CYBERBULLYING CLASSIFICATION

CS6120 GROUP 10 REPORT

Sindhya Balasubramanian
Khoury College of Computer Sciences
Northeastern University
Boston, MA 02115
balasubramanian.si@northeastern.edu

Eileen Chang
Khoury College of Computer Sciences
Northeastern University
Boston, MA 02115
chang.ei@northeastern.edu

Pushyanth Damarapati
Khoury College of Computer Sciences
Northeastern University
Boston, MA 02115
damarapati.p@northeastern.edu

Priyanka Padinam
Khoury College of Computer Sciences
Northeastern University
Boston, MA 02115
padinam.p@northeastern.edu

December 11, 2022

ABSTRACT

We implemented sentiment classification models for two separate tasks: binary and multi-class text classification of tweets posted to Twitter. The models were evaluated on the Cyberbullying Classification dataset [1], which labels tweets according to 6 classes of cyberbullying. In binary text classification we label tweets as either "cyberbullying" or "not cyberbullying," while in multi-class text classification we label tweets as a specific subcategory of cyberbullying. The same four models–Naive Bayes, ANN, CNN, and RNN–were used for both classification tasks in order to draw comparisons. Binary classification performed highest with the ANN model with 0.91 accuracy, while multi-class classification performed highest with the CNN model with 0.87 accuracy. In future applications of our work, we aim to train the best performing models from our results with larger datasets or with similar datasets from other social media sites.

Keywords nlp · sentiment · classification

1 Introduction

The rise of social media and the recent couple of years of covid-19 lockdown has led to a concerning increase in cyberbullying cases. In 2020, UNICEF even issued a warning in response to the increased cyberbullying compounded by social distancing and increased screen-time. Those who participate in cyberbullying have the convenience of being able to hide anonymously behind a screen, but the targets of bullying are likely to develop mental-health issues that persist even after the bullying has ceased. Due to social media's ability to spread information quickly and anonymously, a single person can end up being targeted by a large number of people of various demographics on an extreme scale. We aim to create a model that will flag harmful tweets and, therefore, protect targets of cyberbullying.

1.1 Dataset

We will be using a kaggle dataset, Cyberbullying Classification [1], consisting of more than 47,000 tweets labeled according to 6 classes of cyberbullying: Age, Ethnicity, Gender, Religion, Other Type of Cyberbullying, and Not cyberbullying. The data has been balanced in order to contain 8000 of each class. Each row of the dataset will have a tweet and its class of cyberbullying. The dataset is meant to be used to create a multi-classification model to predict cyberbullying type, create a binary classification model to flag potentially harmful tweets, and examine words and patterns associated with each type of cyberbullying.

2 Method

We implemented two separate classification tasks with alterations to our original dataset for each task. Our first task is **Binary (Two-Class) Text Classification** and our second task is **Multi-Class Text Classification**. The goals for each subproblem are as follows:

- **Binary Text Classification:** Identify potential for cyberbullying based on a textual comment on a social networking website
- **Multi-Class Text Classification:** Identify key features of a cyberbullying comment and understand the demographic being targeted in each comment

2.1 Dataset and Preprocessing

Prior to preprocessing, we made alterations to the original dataset in order to better meet our two classification goals, and the changes to the dataset were different for binary classification and multi-class classification.

For completing **Binary (Two-Class) Text Classification**, we generated a new dataset from the original text corpus with the text labeled as either:

- **Cyberbullying** (umbrella class for the subcategories: Age, Ethnicity, Gender, Religion, and Other)
- **Not Cyberbullying**

For completing **Multi-Class Text Classification**, we generated a new dataset from the original text corpus with the text labeled as a subcategory of cyberbullying with the removal of the **Other Type of Cyberbullying** subcategory. We decided to remove this subcategory as we believed that there is not much imbalance between different cyberbullying types and it could cause confusion for the models when differentiating the other specified cyberbullying types:

- **Age**
- **Ethnicity**
- **Gender**
- **Religion**

Once we created the appropriate datasets, we preprocessed the data by removing punctuation, eliminating stop words, stemming, lemmatizing, and removing numbers and URL tags.

