Metadata of the chapter that will be visualized in SpringerLink

Book Title	Proceedings of the Future Technologies Conference (FTC) 2021, Volume 2	
Series Title		
Chapter Title	Spam Detection Over Call Transcript Using Deep Learning	
Copyright Year	2022	
Copyright HolderName	The Author(s), under exclusive license to Springer Nature Switzerland AG	
Corresponding Author	Family Name	**Natarajan**
	Particle	
	Given Name	**Abhiram**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	RV College of Engineering
	Address	Bangalore, India
	Email	abhiram.natarjan@gmail.com
Author	Family Name	**Kannan**
	Particle	
	Given Name	**Anirudh**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	RV College of Engineering
	Address	Bangalore, India
	Email	
Author	Family Name	**Belagali**
	Particle	
	Given Name	**Varun**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	RV College of Engineering
	Address	Bangalore, India
	Email	
Author	Family Name	**Pai**
	Particle	
	Given Name	**Vaibhavi N.**
	Prefix	
	Suffix	

	Role	
	Division	
	Organization	Samsung R&D Institute India
	Address	Bangalore, India
	Email	
Author	Family Name	**Shettar**
	Particle	
	Given Name	**Rajashree**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	RV College of Engineering
	Address	Bangalore, India
	Email	
Author	Family Name	**Ghuli**
	Particle	
	Given Name	**Poonam**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	RV College of Engineering
	Address	Bangalore, India
	Email	

Abstract	An increasing volume of malicious spam calls has ushered an urgent need for a system that can automatically detect them. A robust spam call detection system must identify and block spam calls with high accuracy. Unlike current techniques that use crowd sourcing and are prone to phone number spoofing, this paper proposes a novel deep learning-based approach to classify phone calls, as spam or ham, based on the generated call transcript. This paper proposes a means of classification using embedded features with a 2-Path Hybrid Model (2PHM), comprising of a Long Short Term Memory Network (LSTM) and a Sentence Similarity Network. The 2PHM uses the transcript generated from the call by converting it into feature vectors, embedded in a high dimensional latent space, using the Universal Sentence Encoder embedding technique. Both paths of the hybrid model generate results that are weighed in an 85:15 ratio to finally generate a classification. The proposed system has been tested against Machine Learning models like Naïve Bayes, K-Nearest Neighbors, Random Forests and Support Vector Machines and performs better than them, with an overall classification accuracy of 93% and a latency of 3 s.

Keywords (separated by '-')	Spam call detection - Deep learning - Long short term memory - Sentence similarity network - Latent space

Spam Detection Over Call Transcript Using Deep Learning

Abhiram Natarajan[1(✉)], Anirudh Kannan[1], Varun Belagali[1], Vaibhavi N. Pai[2], Rajashree Shettar[1], and Poonam Ghuli[1]

[1] RV College of Engineering, Bangalore, India
[2] Samsung R&D Institute India, Bangalore, India

Abstract. An increasing volume of malicious spam calls has ushered an urgent need for a system that can automatically detect them. A robust spam call detection system must identify and block spam calls with high accuracy. Unlike current techniques that use crowd sourcing and are prone to phone number spoofing, this paper proposes a novel deep learning-based approach to classify phone calls, as spam or ham, based on the generated call transcript. This paper proposes a means of classification using embedded features with a 2-Path Hybrid Model (2PHM), comprising of a Long Short Term Memory Network (LSTM) and a Sentence Similarity Network. The 2PHM uses the transcript generated from the call by converting it into feature vectors, embedded in a high dimensional latent space, using the Universal Sentence Encoder embedding technique. Both paths of the hybrid model generate results that are weighed in an 85:15 ratio to finally generate a classification. The proposed system has been tested against Machine Learning models like Naïve Bayes, K-Nearest Neighbors, Random Forests and Support Vector Machines and performs better than them, with an overall classification accuracy of 93% and a latency of 3 s.

