Deep Learning based Crime Detection and Resource Creation Approach From Bengali Voice Calls

Khalid Saifullah, Mohammad Masudul Alam, Prosenjit Majumder Joy, Jadir Ibna Hasan, Salekul Islam and Nahid Hossain [*]
Department of Computer Science and Engineering, United International University, Dhaka, Bangladesh
E-mail:

Abstract. Mobile phones have revolutionized our way of communication. Despite its numerous benefits, it has become a great utility for conducting crimes or making threats. Due to the large number of users it is almost impossible for security forces to take proactive measures against those crimes. In this paper, with the help of machine learning, we focus on building a system that can detect potential threats in phone calls. We develop (to the best of our knowledge) the very first Bengali voice call dataset to train the machine learning system. Our system takes a voice call and uses a Deep 1D Convolutional Neural Network to analyze the call and a Multi-Layer Perceptron to decide whether any threats exist or not. The proposed simple baseline solution, trained on our ∼9hrs. worth voice call dataset, is able to achieve 91% precision, recall and F1-score in detecting the crime calls. We believe, in future these systems will aid in assisting in investigations, evaluating voice conversations, and giving predictions and estimations for potential threats. All of our recorded calls are freely available to use by the future researchers at: *https://tinyurl.com/detecThreats*

Keywords: Bengali, threat detection, crime detection, voice-call analysis, audio analysis, deep learning

1. Introduction

Mobile phone is considered as one of the greatest inventions of all time. It revolutionizes our way of communication. It has been one of the main medium of communication for decades. On the other hand, with the advancement in networking and telecommunication technologies, we have seen the birth of many social media and instant messaging applications. Thus, new technologies like instant text messages, voice messages, audio calls, and video calls have made communication easier. As a result criminals and wrongdoers can easily exploit these diversified communication media and they can take the advantages of these communication media to plot crimes or give threats to innocent people through voice calls. One way to prevent crimes like these is to build a system that detects potential crimes and threats from voice calls and flags them accordingly.

Ifaz et al. [1] have shown that one can perform early threat detection by training a machine learning model on speech data. However, the datasets they have worked on, such as RAVDESS [2]

[*] Corresponding author. E-mail: nahid@cse.uiu.ac.bd.

and TESS [3] are for English. Although it is a very important topic in today's world, no such work is found Bengali language even though it is spoken by more than 250 million people worldwide[4]. Moreover, in the literature there no voice call dataset is available for Bengali speaking people. This motivates us to work on this important topic with the guidance from an expert of the Crime Investigation Department (CID), Bangladesh.

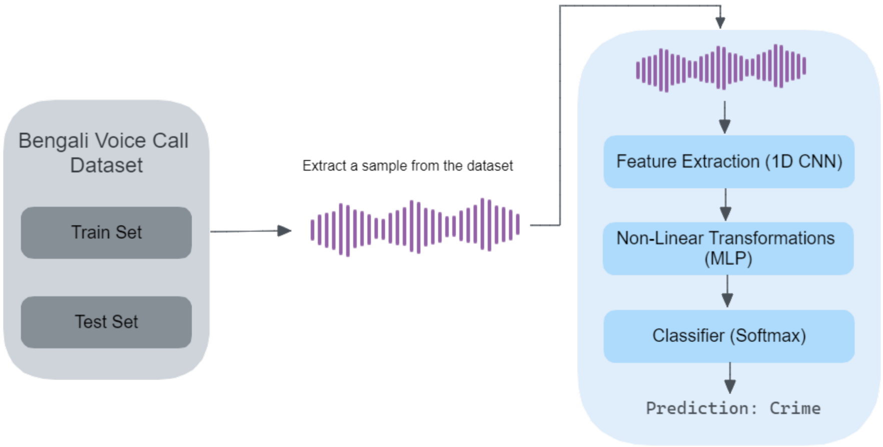
Bengali Voice Call Dataset
Train Set
Test Set
Extract a sample from the dataset
Feature Extraction (1D CNN)
Non-Linear Transformations (MLP)
Classifier (Softmax)
Prediction: Crime

Figure 1. High-level overview of our model

To develop any successful machine learning model, a comprehensive dataset is the first and foremost necessary requirement. Since there is no dataset for Bengali voice calls that targets to detect criminal and threat activities, we focus on building the very first Bengali voice call dataset to classify crime or threats. Then, we have trained a simple model on our own dataset as a baseline to detect possible criminal or threat activities. Once detected, security forces can be informed about the crimes before happening.

