How to Cite:
Naidu, D. (2022). Voice analysis system for detection of vishing using deep learning. *International Journal of Health Sciences*, *6*(S1), 10457–10466. https://doi.org/10.53730/ijhs.v6nS1.7520

Voice analysis system for detection of vishing using deep learning

Devishree Naidu
CSE Department, Shri Ramdeobaba College of Engineering and Management, India.
Corresponding author email: Naidunaidud@rknec.edu

***Abstract*---**For many years, telecom fraud has caused significant financial harm to Indian telecommunications users. Traditional approaches for identifying telecom fraud frequently center on boycotting of faux phone numbers. Attackers, on the other hand, could merely escape discernment by modifying their phone numbers, which is fairly straightforward with VoIP (Voice over IP).To address this issue, this method detects telecommunication fraud supporting the substance of a spoken language rather than merely the caller's sign. This paper collects chronicles of telecommunication deceit, above all, from press sources and social platforms. To make datasets, our planned model utilizes machine learning techniques to look at knowledge and opt for high-quality descriptions from antecedently gathered knowledge. After that, Natural language processing is employed to draw out characteristics from the text- based data. Then, for extra telecommunication fraud detection, criteria to acknowledge identical material within the same call are formed. To spot telecommunication fraud online, the system provides an associate degree android application that will be loaded on a customer's smartphone. Once an associate degree incoming fraud decision is answered, the program will dynamically measure the call's contents to sight fraud. Our findings demonstrate that the system will effectively safeguard clients.

***Keywords*---**VoIP, fraud, fake phone numbers, telecom.

Introduction

India comes below the highest ten most spammed countries within the world. The scammers attempt to hook unsuspecting individuals' exploitation either phone calls or SMS and therefore the plan of action is usually the same: they'll attempt to get you to offer up sensitive data regarding your financials or force you to you reveal a secret OTP with the last word aim of extracting money

from your bank accounts or digital wallets. Most current techniques to sleuthing telecommunication fraud focused on tagging caller numbers that users discover as fraudulent. At a similar time, most researchers use machine learning approaches to find defrauding calls. They opt for options reckoning on criteria like contact numbers and call groups. They train models utilizing machine learning methods and utilize these models to track down fake calls, which may convey high location precision. However, as a result of a variety of modification software systems that are usually used, fraudsters utilize it to unanimously modify their phone number or disguise their number as the government institution. Due to these factors, traditional number-based detection procedures will be promptly circumvented.

Most current ways to detect telecommunication swindles are focused on marking caller numbers that are detected as deceptive by customers. We use real time voice detection using machine learning algorithms to detect whether the call is fraudulent or not.

Background and Related Work

Nowadays, additional and additional fraud criminals use amendment variety packages to perpetually amend their phone numbers in India. As a comparative measure, there are numerous cozeners who utilize number-commuting programming to mask their contacts as government office numbers, similar to World Health Organization, hospitals and police headquarters. Hence, antiquated element discovery upheld telephone numbers isn't any more extended solid and may basically be evaded by misrepresentation. Hence, this paper proposes a content-based telecommunication fraud detection method which relies on incoming caller numbers and associated information. Moreover it can identify deceitful calls with any telephone number and most of previous investigations are directed on the theory that the telecom organization may give more data, though our examination depends on the customers completely, where there is a requirement for the substance of the call to decide whether it is an underhanded call.

Fraud detection

The location of extortion has been the subject of several research and editorial writings as a result of the momentous harm to the overall public. Delamaire et al. (2009) proposed various sorts of Mastercard fakes, like liquidation extortion, burglary misrepresentation/fake misrepresentation, application extortion and social misrepresentation, examining the achievability of different methods to battle this kind of misrepresentation, for example, choice tree, hereditary calculations, bunching strategies and neural organizations. Rebahi et al. (2011) proposed the VoIP extortion and the misrepresentation discovery frameworks to it, really taking a look at their accessibility in VoIP conditions in different fields. There are two types of location frameworks: rule-based and solo strategies. LookmanSithic and Balasubramanian (2013) looked into the different types of misrepresentation in vehicle insurance systems and medical fields. Different types of information mining methods were utilized to recognize misrepresentation there as per the outcomes. The monetary misrepresentation identification has

turned into the most well known subject in the space of extortion discovery (Abdallah et al. 2016) which ordinarily prompts high monetary misfortunes.