Keywords: Spam call detection · Deep learning · Long short term memory · Sentence similarity network · Latent space

1 Introduction

Given the prominence of voice calls in communication, it is almost inevitable that users are frequently disturbed by spam calls made for promotional and fraudulent purposes. This includes (but is not limited to) calls about credit cards, loans, real estate, stock market etc. We consider spam calls as irrelevant and unexpected calls made to a group of recipients on a large scale. A call that lacks spam content is termed as being ham. Spam calls are accepted as a serious issue that not only causes mental agony to the recipient but could also potentially lead to financial scams and losses [1]. While inhabitants of developing countries such as India and Brazil experience a majority of these spam calls from human operators, the calls in developed countries like the United States and the United Kingdom are robotic calls that make use of automated voice call systems to target potential victims [2]. An effective spam call classifier must minimize false positives while maximizing positive spam detection.

K. Arai (Ed.): FTC 2021, LNNS 0000002, pp. 1–13, 2022.
https://doi.org/10.1007/978-3-030-89880-9_10

In order to save the user from any harassment and loss of time, a mechanism to automatically decline calls using their intent has been proposed. This involves a novel deep learning approach, the 2-Path Hybrid Model (2PHM), to automate classification of an incoming call as spam or ham based on the call transcript. The 2PHM has two parallelly executed models to minimize latency without compromising on accuracy. The two architectures are an LSTM network and a Sentence Similarity Model, both of which independently classify a call as either spam or ham. The outputs generated from both these models are weighed and averaged. The net result is additionally biased to classify as ham if either model generates a ham confidence greater than 0.65. This helps prevent the detection of false positives arising from any emergency calls.

The use of a call transcript over phone number based databases ensures that the intent of the caller is correctly identified reducing the possibility of misclassification. The call transcript can be also used in future works to generate actionable items like "Set a reminder" that can help enhance user experience.

2 Recent Work

Most prior work on spam detection has been focused on text-based email and instant messaging data [3, 4]. These techniques convert the text data into word embeddings [5], and cluster these into ham/spam classes using appropriate algorithms. We follow a similar approach in this paper, given that the text-based email and instant messaging datasets used in prior works are similar to our spam call dataset. The authors in [4] compare several machine learning techniques that deal with text-based email spam classification. A few of these models have been explored in this paper as well. These include Support Vector Machine, which fits a hyperplane to separate embedding features [5], K-NN, which classifies embeddings based on their nearest neighbors [6], Naïve Bayes, which is probabilistic classifier [7] and Random Forests, which use decision trees to perform classification [8]. These are further discussed in Sect. 6.

The traditional spam call detection pipelines primarily tackle Spam over Internet Telephony (SPIT) [9–11] i.e. bulk spam messages broadcast over VoIP. Such methods attempt to detect intent before a call is established [12]. A majority of these techniques perform statistical analysis on the logs of telephone call. They therefore provide limited evidence of practical applicability given the existing telephony infrastructure.

More recent work explored the use of machine learning techniques to improve efficiency, and statistical techniques used to analyze call traffic logs were replaced by neural networks. One technique discussed by the authors in [13] used a machine learning system to detect and classify malicious spam calls on the basis of trust values. This was innovative in that, the values assigned were based on telephone logs and user feedback. However, this technique did not detect the intent of the call making it impractical as phone numbers can be easily spoofed.

Following the intuitions gained from prior work dealing with email and instant messaging-based spam by the authors in [4], a novel deep learning model is proposed in this paper which classifies phone calls based on their intent. The use of hybrid architecture with dual weighted parallel models compensates any bias in the dataset, improving domain generalization to a great extent. The use of intent based classification improves reliability and comes with the added benefit of being independent of existing telephony infrastructure.

In addition, by allowing user feedback on the classification, we empower the customer with a means to personalize and enhance the spam detection mechanism to their liking, unlike current detection systems. This data remains on the local device and thus does not lead to any privacy concerns.