Making threatening calls to someone with the intention of real-life killing, looting, or any other atrocious activities are considered as `crime` samples. However, making calls to someone known (especially to friends) in a sarcastic manner with no intention of harmful activities is categorized as `sarcastic` samples. In a sarcastic call, a caller uses almost the same threatening words, however, in a sarcastic manner to a friend, relative or a known person. Das et al. [5] recently proposed a speech emotion recognition dataset for the Bengali language. Although it includes five rudimentary human emotional states, it does not include the sarcastic emotion, which is also very important in situations such as this proposed system. The proposed system classifies a call using both audio emotion and triggering words. Triggering words in calls are words that initiate suspecting a call belongs to either in a crime class or in sarcastic class. Some triggering words are খুন (kill), মৃত্যু (death), বোমা (bomb), মারামারি (flight), কিডন্যাপ (kidnap), চুরি (theta), and ডাকাতি (robbery). On the other hand, the calls which have no triggering or abusive words in threatening or sarcastic voices are categorized as normal calls.

Figure 1 shows our complete pipeline in a simplified manner. It monitors and analyzes voice calls at its raw form and finally predicts if the voice call contains any threat or not. Firstly a sample gets picked from our devised voice call dataset, then it goes through the inference pipeline in the following way: The 1D Convolutional Neural Network (CNN) [6] extracts the feature from the data sample, then that is fed into the Multilayer Perceptron (MLP) [7] which performs several non-linear transformations. Later, the softmax layer is used to detect the emotion in the voice sample (Crime, Normal, or Sarcastic).

The significant contributions of our work are summarized in the following:

- Developing the very first Bengali voice call dataset for classification with annotated labels.
- Developing a baseline model to detect crime or threats from Bengali voice calls.

The rest of the paper is organised as follows: section 2 describes the related works and identifies the limitations of previous approaches. The step-by-step data collection process, architecture and implementation of the proposed system have been briefly explained and described with tables in section 3. Section 4 demonstrates the experimental results and the analysis of the results as well. Finally, section 5 concludes the paper by mentioning future work.

2. Related work

The very first issue one has to deal with while working with audio data is data scarcity. A great deal of research has been done to develop numerous techniques that solve various audio data-related tasks such as speech recognition, etc. however, the amount of publicly available labelled data is still very limited.

There have been some recent works that show one can learn much richer and contextual representations from audio data [8, 9]. Another group of works has demonstrated the opportunities of Transfer Learning and Self-Supervised Pretraining in speech domain [10–12]. These methods are highly sample-efficient and capable of learning in small-data regime. The framework proposed by Schneider et al. [10] is a self-supervised approach that promises a way out of this data scarcity issue. The Internet is full of audio data, movies, songs, talk shows, etc. There is no limitation of that and everyday more audio visual data are being added. Therefore, with proper scrutiny we can use those data. This is the question the authors ask in their work [10]. They use a great amount of unlabelled data from the Internet to learn a meaningful representation. This phase is called 'pretraining'. After the model has learned the semantics of audio data it can represent them in a meaningful way. Those representations then can be used to solve any sort of downstream task. The authors have used an encoder (CNN) network and a context network (CNN) to learn the representation from the raw audio data. This end-to-end approach makes it a very likeable option for both companies. Thus, it can be said that more work will be seen based on this [10] approach.

In other related work, Ifaz et al. [1] have also attempted to detect threats from voice calls. They first did the conversion from speech to text. Then analyzing this text they have generated some threat words level. Secondly, they classified the emotion of the speech by generating some features directly from the audio signals and applying SVM and Naive Bayes algorithms. After that, they have compared their generated threat words level with their pre-determined threshold. Lastly, they have combined this result with the emotion of the speech to conclude with the decision if threat is present or not. Since recognizing the emotion from a conversation is the most vital part for correctly predicting the threat and detecting threat words are more or less trivial, we have to be very accurate in recognizing the emotion step. Basu et al. [13] has used a technique for emotion recognition by using CNN and RNN architecture and they have obtained up to 80% accuracy. They have extracted Mel Frequency Cepstral Coefficient (MFCC) features and passed these to the CNN network. Then they have used the output of the CNN as the input of LSTM to build their model.Abdulaziz et al. [14] have followed a simple approach where first they have generated text from the audio files using Google Speech-to-Text API. Then n-grams are generated from the text files using TensorFlow and the LSTM model. They have compared individual tokens they have got from n-grams with their lexicon of threat words that contains the threat words. If the token matches, then they have flagged the sentence as a threat. Lim et al. [15] have proposed a model based on both CNN and RNN models. They call it Time Distributed CNN. The main goal is not to use any predefined features but to create a model that learns independently. As a result, there is no need for semantic analysis. Here the authors have provided input to CNN as a time-frequency, 2D representation of audio using Short-time Fourier transform (STFT). With the combined time distributed deep hierarchical CNNs with the LSTM network layer, they have achieved a much better result.