Overview

In this segment, we'll give an overview of the media transmission extortion problem and our solution, as well as briefly explain our notion of telecom misrepresentation discovery. The reason for this article's focus on the location of media transmission extortion is that when clients receive bogus calls, they may be cautioned by warnings from the application on the Android platform. The entire cycle consists of three sections: the first being collection and preprocessing of extortion information from media broadcasts. The next step is to remove components and create identification rules. The final step is to run ready-to-extort programs for telecom extortion. The outline of our methodology is displayed in Fig. 1.

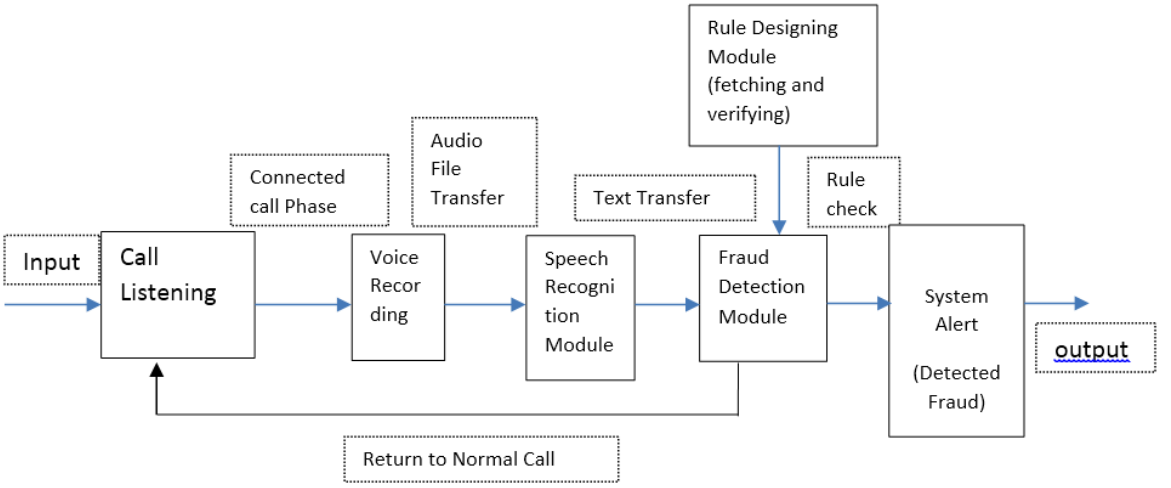

Figure 1: Telecommunication fraud detection mechanism for alert application

The collecting of telecoms fraud data is the initial step. The first step in analyzing the features and mechanisms of telecommunication fraud is to collect textual data. The purpose of the data collection is to collect texts connected to telecommunication fraud. The target data contains a case of fraudulent calls and the normal calls. We prepared our own data set. We used Microsoft Azure Speech Translation to convert Hindi speech to transcript. We then used Google translate code on Devanagari script to convert in equivalent English. To increase the dataset, we used data augmentation.

The feature extraction and rule-building process is the next step. It's crucial to extract features and construct criteria for detecting telecommunication fraud after the data acquired in the first step. Natural language processing technology is used. In this project, we considered to extract features such as keywords from fraudulent conversations. We also employ machine learning methods to demonstrate the suitability of the textual data we gathered and the authenticity

of the keywords we extracted. The research then constructs telecommunication fraud detection criteria based on the features retrieved from the text.