2.2 Clustering

Clustering is the task of grouping a set of objects in such a way that objects in the same group, called a cluster, are more similar in features to each other than to those in other groups. We processed all the text data into TFIDF vectors and ran a K-Means Algorithm on it. We tested the optimal number of clusters using the K-Means Elbow Method as depicted in Figure 1 1.

We then visualized the top keywords for each cluster, as seen in Figure 2 2, and found well-formed clusters for 4 distinct subcategories. In cluster 0 we found mostly gender related cyberbullying, in cluster 1 we found mostly age related cyberbullying, in cluster 2 we found mostly ethnicity based cyberbullyingm, and in cluster 3 we found mostly religion based cyberbullying.

2.3 Models

We trained the same four models–**Naïve Bayes, ANN, CNN, and RNN**–for both binary and multi-class text classification tasks in order to draw comparisons. Each model implementation is discussed in further detail in the following subsections.

2.3.1 Naïve Bayes

The Naïve Bayes classifier is a classification algorithm mainly used for binary and multi-class classification problems. It is a classifier that works on Bayes theorem in which membership probabilities are predicted for each class, such as the probability of data points belonging to a specific class. The class having maximum probability is considered to be the most suitable class. Naïve Bayes classifiers assume that all the features and variables are not related to each

other, the absence or existence of one of the variables do not impact the absence or existence of another variable. Naïve Bayes is widely used as it is simple to implement and yields good results when used for text classification tasks such as sentiment analysis, intent detection, topic labelling, etc. It is highly extensible with respect to both data and the number of predictor classes as well. In the paper, Zhang et al. [3], a spam email detector is developed using the Naïve Bayes algorithm, and we referenced this work for implementing our classification tasks. There are three models that are built upon Naïve Bayes algorithm but we're concerned with Multinomial Naïve Bayes as our cyberbullying data is multinomial distributed and we are trying to analyse the sentiment of the data as well.

2.3.2 ANN

An artificial neural network (ANN) is a type of computational model that is inspired by the way biological neurons work in the human brain. ANNs are composed of many interconnected processing nodes that are used to model complex relationships between input and output data. These networks are trained using large amounts of data and a learning algorithm, which enables them to make predictions or decisions based on new input data. ANNs are commonly used in a wide range of applications, such as image and speech recognition, natural language processing, and predictive modeling. For this project, we have converted the preprocessed(cleaned) text into TFIDF vectors. Then for classifying whether a given tweet is bullying or not we have developed an Artificial Neural Network.

For **binary class classification**, we started with the less dense network with four layers (one input and output and two hidden layers) which yielded poor results. Later we increased the depth of the network. We observed that additional layers result in the model overfitting to the training data. In order to avoid overfitting, we used dropout layers at a rate of 0.5 which randomly sets input units to 0. We also experimented with different activation functions like Swish, Relu, Lekyrelu, tanh, etc., and observed that the "tanh" activation function performed well for this task. Our final ANN consists of seven layers (See Figure 3. The final classification layer has a sigmoid activation function. The model was trained for 30 epochs with a loss function of Binary CrossEntropy, RMSProp as Optimizer, and the total number of trainable parameters was 614,721.

For **multi-class classification**, we converted the targets into one hot encoding format. We also followed a similar network architecture as mentioned above, with seven layers (See Figure 4), and the final layer activation function was changed to softmax, which returned the probability of each class. The model was trained for 30 epochs with a loss function of Categorical CrossEntropy, ADAM as Optimizer with an initial learning rate of 1e-03. The total number of trainable parameters was 614,806.