By studying the existing works, we can observe that in the past several works have been carried out to improve the performance of different machine learning models. However, no work has targeted to devise new dataset, especially in Bengali language. We would like to mention reiterate that current machine learning models are of no use without data. Thus, our contribution in building the very first Bengali voice call dataset will hopefully help the community to involve in more research in the Speech Emotion Recognition (SER) domain.

3. Proposed Method

3.1. Data Collection

In our dataset we have total 612 number of samples. Note that, in Machine Learning community it is very common and preferable to express an audio dataset size in terms of hours. Accordingly, our dataset have more than 9 hours of audio data. More than one-third of the voice calls in the 'crime' (positive) class are taken from public online platforms (e.g., leaked crime/threat calls from the news, social media, etc.). One limitation is that around 70% of the calls are in male voice although this is expected. The data samples are distributed into three classes:

- Crime: A voice call where the conversation is about plotting a crime or making threat to someone
- Normal: A usual voice call that is not related to crime or threats.
- Sarcastic: A voice call where crime or threat related words are being used (e.g., gun, kill, etc.) but not in a serious fashion (e.g., two friends discussing a movie plot).

Table 1 shows the class distribution of the dataset, which also shows that for the sarcastic class both the number of samples and the duration are relatively lower with compared to the other two classes. Thus, our dataset can be considered as an imbalanced dataset. Table 2 illustrates textual sample of each class. Each sample is the exact textual representation in Bengali for a random phone call from each class. All the call recordings are freely available in a public repository for future researchers[1].

[1] https://tinyurl.com/detecThreats

Table 1

Distribution of data into three classes

Class	No. of Sample	Hours
Crime	280	4.46
Normal	305	4.22
Sarcastic	26	0.48

Table 2

Textual sample of each class

Class	Textual sample of calls
Crime	-“হ্যালো স্যার, রাজশাহীতে দুই আগস্ট পরীক্ষা, সবাই ট্রেনের টিকেট খুজবে। ওই দিনের সব টিকেট যদি কালোবাজারে ছাড়েন, তাহলে এক একটা টিকেট চার পাঁচ গুন দামে বেচা যাবে।” - “সবগুলা টিকেট কালোবাজারে ছাড়া সম্ভব না।” - “স্যার সমস্যা নাই, আপনি সবগুলা টিকেট আমাদের দিয়ে দিবেন, আমরা প্রতি টিকেট দিগুন দামে কিনে নিব, তার অর্ধেক আপনাদের। আর সকালে তিন-চারটা টিকেট বিক্রি করে বলবেন সব টিকেট শেষ হয়ে গেছে।” “ঠিক আছে।” ("Hello sir, there is an exam on 2nd August in Rajshahi, everyone will be looking for train tickets. If all the tickets on that day are released in the black market, then each ticket will be sold for four to five times the price." - "We can’t sell all tickets in black market." - "Sir, no problem, you will give us all the tickets, we will buy each ticket for two times the price, half of it will be yours, and you should sell three or four tickets in the morning and should say that all the tickets are sold out." -“You are right”)
Normal	-“হ্যালো” - “হ্যাঁ হ্যালো, প্রান্ত কি খবর?” - “এইতো, কেমন আছেন ভাইয়া?” - “ভালো আছি, তোমার কি খবর?” - “এইতো দোকানে আছি” - “এখন কি ব্যস্ত না ফ্রি আছো?” - “না বলেন” - “ওইদিকে আমের ফলন কেমন হইছে?” - “ফলন ঠিক আছে, দাম একটু বেড়ে গেছে” - “ও তাহলেতো তোমরা যারা বাগান কিনেছো তাদের জন্যতো সুবিধাই হয়েছে না?” - “না দাম বেড়ে গেছেতো, যাদের বাগানে আম বেশী আছে শুধু তাদেরই সুবিধা হইসে” - “ও আচ্ছা, ঠিক আছে আজ রাখি, দেখা হবে” - “ওকে আল্লাহ হাফেজ।” (-“Hello” - “Yes, hello, what's up with you?” - "Nothing much, how are you brother?" - "I'm fine, how about you?" - "Nothing, I am in the shop right now" - "Are you busy now or free?" - "No, we can talk" - "How is the production of mango this year over there?" - "Not bad, the price has gone up a little" - "Ok then those of you who bought the garden have been benefited, right?" - "No, the price is increased, only those who have more mangoes in their garden will get the benefit" - "Oh okay, we'll see you" - "May god protect us.")
Sarcastic	-“হ্যালো” - “কোথায় তুই?” - “বাসায় কেনো?” - “আজকে না তোর আমাদের সাথে দেখা করার কথা ছিল?” - “সরি বন্ধু, বাসায় মেহমান আসছে তাই আজকে ওইদিকে যাওয়া হবে না” - “মানে কি? তুই সব সময় এমন করিস। আজকে তুই যদি না দেখা করস, তোর বাসায় গিয়ে তোরে খুন করে আসবো, আর মাইর একটাও নিচে পড়বে না।” - “সরি বন্ধু” - “রাখ তোর সরি”(- "Hello" - "Where are you?" - "at home, why?" - "You were supposed to meet us today." - "Sorry friend, guests are coming home so I won't go there today" - "What do you mean? You always do that. If you don't show up today, I'll go to your house and kill you, and I will beat you black and blue." - “Sorry friend” - “Keep your sorry to yourself.”)