3 Dataset

A spam dataset has been locally generated had 30,000 text samples belonging to five sectors – Banking and Loans; Tourism; ISP; Credit Cards; Real Estate. Ham data has been acquired by scraping websites containing conversational information. Information was also extracted from relevant public domain videos, using speech-to-text conversion systems. The average word length of these sample sentences is approximately 20 words. Each data sample contains at most 3 sentences, but typically only a single sentence. Each sample acquired has been further multiplexed – by replacing common nouns with their synonyms and proper nouns with relevant alternatives to generate a total of 15,000 data samples for ham.

To prevent unwanted bias associated with an imbalanced dataset, the 30,000 samples of spam data is randomly down sampled to 15,000. To ensure a uniformly distributed dataset across aforementioned domains, this random down sampling was performed with weights assigned to each domain. The final dataset hence contained an equal number of ham and spam samples. The train-validation-test split was kept at 25:2:3. In the training phase, 12,500 randomly picked spam and 12,500 randomly picked ham samples were used to train model 1 (Fig. 1). Validation was done by comparing the results of 1000 samples to other popular models discussed in Sect. 6. Finally, 1500 new samples of ham and spam were used to test the model. Table 1 denotes the datasets generated for spam and ham for training, validation and testing. Table 2 gives a sample of the labelled call transcripts present in the dataset.

Table 1. Information on the labelled dataset generated

Type	Training	Validation	Testing
Ham	12,500	1,000	1,500
Spam	12,500	1,000	1,500
Total	25,000	2,000	3,000

Table 2. Sample sentences from the labelled dataset, generated by applying scraping and data augmentation techniques

Sl. No	Transcript	Label (Spam: 1, Ham: 0)
1	Good Evening, I am Timothy calling from Vibha Real Estates. Would you be interested in buying our 2bhk and 3bhk flats?	1
2	Hi, this is John from Olobank, are you interested in personal loans. We have many good offers that. Please let me know if you're interested	1
3	Hi sir, I am calling from FoodCart. I am coming to deliver your package in the next 30 min	0
4	Hi Ma'am, I am Michele calling from BestDestinations, enjoy latest discounts and offers by buying our new privilege card	1
5	Hello, this is Rahul. I would like to inform that your appointment for Dr. Rajesh has been scheduled tomorrow at 10 AM	0

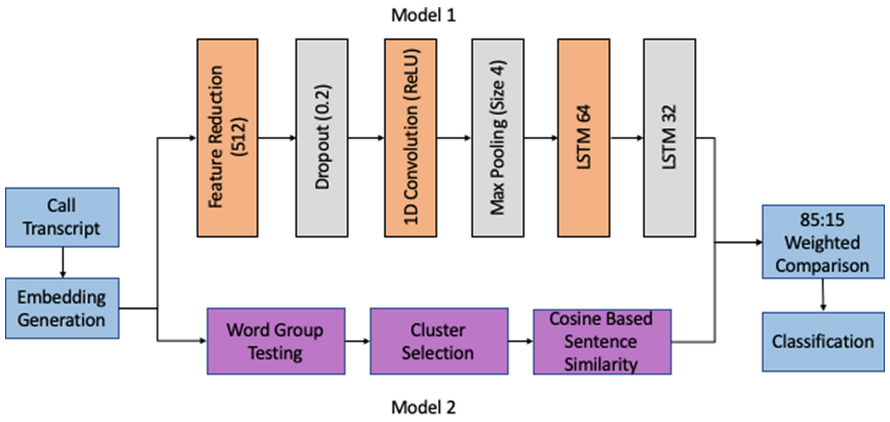

Fig. 1. Flow diagram of the 2PHM approach

4 System Architecture

The flow diagram of the 2PHM approach as shown in Fig. 1 takes an input in the form of an audio recording – the input from the caller and uses a signal processing unit to convert the speech embedded within the audio into text – a transcript of the call. To cover for the lower accuracy in the speech to text system that are often as low as 0.75 when taken word by word, a Trigram based correction system [14] is used in text pre-processing. This helps correct sentences by replacing words that don't fit the context with others from the corpus of 45000 sentences that it was trained on.