2.3.3 CNN

A Convolutional Neural Network or CNN is a type of Artificial Neural Network that has been used widely in image recognition and object detection under computer vision. It is a feed forward network which indicates that the information flowing is in a single direction only. The working of this model can be divided into two main layers that are linked together and furthermore linked to a fully connected network to improve power of prediction. The two main layers can be looked at as the Convolution layer and the Pooling layer, and each has its own specific use. The convolution layer herein, is a transformative layer – it takes one piece at a time of the larger multidimensional input vector, applies a transformation on this piece and then moves to the next set. This transformed vector is the input received by the Pooling layer from the Convolutional layer. The Pooling layer herein, uses this transformed set and finds a representative value for each piece transformed by the previous layer. This is a form of feature extraction performed by the CNN that has enhanced the process of pattern recognition across a multitude of input varieties including text, audio and images. We referenced the paper, Luan et al. [2], for our own CNN implementation below.

For **binary class classification**, our CNN model made use of the following layers to process the textual padded count vectorized content:

- Embedding layer as an input to the sequential model
- Convolutional (1D) layer
- Global Pooling layer
- A fully connected network of Dense and dropout layers as hidden layers
- An output layer containing binary classification with 'rmsprop' as a metric

The model was evaluated on the following training and validation criteria: AUC, Loss, Accuracy, Precision, and Recall.

For **multi-class classification**, our CNN model made use of the following layers to process the textual padded count vectorized content:

- Embedding layer as an input to the sequential model
- Convolutional (1D) layer
- Global Pooling layer
- A fully connected network of Dense layers as hidden layers
- An output layer containing categorical classification

The model was evaluated on the following training and validation criteria: AUC, Loss, and Accuracy.

2.3.4 RNN

Recurrent Neural Network or RNN is a type of Neural Network that has been used widely in sequential data processing. It is a feed forward network which indicates that the information flowing is in a single direction only. The working of this model can be looked at in a single layer linked to a fully connected network to improve power of prediction. The layers looked at in this project are simple RNN which was introduced in the 1980s originally and the Long Short Term Memory layer which advanced this model further. The RNN layers typically store prior inputs that are necessary additions to improve the power of prediction and are stored in the hidden layer. In our project, we used the many to one and many to many architecture taking multiple inputs in the form of text counts padded sequences to produce a binary and a categorical classification model.

For **binary class classification** our RNN model made use of the following layers to process the textual padded count vectorized content:

- Embedding layer as an input to the sequential model
- Simple RNN layer
- A fully connected network of Dense and dropout layers as hidden layers
- An output layer containing binary classification with 'rmsprop' as a metric

The model was evaluated on the following training and validation criteria: AUC, Loss, Accuracy, Precision, and Recall.

For **mult-class classification** our RNN model made use of the following layers to process the textual padded count vectorized content:

- Embedding layer as an input to the sequential model
- LSTM Layer
- A fully connected network of Dense layers as hidden layers
- An output layer containing categorical classification

The model was evaluated on the following training and validation criteria: AUC, Loss, and Accuracy.

3 Results

We evaluated all four models on either all or a subset of the following training and validation criteria: **AUC, Loss, Accuracy, Recall, and Precision**. Our results implicated that ANN and CNN were our best performing models, and we generated plots of these results. These plots correspond to metrics computed at the completion of each epoch and, therefore, are functions of epochs. These plots are included in our Appendix and referenced below:

- ANN binary class classification: Figure 5
- ANN multi-class classification: Figure 6
- CNN binary class classification: Figure 7
- CNN multi-class classification: Figure 8

For further analysis of our results, we compiled the values we obtained from all of our model evaluations for binary class classification and multi-class classification into tables. The binary class classification data revealed that the best performing models in order are: ANN, Naive Bayes, CNN, and RNN. The multi-class classification data revealed that the best performing models in order are: CNN, Naive Bayes, ANN, and RNN. Please refer to Table 1 for the binary class classification data and Table 2 for the multi-class classification data below:

3.1 Model Training and Validation Data

Binary Class Classification:

Model	val_auc_1	val_binary_accuracy	val_recall	val_precision
Naive Bayes	0.89	0.83	0.79	0.81
ANN	0.90	0.87	0.93	0.91
CNN	0.60	0.58	0.36	0.75
RNN	0.49	0.49	1.00	0.51

Table 1: Binary Class Classification Results

Multi-Class Classification:

Model	val_auc_1	val_binary_accuracy	val_recall	val_precision
Naive Bayes	0.98	0.85	0.84	0.86
ANN	0.98	0.83	0.75	0.88
CNN	0.99	0.91	-	-
RNN	0.50	0.20	-	-

Table 2: Multi-Class Classification Results

4 Evaluation

Our results show that the ANN model is the best performing model for binary class classification with an 87% accuracy. The best performing model for multi-class classification is the CNN model with a 91% accuracy. We found it interesting that although the CNN model performed optimally with multi-class classification, it performed relatively poorly with binary class classification, with an accuracy of 58%. CNN models are prone to overfitting, which happens when models fit too well to the training set and make it difficult for the models to generalize when introduced to new examples that did not exist in the training set. A potential cause, therefore, of the low CNN accuracy with binary classification could be overfitting. The accuracy of the Naive Bayes and ANN models remained relatively consistent for both tasks, and the values ranged from 83-87% accuracy.

The RNN model was our lowest performing model for both of our tasks, with 49% for binary class classification and 20% for multi-class classification. It should be noted that we decided to use the Simple RNN layer for binary class classification and the LSTM layer for multi-class classification, which implies that the Simple RNN layer performed slightly better than the LSTM layer. We referenced Sandag et al. [4], which completed sentiment analysis of COVID-19 vaccine tweets using RNN and reported high performance with LSTM with a slightly loower performance with Simple RNN layers. According to the paper, LSTM is capable of long-term memorization and recall, especially on very large data, so we expected similar results with our implementation. A potential cause of our RNN results, may be the occurrence of vanishing gradients, which are noted to be one of the common drawbacks of RNN models.

5 Future Works

In future iterations of our work, we may first attempt to assess the issues, such as overfitting and vanishing gradients, in our lowest performing models, particularly RNN. In terms of further experimentation, we may try to evaluate our models on other available datasets from either twitter or other social media apps, such as instagram, facebook, and tiktok. Another potential experiment is running a multi-class classification task on datasets for various social media apps with our best model, then evaluating which social media site has the highest rates of cyberbullying within the subcategories. There are also additional models we can train on our data, such as Random Forest and KNN. Lastly, we intended to include a Graph Convolutional Network or GCN model implementation in our work while referencing Wang et al. [5], but this requires DQE data created separately which we did not have access to.

6 Github Repository

The code and documentation for this project can be found on our github repository:

https://github.com/eilccn/twitter-sentiment-classification

References

[1] Cyberbullying Classification. Retrieved October 2022, from https://www.kaggle.com/datasets/andrewmvd/cyberbullying-classification

[2] Yuandong Luan and Shaofu Lin Research on Text Classification Based on CNN and LSTM In *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), 2019*, pages 352-355. *doi: 10.1109/ICAICA.2019.8873454*.

[3] Haiyi Zhang and Di Li Naïve Bayes Text Classifier In *IEEE International Conference on Granular Computing (GRC), 2007*, pages 708-708. *doi: 10.1109/GrC.2007.40.*.

[4] Green Arther Sandag, Adinda Marcellina Manueke, and Michael Walean Sentiment Analysis of COVID-19 Vaccine Tweets in Indonesia Using Recurrent Neural Network (RNN) Approach In *3rd International Conference on Cybernetics and Intelligent System (ICORIS), 2021*, pages 1-7. *doi: 10.1109/ICORIS52787.2021.9649648.*.