3.1.1. Annotation Procedure

Due to the nature of our dataset, we had to follow a non-trivial way to create a data sample. Here are the steps through which a data sample is collected:

- At first, choose the data class (‘crime’, ‘sarcastic’, ‘normal’) for which the sample is to be made.
- Coordinate two team members and write a script for the phone call. For example, if it is a ‘crime’ class, then write how the conversation is going to be, who is going to make threats to whom, what kind of crime will it be about, etc.
- Finally, make the phone call and proceed the conversation as per the script. When the call is finished recording, save it in the database under the particular class name.

The phone calls collected from the public domains are annotated manually examining each call and also converted to .wav format as mentioned in section 3.1.2.

3.1.2. Data Processing

Unlike any other ML projects, we have not gone through lengthy preprocessing step as our training pipeline is end-to-end. However, there are some standard perprocessing steps that we have taken and those are the following:

- **Format conversion:** At the very beginning the conversion of .mp4 and .mp3 to .wav format is performed.
- **Dataset split:** We make a 80/10/10 split of our dataset for the later train-test-validation phases.
- **Resampling:** We have resampled all the audios with the least sample rate, 8 khz due to meet the memory constraints and faster training.
- **Stereo to mono conversion:** We reduced the channel dimension of the audio to one for memory constraints and complexity reduction.
- **Truncation:** All the audio that are longer than the max-audio length (i.e., 50 sec or 400,000 samples) are cut down to the max-audio length.
- **Padding:** The audios that are shorter than the max-audio length (i.e., 50 sec. or 400000 samples) are right padded to the max-audio length.

3.1.3. Limitations and Biases

Due to budget, time, resource, and other constraints, a number of biases limitations exist, which introduced few biases into the dataset. Here are a few that we could easily identify without performing any rigorous scientific study:

- **Gender bias:** Unfortunately, as mentioned earlier, the dataset is biased towards male gender (70% calls are in male voice), and less calls are from female gender in different classes.
- **Typical crime plots:** The crime data recordings are mostly based on typical plots one usually sees in the movies or real-life leaked voice calls. As a result, some subtle crime plots are missing in the dataset that in terms create survivorship bias.
- **Dialects:** Bengali has a diverse set of dialects or way of speaking. Our dataset is highly biased with dialects of ‘Dhaka’ and a little of ‘Kolkata’.
- **Limited participants:** All the synthetic voice calls in the ‘crime’ class are made by a team of ten members.
- **Imbalanced Data:** The number of voice calls of ’Sarcastic’ class is very low compared to other two classes making the dataset imbalanced though we have solved it with WeightedRandomSampler (also known as oversampling) and ClassWeight [16] methods.
- **Privacy Violation:** Since a large portion of our call recordings are collected from public domains, we could not gain proper permission for the most of the call recordings from the actual callers or victims to use their recordings due to unavailability of contact information. However, we took verbal permission from some callers depending on availability of their contact information in publication domain.

3.2. Network Architecture

The architecture that we are using is a modified version of the M11 network which was proposed by Wdai et al. [17].

Since we are looking for an architecture that works on raw audio signal and performs well on classification or detection tasks, we have selected the M11 model architecture. It is a classic Deep

Learning architecture for speech classification tasks, and also works as a perfect baseline for our use case. We are calling our architecture as a 'Modified M11' because with the original M11 we have added an additional MLP at the end to convert the classifier more flexible and to converge faster. Figure 2 illustrates the proposed modified M11 model architecture.

