicity Classification on Social Media Using LSTM and Word Embedding

II. Research MEthodology

Dataset from FaceIT Matches



Demo manager to export chats



Keras

It is a complex neural network API which can run in the framework of TensorFlow to create deep learning models.

Pre-processing Dataset

Cleansed dataset

128604	ffe987279560d7ff	baiter fucking idiot	0	1	1
128605	ffea4adeee384e90	You should be ashan	0	0	0
128606	ffee36eab5c267c9	fking noob	0	1	1
128607	fff125370e4aaaf3	And it looks like it w	0	0	0
128608	fff46fc426af1f9a	"	0	0	0

120k comments

```
# Remove numbers, capital letters, punctuation, '\n'
import re
import string

# Remove all random numbers with attached letters
alphanumeric = lambda x: re.sub('\w*\d\w*', ' ', x)

# '[%s]' % re.escape(string.punctuation),' ' - replace punctuation with white space
# .lower() - Convert all strings to lowercase
punc_lower = lambda x: re.sub('[%s]' % re.escape(string.punctuation), ' ', x.lower())

# Remove all '\n' in the string and replace it with a space
filter_line = lambda x: re.sub("\n", " ", x)

# Remove all Non-ASCII characters
filter_non_ascii = lambda x: re.sub(r'[^\x00-\x7f]',r' ', x)

# Apply all the lambda functions wrote previously through .map on the comments column
df['TextComment'] = df['TextComment'].map(alphanumeric).map(punc_lower).map(filter_line).map(filter_non_ascii)
```

Pre-processing Dataset

The dataset will be run using the Jupyter Notebook. A python code consisting of **RE (regular expression)** and **Lambda function** will be used to preprocess the ***train.csv*** dataset to remove non-alphanumeric such as punctuations and newline (\n) on comments.

Pre-processing Dataset

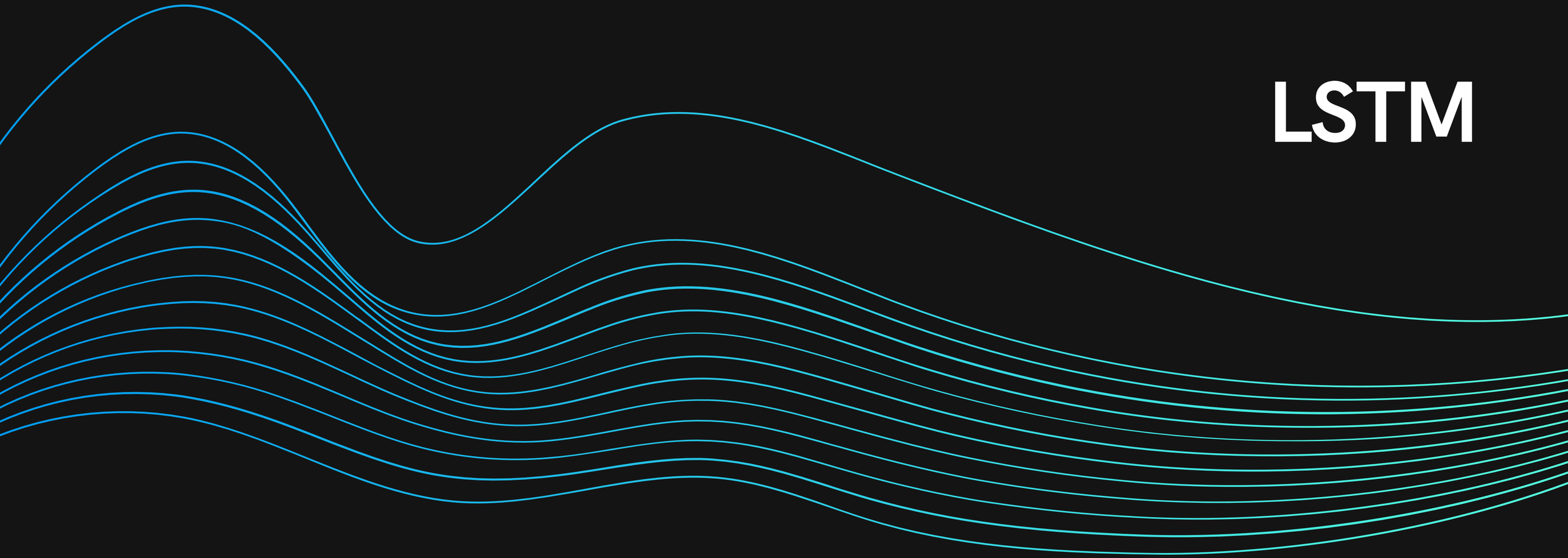
Tokenization

It will be used to preprocess the data gathered in the study. It is the process of splitting the string into tokens.