[5] Jason Wang, Kaiqun Fu, and Chang-Tien Lu. SOSNet: A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection In *IEEE International Conference on Big Data (Big Data), 2020*, pages 1699-1708. *doi: 10.1109/BigData50022.2020.9378065*.

A Appendix

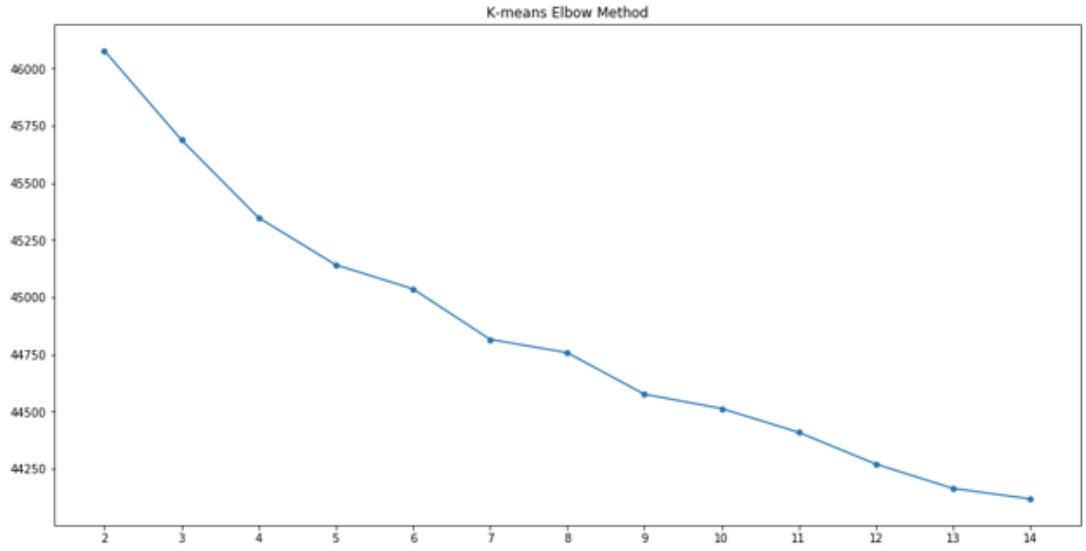

Figure 1: K-Means Elbow Method

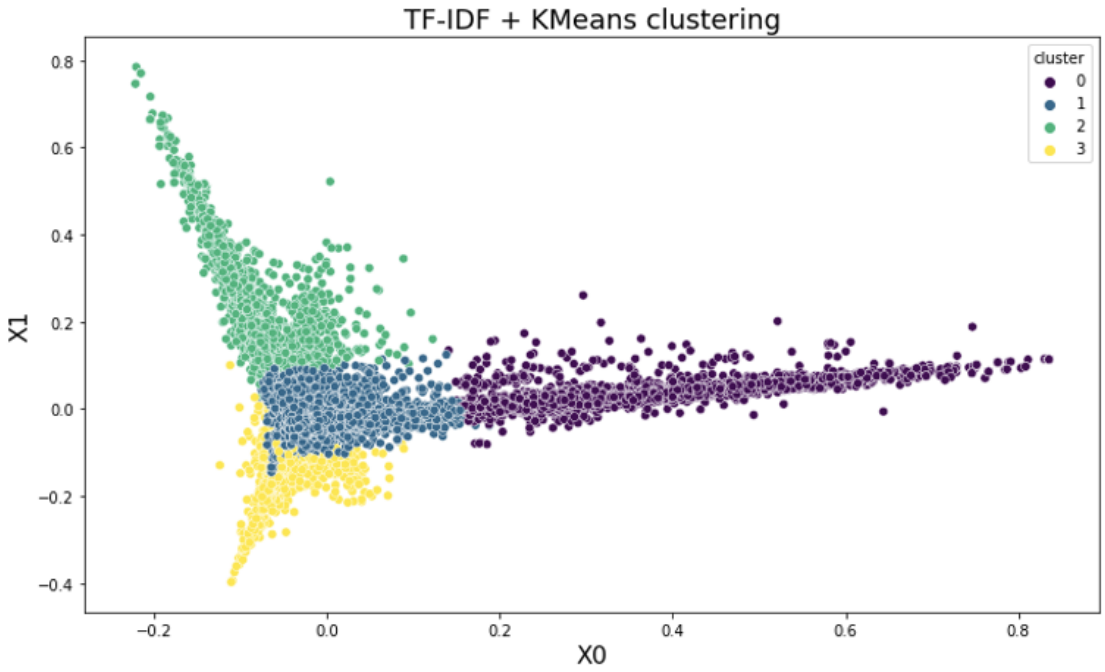

Figure 2: TF-IDF and K-Means Clustering

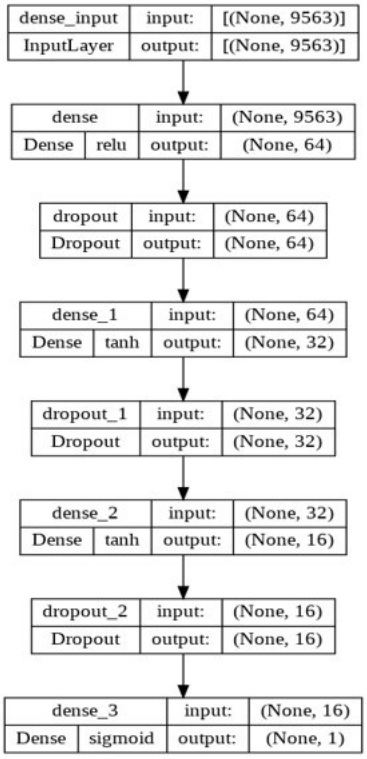

Figure 3: ANN layers for binary class classification

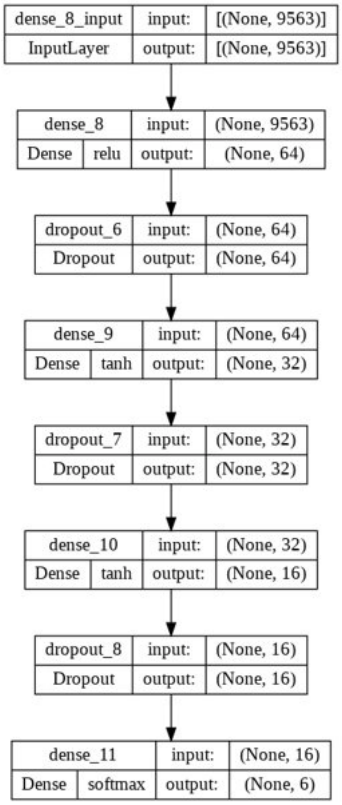