Table 3
Modified M11 Architecture

Modified M11 (1.8M)
Input: 400000x1 (audio signal)
[80/4, 64]
Maxpool: 4x1
[3, 64] × 2
Maxpool: 4x1
[3, 128] × 2
Maxpool: 4x1
[3, 256] × 3
Maxpool: 4x1
[3, 512] × 2
Global average pooling
MLP
Softmax

Table 3 shows the model architecture that is being used in the later experiments. We are using a modified version the original M11 [17], which is an 11-layered architecture with 1.8M parameters. Note that [80/4, 64] denotes the conv1D layer with a kernel size of 80, stride of 4 and 64 filters (stride of 1 is not being mentioned in the table, i.e. [3, 64]). Moreover, [...] $\times n$ denotes n stacked layers. All conv1D layers are followed by a BatchNorm layer but is being omitted in the table to keep it neat and clean. The MLP is a simple fully connected neural net with three hidden layers and dropouts [18] in between.

3.3. Implementation

We have implemented the modified M11 model which is showed in the table 3. The audio signal is preprocessed through resampling, padding, truncation, etc. steps as we have discussed in section 3.1.2. We use BatchNorm [19] right after every conv1D layers to help with distribution shift [20]. Adam optimizer [21] is used with weight decay [22] and a mini-batch size of 64 (the largest we could fit inside the GPU memory) to train the model and the training epoch is set to 80.

4. Experimental Results and Analysis

To execute our experiments we have used free Kaggle GPUs. To be specific, a Tesla P100 with 16 GB of RAM is used to conduct all the experiments, and Google Drive is used for data storage.

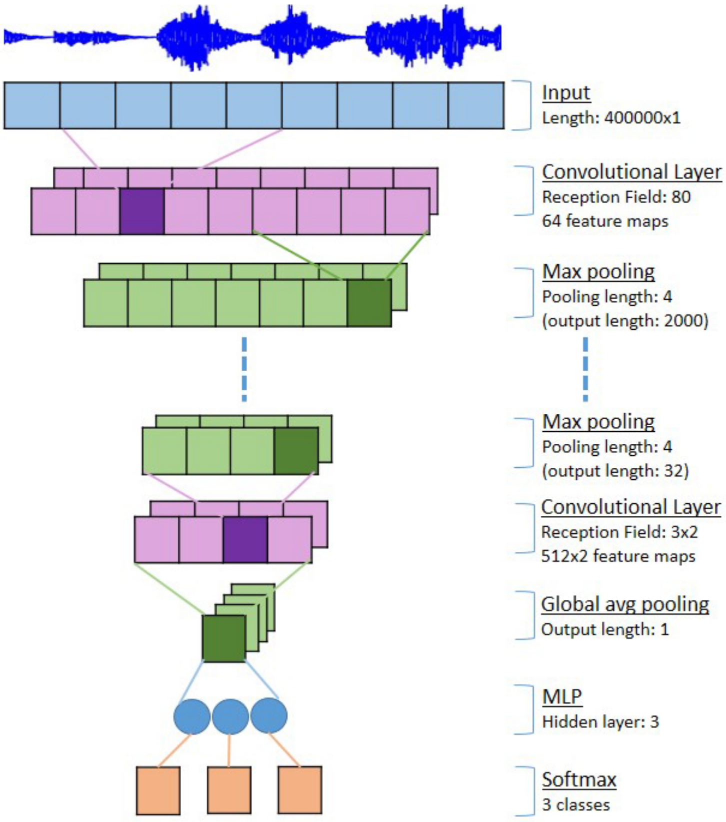

Figure 2. The modified M11 architecture

4.1. Selecting hyperparameter setting

Prior to training the model, we have a number of hyperparamets that we have to deal with, where ‘trial and error’ seems a tedious job. Therefore, we have decided to perform a random search [23] on the hyperparameter space of the model to narrow down the choices.

The sweep starts with a random set of hyperparameter parameters each time and we execute the sweep with 22 agents. Thus, 22 models are trained with different sets of hyperparameters, where each training is run for 30 epochs. Figure 3 shows a graphical summary of the sweep for all the different hyperparameters.

After performing the sweep and analysing all the results, we are able to select the best hyperparameter setting. Table 4 gives a clear picture on the hyperparameters we finally use to train the model.

Other than hyperparameters, we have also made some modifications to our training method. As our dataset is highly imbalanced, we have experimented with WeightedRandomSampler (also known as oversampling) and ClassWeight methods. These two methods are quite effective in that regime. WeightedRandomSampler method balances the data batch by sampling more minority class data, and ClassWeight method penalizes the model more when it makes mistake on minority class. We train the model in both WeightedRandomSampler setting and ClassWeight setting respectively.