Vectorization and Vocabulary

The model converts the string into numerical values. It needs to learn the vocabulary by analyzing it and determining the occurrence of the values.

LSTM



FORGET GATE

It is responsible for keeping or forgetting the information from the previous timestamp.

INPUT GATE

It is responsible for weighing the importance of new information from the input.

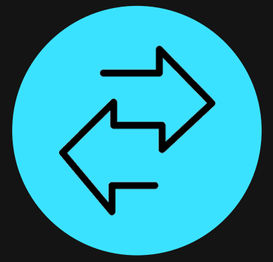
OUTPUT GATE

It is responsible for determining the value of the next hidden state

III. RESEARCH METHODOLOGY

Embedding layer

Epoch



Bidirectional
LSTM

Dense

Dropout

Evaluation and validation

Accuracy

$$\text{Accuracy} = \frac{True\ Positive + True\ Negative}{True\ Positive + False\ Positive + False\ Negative + True\ Negative}$$

Precision

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

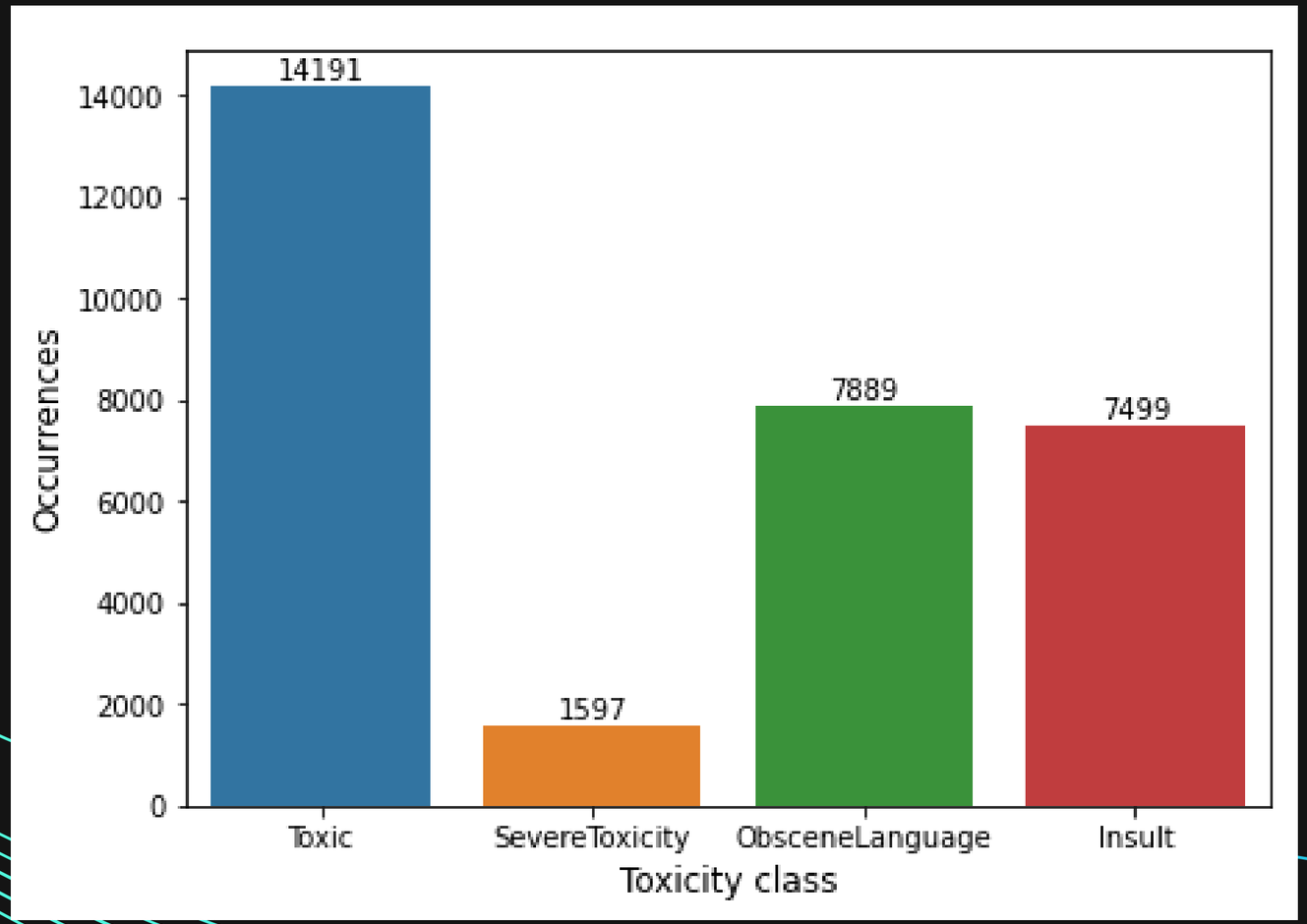
Flask

It is a web framework used to build a user interface for this study in classifying the toxicity in chats