The deployment of telecommunication fraud detection is the final step. We created an Android-based telecommunication fraud alarm application in this paper. When a call is received on the user's phone, the application starts listening in on the incoming call. The application then converts the caller's voice into text using speech recognition technologies. The application then determines the call is fraudulent or not by using the detection rules created in the previous stage. If the application determines that the call is fake or spam, a warning message will appear on the user's smartphone screen, urging them to pay attention to the call.

Methodology

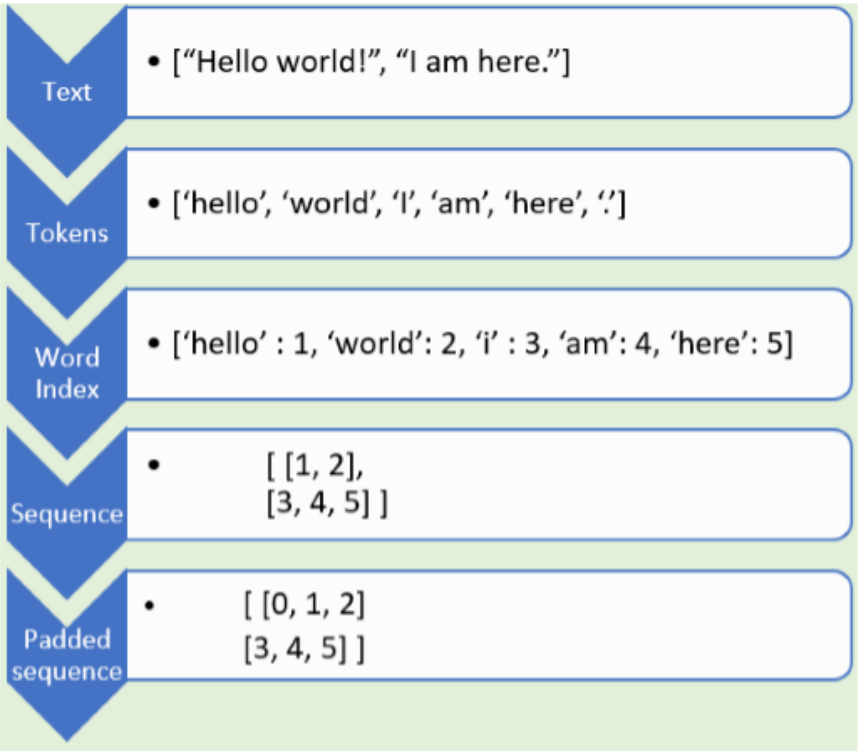

Figure 3: Data Pre-processing Flowchart

1. The text label is converted to numerical value and the data is divided into training and testing sets.
2. Then labels are converted to numpy arrays to fit deep learning models.
3. From the total data, 80% is used for training and 20% for testing purposes.

Tokenization

Deep learning models are unable to comprehend text. Thus, text is converted into numerical representation. For this purpose, the primaryway is to use

Tokenization. The Tokenizer API is used to splitsentences into words and encode these into integers. Tokenization also does the pre-processing by

1. Tokenizing into character or word
2. number of words for maximum number of distinct tokens
3. remove unnecessary punctuation terms
4. change entire words to lowercase
5. change entire words to integer

Hyper-parameters used in Tokenizer objects are: number_of_words and oov_Token.

1. number_of_words: It suggests a number of unique words that are to be loaded in training and testing data. In this paper we selected 500 words, (vocabulory_size)
2. oov_Token: An out of vocabulary token isappended to the word index in the corpusto construct the model. The reason is to replace out of vocabulary words i.ewords that are not in our corpus during texttosequence calls.

Sequencing and Padding

Once tokenization is done, each sentence is represented by a sequence of numbers which uses texts to sequences from the tokenizer object. Eventually, pad the sequence so that we can have the same length of each sequence. Sequencing and padding are performed for both training and testing data. Let's say before padding, the first sequence is 27 words long whereas the second one is 24. Once the padding is applied, both sequences have a length of 50.