Embeddings are generated using the Universal Sentence Encoder (USE) technique. This is a simple, yet very empirically powerful approach that allows for a lot of customization. The encoded vectors are further fed to two parallel models whose results are compared in a weighted manner of 85:15 as discussed in Sect. 6. The first pipeline follows a one-dimensional convolution network followed by a max-pooling unit. The output is then fed to a deep Long Short Term Memory (LSTM) model with 2 layers over the RNN. The second pipeline uses a word group test unit to provide for early elimination of ham data. If the system does not detect the intent to ham by simple word group testing, sentence-based similarity testing [15] is employed. A large dataset consisting of clusters of sentences grouped by similarities in their USE vectors are compared with the input sentence. The group to which the most similar sentence lies defines the classification of the input sentence.

The weighted comparison performed as a final stage averages the outputs generated by both the pipelines and makes the final classification. The decision on the weights is discussed in Sect. 6. This helps significantly minimize false positives in the test set.

5 Methodology

The spam call classification pipeline has various interrelated phases which convert a speech input into a binary output. These are listed below:

5.1 Call Transcript

In order to convert speech to text with minimal latency and large accuracy, we employ the Google Speech to Text [16] API. The output thus generated from this model has a lot of unnecessary punctuations and stop words. This is corrected by using a standard trigram model trained on the remaining sentences of the corpus that aren't used in training. Further processing is performed by the layers that follow this in the architecture.

5.2 Text Preprocessing

The contents of the dataset were manually examined, where invalid and ambiguous samples were removed. The spam dataset was randomly down sampled to 15,000 data samples, to prevent data imbalance. The input sentence (as well as those present in the corpus prior to embedding generation) were cleaned by converting all words into lower case. Further, words were reduced to their base form using lemmatization [17]. Additionally, WordNet [18] is used in generating a map that groups synonymous words being replaced by a single synonym to reduce the variation in the data. For example, "house", "dwelling" and "residence" are all replaced by "home".

5.3 Embedding Generation

Word embeddings are feature representations of text numerically, where words having similar semantic meaning are represented by similar vectors in a predefined vector space.

Each word is mapped to a vector projected onto a higher dimensional embedded space. The notion of distance between the embeddings is used to determine their similarity.

Universal Sentence Encoder [19] is a means of embedding text using transfer learning techniques which perform a lot better than preexistent models. The model used in this paper is that of a Deep Averaging Network wherein input embeddings for words and bigrams are averaged and then passed through a feed forward deep neural network. This supports a linear compute time for the system at hand i.e., an O(n) operation. The equation below estimates similarity using angular distance with cosine-based similarity.

$$sim(u, v) = \left(1 - arccos\left(\frac{u.v}{|u|.|v|.\pi}\right)\right) \tag{1}$$

Hyperparameters are trained using the Vizier and cross validation approaches.

5.4 1 – Dimensional Convolution and LSTM

A 1 – Dimensional convolutional layer followed by an LSTM network can take the context of the sentence and predict the class. Each sentence is converted to into an embedding vector of length 512 using the universal sentence encoder [19] as discussed in the previous section. This vector is used as input to the LSTM network. The LSTM network contains a 1D convolutional layer of 1024 filters and a kernel size of 5, followed by a ReLu activation. This is followed by 1D max pooling layer with a pool size of 4 and a stride of 1, the output shape of the layer is 1024. A LSTM layer with 64 units follows this, which has an output shape of size 64 that leads to an LSTM layer with 32 units, a sigmoid activation is used to generate the final output with shape 1. An output in the range 0 to 0.5 is classified as HAM and one in the range 0.5 to 1 is classified as SPAM. Figure 1 also gives a diagrammatic view of the neural network employed in this task.

5.5 Word Group Testing

A quick means of identifying SPAM is by detecting certain word groups [20] that the user classifies as spam. This may include detection of calls relating to loans, vacations, real estate or even the stock market. Synonyms of these corresponding words can be generated and these are then used in an early extraction of the intent of the conversation.