IV.RESULTS AND DISCUSSION

A. The Developed Model

- The study used 128,606 chats that can be used to classify the toxicity. 70% were used for training sets, 20% were used for validation, and 10% were used for testing.



Toxic = 14,191 (11.03%)

Severe Toxicity=1,597 (1.24%)

Obscene Language=7,889 (6.13%)

Insult=7,499 (5.83%)

Optimizers used in each model

Model	Dropout	Dense	Epoch
Bi-directional LSTM 1	none	128-256	1 epoch
Bi-directional LSTM 2	0.4	256-128	1 epoch
Bi-directional LSTM 3	0.2	128-64	10 epochs
Bi-directional LSTM 4	0.3	128-64	10 epochs
Bi-directional LSTM 5	0.3	128-256	10 epochs
Bi-directional LSTM 6	0.2	128-256	10 epochs
Bi-directional LSTM 7	0.2	128-256	20 epochs

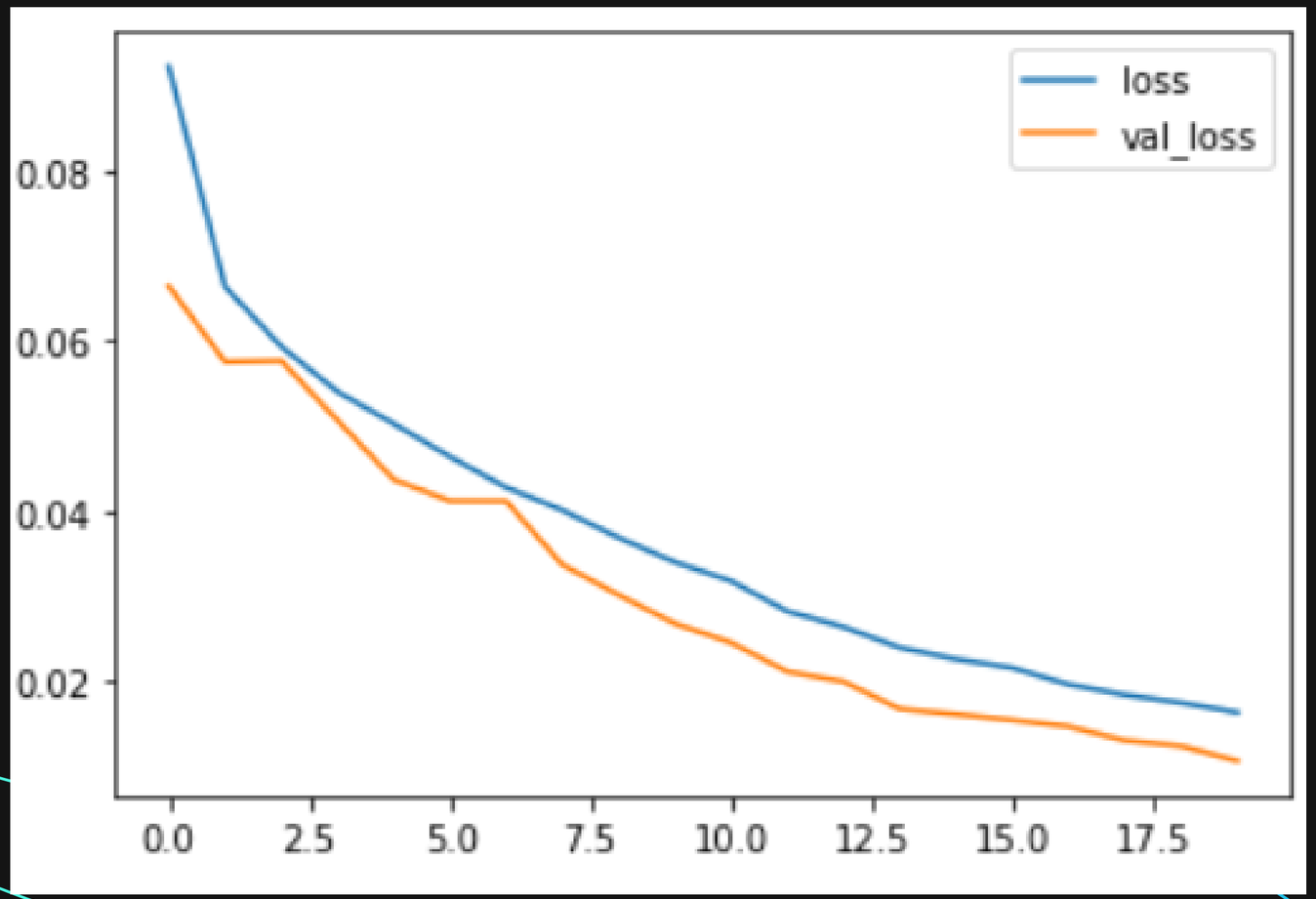
Evaluation of each model

Model	Precision	Recall	Accuracy
Bi-directional LSTM 1	82%	70%	46%
Bi-directional LSTM 2	86%	63%	46%
Bi-directional LSTM 3	84%	80%	49%
Bi-directional LSTM 4	83%	92%	50%
Bi-directional LSTM 5	92%	92%	47%
Bi-directional LSTM 6	92%	91%	67%
Bi-directional LSTM 7	97%	96%	86%

```
# Train the dataset
# Epoch = how many passes for dataset
batch_size=64
archive=model.fit(train, epochs=20, validation_data=val)
```

```
Epoch 1/20
5626/5626 [==============================] - 451s 79ms/step - loss: 0.0924 - val_loss: 0.0665
Epoch 2/20
5626/5626 [==============================] - 440s 78ms/step - loss: 0.0664 - val_loss: 0.0576
Epoch 3/20
5626/5626 [==============================] - 438s 78ms/step - loss: 0.0594 - val_loss: 0.0577
Epoch 4/20
5626/5626 [==============================] - 439s 78ms/step - loss: 0.0540 - val_loss: 0.0506
Epoch 5/20