Training the model

We train our datasets through different models to choose which models are giving the best results. For the purpose of this project we chose Dense Spam Detection Architecture, Long Short-Term Memory (LSTM) layer architecture, Bidirectional LSTM Spam detection architecture.

Dense Spam Detection Architecture:

1. This is a sequential model, which means that the layers are put up in a sequential order.
2. The embedding layer, which converts every word to a N-dimensional vector of real numbers, is the architecture's initial hidden layer.
   a. model.add (Embedding(vocab_size, embeding_dim, input_length=max_len))
   b. The amount of words you wish to tokenize or the maximum number of words you want to maintain is determined by vocab_size.
   c. The length of the vector that we give to the layer is the embedding dimension.
   d. The input_length specifies the size of the input layer to be passed.
3. The pooling layer is the following layer, which helps us reduce the size of the model's parameters. This allows the parameters to be more resistant to changes in their position. Average pooling and max pooling are two common pooling algorithms that summarizes a feature's average presence

and most activated presence, respectively. For our purposes, we employed Average pooling and thereby transformed the layer to a single dimension. The pooling layer aids in the model's overfitting avoidance.

4. A dense layer was the next layer we applied. It is a deep layer of a neural network in which each neuron in a dense layer receives input from all neurons in the preceding layer.
   We used the 'relu' activation function in this layer. The activation parameter is helpful in applying the element-wise activation function in a dense layer. The rectified linear activation function, or ReLU for short, is a piecewise linear function that, if the input is positive, outputs the input directly; else, it outputs zero. To avoid overfitting, we employed the dropout layer after that. The Dropout layer, which helps minimise overfitting, changes input units to 0 at random with a rate frequency at each step during training time.Inputs that aren't set to 0 are scaled up by 1/(1 - rate) so that the total sum remains the same.
5. The last layer was again a dense layer with activation function as 'sigmoid'.it is especially used for models where we have to predict the probability as an output. We simply utilised one output neuron because there are only two classifications to classify (fraud or not fraud). Probabilities between 0 and 1 are output by the sigmoid activation function.
6. After this we compile our model. We used 'Adam' as an optimizer. Adam optimization is a stochastic gradient descent approach that uses adaptive first-order and second-order moment estimation.

Long Short-Term Memory (LSTM) Model:

Long Short-Term Memory Network is an advanced RNN, a sequential network, that allows information to persist. It is equipped for dealing with the evaporating slope issue looked at by RNN. An intermittent impartial network is otherwise called RNN utilized for steady memory.
The spam detection model is fitted using LSTM as given below. The following are some new hyper-parameters utilised in LSTM:

1. SpatialDropout1D is used to drop out our embedding layer by using drop_embed =0.2. The SpatialDropout1D helps to drop entire 1D feature maps instead of individual elements.
2. n_lstmequals to 20 means the number of nodes in the LSTM cell's hidden layers
3. If you set return_sequences=True, the LSTM cell will output every unrolled LSTM cell's output over time. If this parameter is omitted, the LSTM cell will return the output of the preceding step's LSTM cell.

Epoch: The number of iterations of the learning algorithm on the whole training data set. The epochs was fixed to 30.

Bi-directional Long Short-Term Memory (BiLSTM) Model:

Bidirectional recurrent neural networks (RNN) are just two separate RNNs joined together. At each time step, this structure enables the networks to get both backward and forward feedback about the sequence. When you choose bidirectional, your inputs will be sent in two distinct directions, one from backward to forward and one from forward to backward and what contrasts this

methodology from unidirectional is that in the LSTM that runs in reverse, you protect data from the future and using the two hidden states together, you can save past information and future information at any moment in time.

In contrast to LSTM, the Bi-LSTM recognizes patterns from each token before and after it in a document. In time, the Bi-LSTM back-propagates both in forward and reverse directions. As a result, the computational time is longer than using LSTM. Bi-LSTM, on the other hand, is more accurate in the vast majority of circumstances.