Figure 4: ANN layers for multi-class classification

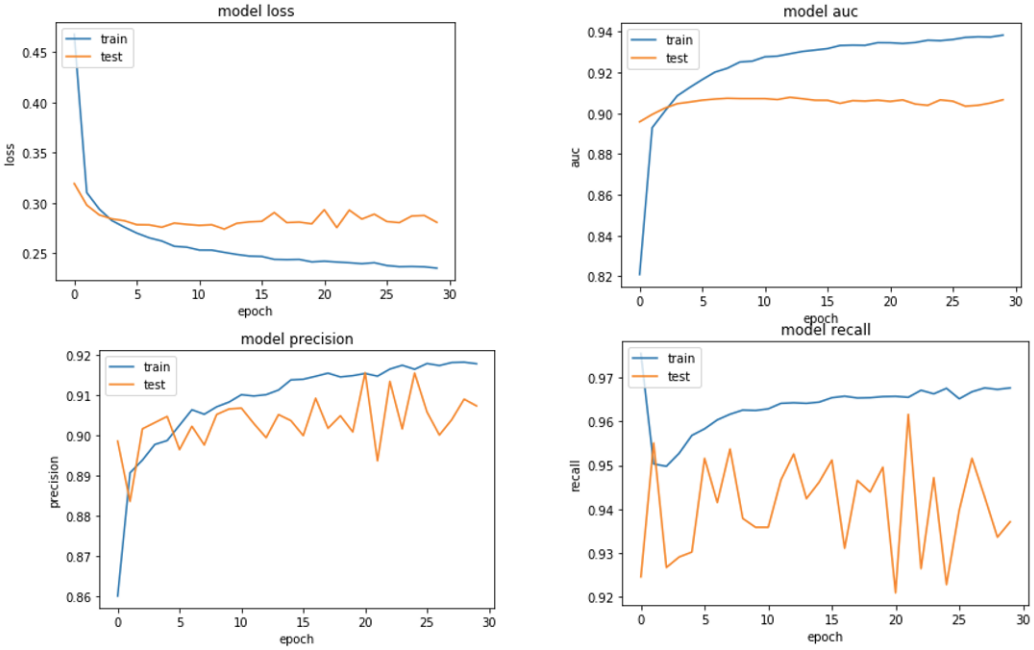

Figure 5: ANN training and validation plots for binary class classification

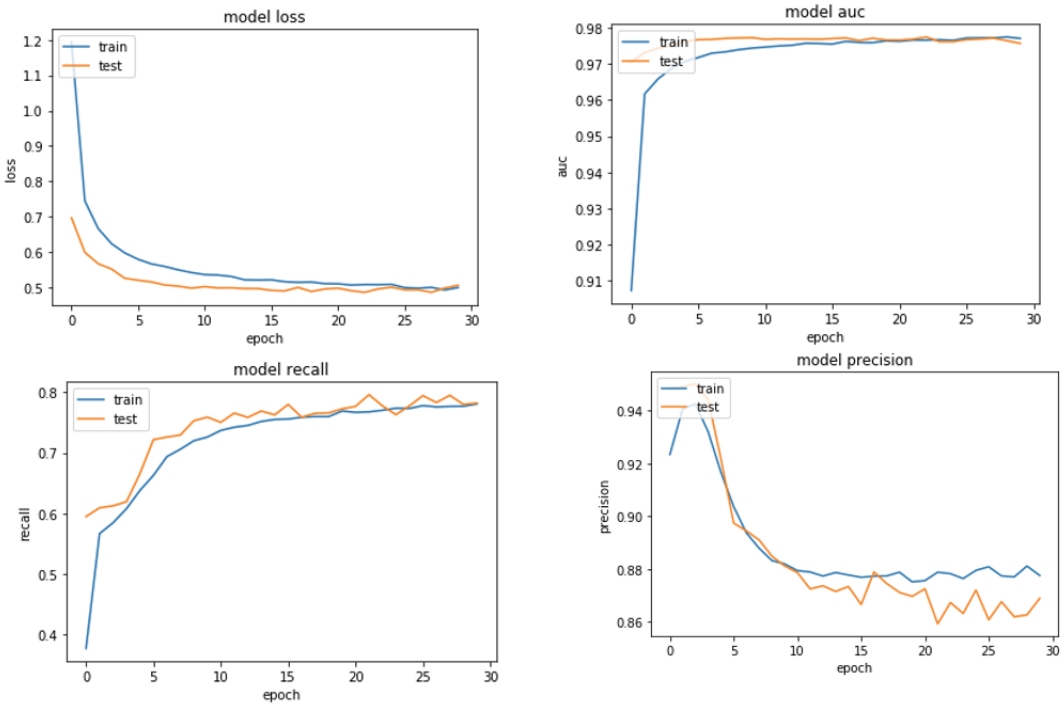

Figure 6: ANN training and validation plots for multi-class classification