5626/5626 [==============================] - 437s 78ms/step - loss: 0.0502 - val_loss: 0.0437
Epoch 6/20
5626/5626 [==============================] - 438s 78ms/step - loss: 0.0464 - val_loss: 0.0411
Epoch 7/20
5626/5626 [==============================] - 439s 78ms/step - loss: 0.0427 - val_loss: 0.0411
Epoch 8/20
5626/5626 [==============================] - 438s 78ms/step - loss: 0.0400 - val_loss: 0.0336
Epoch 9/20
5626/5626 [==============================] - 439s 78ms/step - loss: 0.0368 - val_loss: 0.0300
Epoch 10/20
5626/5626 [==============================] - 439s 78ms/step - loss: 0.0339 - val_loss: 0.0267
Epoch 11/20
5626/5626 [==============================] - 438s 78ms/step - loss: 0.0317 - val_loss: 0.0244
Epoch 12/20
5626/5626 [==============================] - 439s 78ms/step - loss: 0.0281 - val_loss: 0.0210
Epoch 13/20
5626/5626 [==============================] - 440s 78ms/step - loss: 0.0263 - val_loss: 0.0198
Epoch 14/20
5626/5626 [==============================] - 438s 78ms/step - loss: 0.0238 - val_loss: 0.0166
Epoch 15/20
5626/5626 [==============================] - 439s 78ms/step - loss: 0.0225 - val_loss: 0.0160
Epoch 16/20
5626/5626 [==============================] - 438s 78ms/step - loss: 0.0214 - val_loss: 0.0153
Epoch 17/20
5626/5626 [==============================] - 441s 78ms/step - loss: 0.0195 - val_loss: 0.0145
Epoch 18/20
5626/5626 [==============================] - 440s 78ms/step - loss: 0.0183 - val_loss: 0.0129
Epoch 19/20
5626/5626 [==============================] - 442s 79ms/step - loss: 0.0173 - val_loss: 0.0122
Epoch 20/20
5626/5626 [==============================] - 443s 79ms/step - loss: 0.0162 - val_loss: 0.0105
```