BERT model

BERT stands for Bidirectional Encoder Representations from Transformers. By reciprocally creating on each left and right context, it is possible to pre-train deep bidirectional representations from unlabeled messages. Following that, the pre-trained BERT model is fine-tuned with one additional output layer to produce advanced models for a wide range of NLP tasks.

This model accepts a CLS token as the initial input, followed by a sequence of words. CLS symbolizes classification tokens. CLS then transmits the data to the layers above it. Each layer performs self-attention, then proceeds the result through a feedforward network prior to passing the encoder to the next. The model generates a hidden-size vector (768 for BERT BASE). We may use the output corresponding to the CLS token to generate a classifier from this model.

This trained vector can now be utilized for a variety of tasks, including classification, translation, and so on. In the classification job, for example, the study produces excellent repercussions simply by employing a single layer NN on the BERT model.

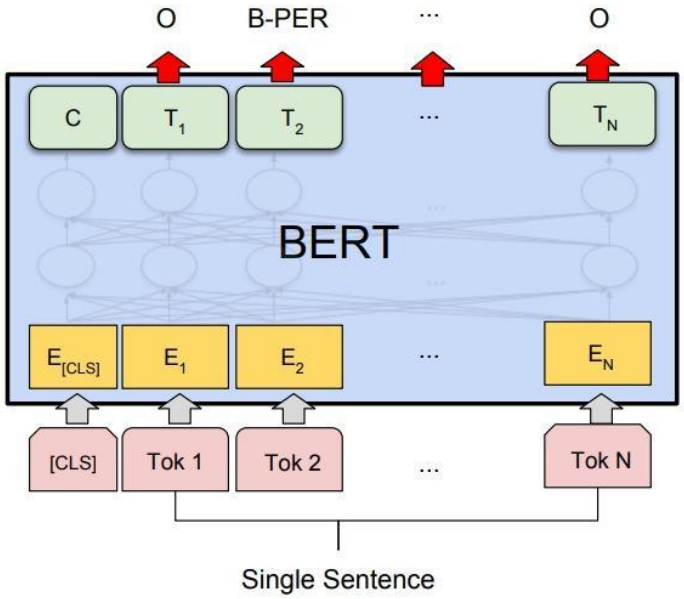

Figure 4: BERT Model Architecture

1. First, we installed Transformers Library then we uploaded the downloaded spam dataset to our Collab runtime.
2. Then, we import the BERT Model and BERT Tokenizer. Since the messages (text) in the dataset are of varying length, therefore we will use padding to make all the messages have the same length.
3. Next, we convert the integer sequences to tensors. Now we create data loaders for both the train and validation set.
4. So, till now we have defined the model architecture, we have specified the optimizer and the loss function, and our data loaders are also ready. Now we have to define a couple of functions to train (fine-tune) and evaluate the model, respectively.
5. Once the weights are loaded, we can use the fine-tuned model to make predictions on the test set.

Result

After implementing various models on our dataset, the following output was recorded. The accuracy and validation loss from LSTM are 89% and 0.33 respectively.