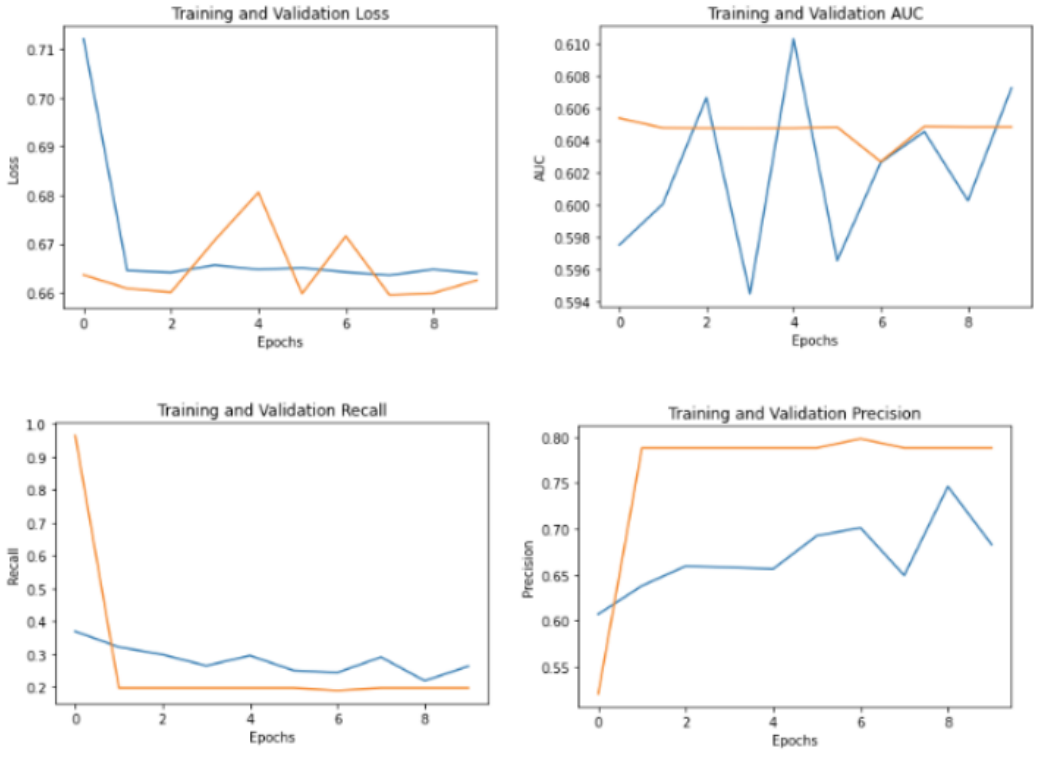

Figure 7: CNN training and validation plots for binary class classification

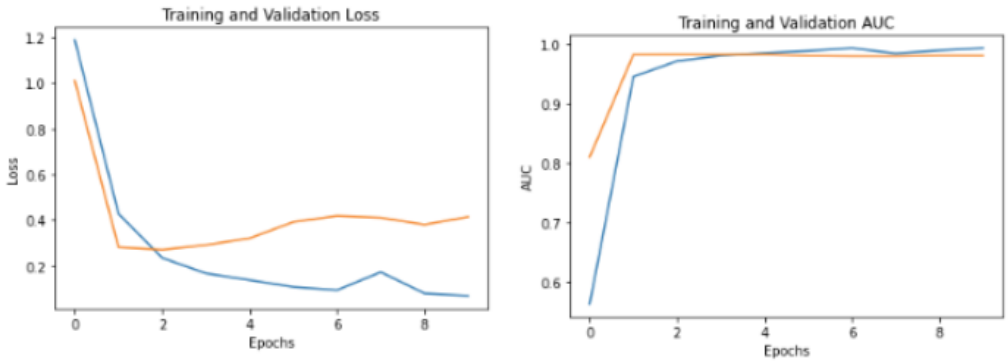

Figure 8: CNN training and validation plots for multi-class classification