Training data of the Bidirectional LSTM-7 Model

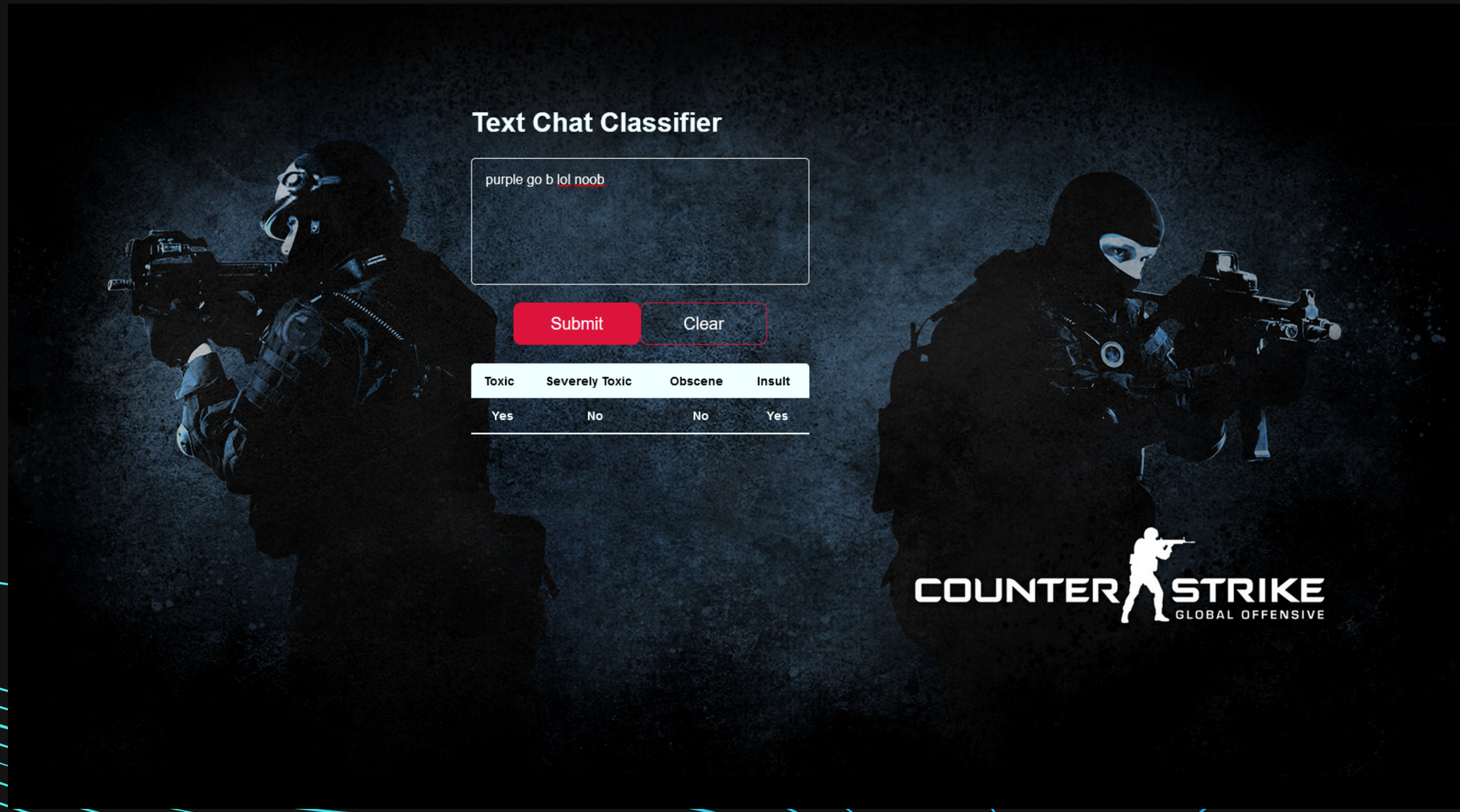


Loss and val_loss of Bidirectional LSTM 7 Model

A. The Developed Model

- To develop a computerized model that classifies the toxicity of the chat.

B. Toxicity Classifier GUI (web app)



V. CONCLUSION AND RECOMMENDATION

This study concludes the following:

1. The study created a model that will classify the toxicity using Bi-directonal LSTM Sequential model.

2. Model with pre-trained word embedding still outperformed the model created by this study that uses word embedding from KERAS.

3. This study can be used by the developers of the game to help in deciding the weight of penalties for the gamers since it resulted with high accuracy.

- The summary result presented that the researchers found Bi-directional LSTM model has high accuracy obtaining 86%.
- The loss of the model 0.0162 which is considered a very low chance of error in the prediction.
- The model also has very high precision and recall. The model got 97% of precision while 96% recall.

- The researchers recommend using a higher-spec GPU (preferably NVIDIA RTX 3000 Series since it has 3rd generation tensor cores).
- Future researchers should be careful in gathering data to prevent imbalance in the datasets.
- The researchers also recommend future researchers to find other pre-trained word embedding that can help in improving the accuracy of the study.

Conclusion of the presentation