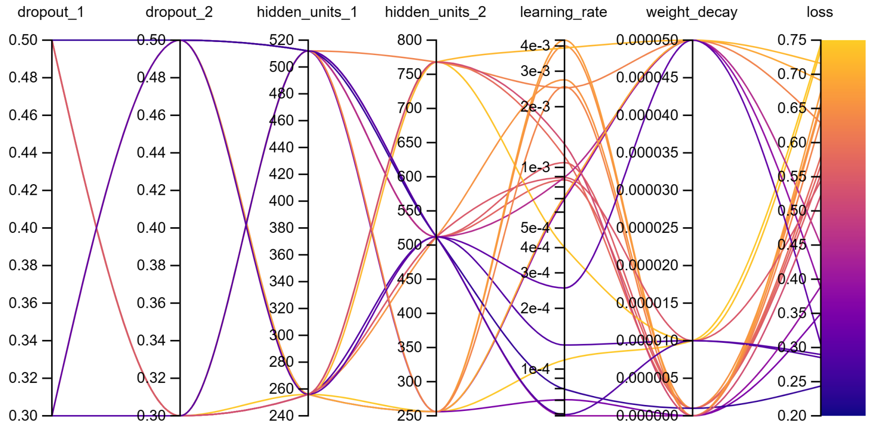

Figure 3. The graphical sweep summary

Table 4

Selected hyperparameters of the model

Hyperparameter	Value
dropout_1	0.3
dropout_2	0.5
weight_decay	0.00005
learning_rate	0.000696
hidden_units_1	256
hidden_units_1	256
optimizer	Adam
batch_size	64
epoch	80

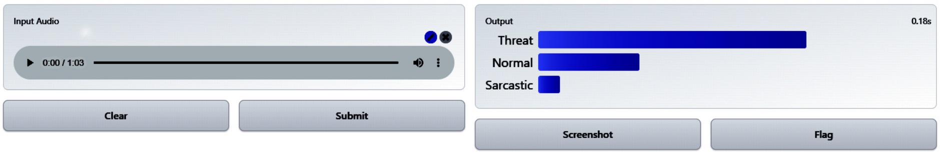

Figure 4. The deployed model

4.2. Results and Discussion

Figures 5–8 show the behavior of the model throughout the training period in both WeightedRandomSampler and ClassWeight settings. We can observe that two settings are very close in all of the metrics. However, one important observation is, even though the training loss is monotonically decreasing the validation metrics are fluctuating, which indicates the presence of overfitting. Therefore, we have used early stopping [24] to pick the best model from training iterations.

Finally, we have used the checkpoints of our trained model and test it on the held out data. Table 5 summarizes the performance of the two models. It is evident that there are some trade-

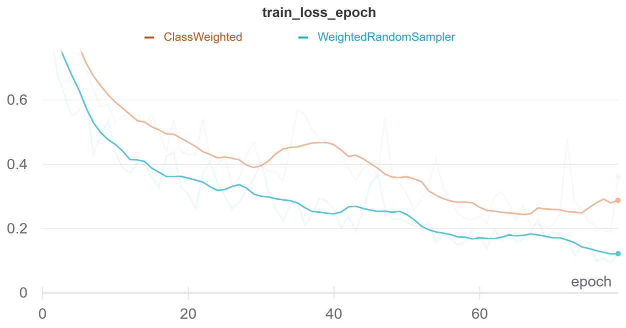

Figure 5. Loss history of the model

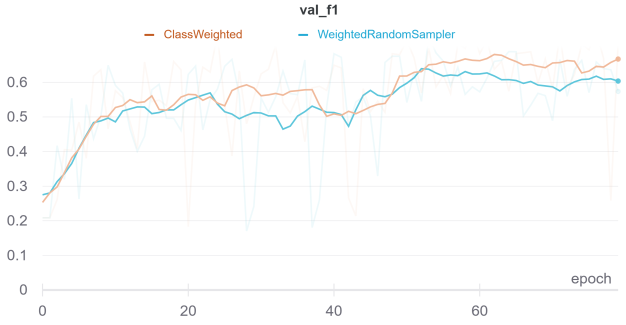

Figure 6. F1-score of the model during training

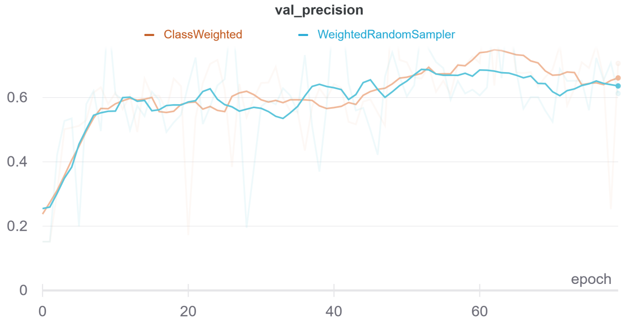

Figure 7. Precision score of the model during training

offs in both of the models: the WeightedRandomSampler setting has higher recall value but lower precision for the ‘crime’ class, whereas the ClassWeighted setting has a stable precision-recall. If higher recall is needed then the WeightedRandomSampler model is clearly a better choice.