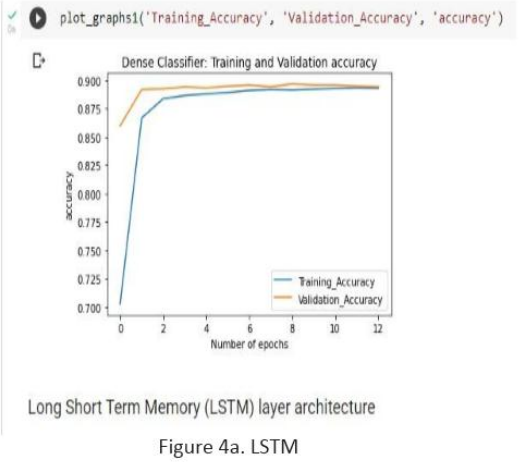

Figure 4a. LSTM

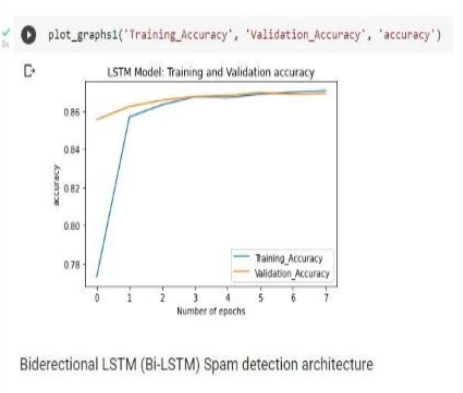

Figure 4b. Bi-LSTM

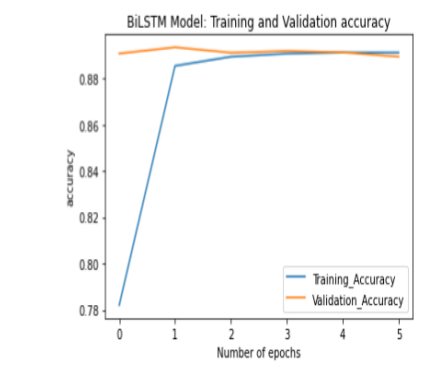

Figure 4c. Training of BiLSTM

The accuracy and validation loss from BiLSTM are 94% and 0.18, respectively. On the basis of accuracy we have selected the Bi-directional Long Short Term Memory (BiLSTM) Model. Telecommunication infidelity has cost telecommunication customers a lot of money in the past.

Table 1
Model Validation and Accuracy Testing

Model type	Validation Loss	Accuracy
BiLSTM	0.18	92%
LSTM	0.33	89%
BERT	0.16	94%

Conclusion and Future Work

Long-established methods to find telecommunication swindles sometimes give confidence in building a negative list of extortion phone numbers. Nonetheless, invaders will merely elude this type of discernment by modifying the numbers. To resolve the problem, our negotiated method gets telecommunication content fraud on the phone in lieu of using the user's contact. We gather descriptions of fraudulent telecommunications from news publications, social media, and other online sources in particular. Our suggested method examines data and selects high-quality definitions from previously collected data to construct data sets using machine learning methods. Then, to extract features from text data, we recommend using natural language processing. To detect fraudulent telecommunications, we establish rules for detecting the same material in the conversation. To obtain online finding of social media, developing an Android application that can be installed on a consumer's smartphone to stay vigilant. When a coming fraud call is attended, the app can then zestfully inspect the contents of the call for identification of guile. As tested with model accuracy this approach works effectively to detect fake calls and protect users of the implemented application.

References

1. Jabbar, M. A. and S. B. Suharjito. "Fraud Detection Call Detail Record Using Machine Learning in Telecommunications Company." Advances in Science, Technology and Engineering Systems Journal 5 (2020): 63-69.
2. J. Brownlee " A Gentle Introduction to Long Short-Term Memory Networks", July 7, 2021 in Long Short-Term Memory Networks.
3. E. Ma "Data Augmentation library for text", Towards Data Science, Apr 21, 2019.
4. J. Xing , M. Yu , S. Wang ,Y. Zhang , and Y. Ding - Automated Fraudulent Phone Call Recognition through Deep Learning, Wireless Communications and Mobile Computing Volume 2020 | Article ID 8853468.
5. Ting Sun.Miklos A. Vasarhelyi- Embracing Textual Data Analytics in Auditing with Deep Learning, The International Journal of Digital Accounting Research Vol. 18, 2018, pp. 49-67 ISSN: 2340-5058 Submitted December 2017 DOI: 10.4192/1577-8517-v18_3.

6. M. Swarnkar, N. Hubballi - SpamDetector: Detecting spam callers in Voice over Internet Protocol with graph anomalies, Security And Privacy, Volume2, Issue 1, 19 December 2018.
7. Zhao, Q., Chen, K., Li, T. et al. "Detecting telecommunication fraud by understanding the contents of a call". Cybersecur 1, 8 (2018).