Figures 9 and 10 show the confusion matrices for both ClassWeighted and WeightedRandomSampler settings, respectively. We can see that the latter setting is able to classify more number of positive class (i.e. Crime) samples than the former. Therefore, in this scenario we can be ended up using the F1-score to choose the final model. Based on the F1-score of all three classes, the ClassWeighted setting has better numbers, so that is our final model. We have deployed our trained model in a web server. Figure 4 shows that we can feed raw audio files into the system

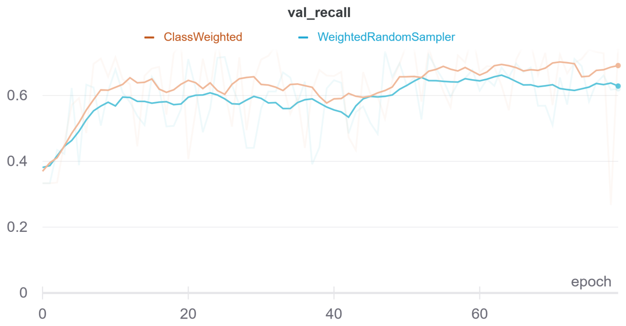

Figure 8. Recall score of the model during training

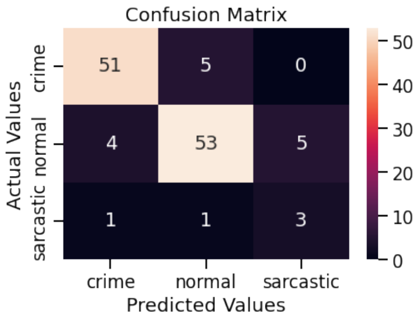

Figure 9. Confusion matrix of ClassWeighted setting

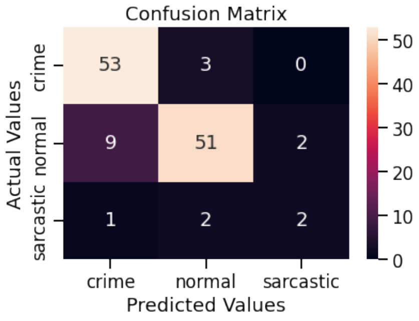

Figure 10. Confusion matrix of WeightedRandomSampler setting

and it can produce the normalized probabilities for each class.The deployed version is currently available for testing in the Hugging Face server[2] and can be accessed by any browser. The final improved version of the project will be hosted in a premium domain but the codes and the data will be always freely available.

[2] https://tinyurl.com/crimeFromCallApp

Table 5

Performance of the models

	Class	**Precision**	**Recall**	**F1-score**
Class Weighted Setting	Crime	0.91	0.91	0.91
	Normal	0.90	0.85	0.88
	Sarcastic	0.38	0.60	0.46
Weighted Random Sampler Setting	Crime	0.84	0.95	0.89
	Normal	0.91	0.82	0.86
	Sarcastic	0.50	0.40	0.44

5. Conclusion

As we live in the era of cellular phones, most of the crime or crime plots are made over cellphones these days. In this work our target is to detect those threat or crime related phone calls and to help the law enforcement agencies. We have collected and built a voice call dataset and trained a baseline 1D convolutional model. For the best hyperparameter setting we have used random search on the hyperparameter space, and due to the imbalance class distribution in the dataset we have explored both the ClassWeighted setting and WeightedRandomSampler setting to train the baseline model.

The main challenge we have faced in our work is to correctly classify the ‘sarcastic’ class. Identifying the sarcastic class is very confusing for the model, since most of our data are prepared by us or are taken from the Internet. These data are not prepared or taken from professionals but what we have seen is that our model is working very well in these data. From the experience of this work, in the future we are looking for coordinating with the law enforcement agencies and professional actors who can help us prepare a comprehensive dataset.

6. Acknowledgement

We would like to express our gratitude to Mr. Nesar Uddin Quadrey, Officer in Charge, Bangladesh IT Forensics, for sharing his real life experiences and providing his valuable suggestions.

References

[1] R.H.U. Ifaz Ishtiak Mazedur Rahman, Early Threat Warning Via Speech and Emotion Recognition from Voice Calls, 2018.

[2] S.R. Livingstone and F.A. Russo, The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), Zenodo, 2018, Funding Information Natural Sciences and Engineering Research Council of Canada: 2012-341583 Hear the world research chair in music and emotional speech from Phonak. doi:10.5281/zenodo.1188976.

[3] M.K. Pichora-Fuller and K. Dupuis, Toronto emotional speech set (TESS), Scholars Portal Dataverse, 2020. doi:10.5683/SP2/E8H2MF.

[4] N. Hossain, S. Islam and M.N. Huda, Development of Bangla Spell and Grammar Checkers: Resource Creation and Evaluation, *IEEE Access* **9** (2021), 141079–141097. doi:10.1109/ACCESS.2021.3119627.

[5] R.K. Das, N. Islam, M.R. Ahmed, S. Islam, S. Shatabda and A.K.M.M. Islam, BanglaSER: A speech emotion recognition dataset for the Bangla language, *Data in Brief* **42** (2022), 108091.

[6] H. Li, X. Yue, Z. Wang, W. Wang, H. Tomiyama and L. Meng, A survey of Convolutional Neural Networks —From software to hardware and the applications in measurement, *Measurement: Sensors* **18** (2021), 100080. doi:https://doi.org/10.1016/j.measen.2021.100080.

[7] H. Taud and J.F. Mas, *Multilayer Perceptron (MLP)*, in: *Geomatic Approaches for Modeling Land Change Scenarios*, M.T. Camacho Olmedo, M. Paegelow, J.-F. Mas and F. Escobar, eds, Springer International Publishing, 2018, pp. 451–455.

[8] A. van den Oord, O. Vinyals and K. Kavukcuoglu, Neural Discrete Representation Learning, in: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 6309–6318–. ISBN 9781510860964.

[9] B. van Niekerk, L. Nortje and H. Kamper, Vector-quantized neural networks for acoustic unit discovery in the ZeroSpeech 2020 challenge, in: *INTERSPEECH*, 2020.

[10] S. Schneider, A. Baevski, R. Collobert and M. Auli, wav2vec: Unsupervised Pre-Training for Speech Recognition, 2019, pp. 3465–3469. doi:10.21437/Interspeech.2019-1873.

[11] A. Baevski, Y. Zhou, A. Mohamed and M. Auli, wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, in: *Advances in Neural Information Processing Systems*, Vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan and H. Lin, eds, Curran Associates, Inc., 2020, pp. 12449–12460.

[12] W.-N. Hsu, B. Bolte, Y.-H.H. Tsai, K. Lakhotia, R. Salakhutdinov and A. Mohamed, HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **29** (2021), 3451–3460.

[13] S. Basu, J. Chakraborty and M. Aftabuddin, Emotion recognition from speech using convolutional neural network with recurrent neural network architecture, in: *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, 2017, pp. 333–336. doi:10.1109/CESYS.2017.8321292.

[14] A. Al-Tameem and A. Saudagar, Machine learning approach for identification of threat content in audio messages shared on social media, *Journal of Discrete Mathematical Sciences and Cryptography* **23** (2020), 83–93. doi:10.1080/09720529.2020.1721876.

[15] W. Lim, D. Jang and T. Lee, Speech emotion recognition using convolutional and Recurrent Neural Networks, in: *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2016, pp. 1–4. doi:10.1109/APSIPA.2016.7820699.

[16] J. Byrd and Z. Lipton, What is the Effect of Importance Weighting in Deep Learning?, in: *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, eds, Proceedings of Machine Learning Research, Vol. 97, PMLR, 2019, pp. 872–881.

[17] W. Dai, C. Dai, S. Qu, J. Li and S. Das, Very deep convolutional neural networks for raw waveforms, 2017, pp. 421–425. doi:10.1109/ICASSP.2017.7952190.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *J. Mach. Learn. Res.* **15**(1) (2014), 1929–1958–.

[19] S. Ioffe and C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, in: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, JMLR.org, 2015, pp. 448–456–.

[20] H. Shimodaira, Improving predictive inference under covariate shift by weighting the log-likelihood function, *Journal of Statistical Planning and Inference* **90** (2000), 227–244.

[21] D.P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, in: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds, 2015.

[22] A. Krogh and J. Hertz, A Simple Weight Decay Can Improve Generalization, in: *Advances in Neural Information Processing Systems*, Vol. 4, J. Moody, S. Hanson and R.P. Lippmann, eds, Morgan-Kaufmann, 1992.

[23] J. Bergstra and Y. Bengio, Random Search for Hyper-Parameter Optimization, *J. Mach. Learn. Res.* **13**(null) (2012), 281–305–.

[24] Y. Yao, L. Rosasco and A. Caponnetto, On Early Stopping in Gradient Descent Learning, *Constructive Approximation* **26** (2007), 289–315.