A set instance classifier is used to decide on whether a synonym set S should include an instance t denoted as f(S,t). The function f here is invariant to the internal ordering of the set S. A neural network architecture was designed so that it directly learns to represent the permutation inversion set.

If we take the quality score of an input set S as q(S) and the quality score, post addition of the instance t as q(S ∪ t) then

$$f(S, t) = \Phi(q(S \cup t) - q(S)) \tag{2}$$

where Φ(x) is the Sigmoid function.

Word Group testing returns the intent, if detected with minimal latency and thus, if classified as SPAM, can be used to exit the entire operation immediately. If this approach fails to detect the intent of the word, we move forward with the computationally more expensive sentence similarity approach.

5.6 Model Based Clustering

Prior to performing sentence similarity, that compares the sentence's embedding with thousands of others, a model-based clustering approach with 5 clusters [21] is used. In model-based clustering, data is believed to be pre-clustered with variant distributions. The authors in [22] define the probability density function for finite mixture distribution models as given in Eq. (3)

$$f(y_i, \Psi) = \sum_{i=1}^{g} \Pi_i f_i(y_i, \theta_i) \tag{3}$$

where $f(y_i,\Psi)$ is the density function of the components and $\Psi = (\Pi, \theta)$ holds the parameters of the mixture model. θ is used to represent a vector representing the unknown parameters of the Probabilistic Distribution Function (PDF) in the mixture model. Using this, the maximum likelihood is as given in Eq. (4)

$$L(\Psi) = \prod_{j=1}^{n} \sum_{i=1}^{g} \Pi_i f_i(y_i|\theta_i) \tag{4}$$

This model also uses the Analytic Hierarchy Process (AHP) which is a multiple criteria decision-making tool. It is divided into a goal, criteria and best alternative, where in the alternative is decided on by the goals and criteria. The AHP uses AIC [23], AWE [24], BIC [25], CLC [26] and KIC [27] as criteria in making a decision with a relative importance vector (RIV). For each of the alternatives – the number of clusters, the Composite Relative Importance Vector (C-RIV) is calculated and the one with the highest value is decided on as the number of clusters.

The approach generated 5 clusters for the dataset which is the same as the number of domains present as discussed earlier. A mean value of the embeddings at each of these 5 clusters is taken and the one with the least loss [28] with respect to the input sentence is considered as the parent cluster. Further comparisons are made within the cluster to significantly reduce computational costs.

5.7 Cosine Based Similarity

While using the neural network generates results that retain context a lot better than plain vector embeddings, they are both time consuming as well as imperfect in few scenarios leading up to false positives. As a secondary test on the classification, a sentence similarity test is conducted. Given the size of our dataset and the excessive use of multiplexing and contextually similar sentences, the likelihood of incorrect classifications is reduced significantly using this approach.

The best matching sentence from amongst the corpus is selected by measuring the cosine of the angle between their vector representations and the class of the most similar sentence is returned as the class of the input sentence.

Two sentences are compared by converting them into embeddings of equal length and measuring the cosine similarity between them. The embeddings are generated using the universal sentence encoder that generates a vector of size 512 irrespective of the sentence length to support easy comparison. Additionally, this also retains context of words in

the sentence unlike simpler techniques like Word2Vec [29]. The cosine similarity metric [30] is defined in Eq. (5):

$$S(a, b) = \frac{a.b}{|a|.|b|} \tag{5}$$

Where S is a function measuring the similarity between two entities a and b. Here a and b are the embedding vectors generated and a dot product is performed on them.

This metric generated a better hyperplane when compared to other metrics such as the Manhattan distance or Euclidian distance measures. Two sentences are considered equivalent if they have a similarity index of value 1 and completely dissimilar if the value is 0. The best matching sentence from amongst the corpus is selected by performing this operation and the class of the most similar sentence is returned as the class of the input sentence.

6 Analysis

The 2PHM model discussed in the previous sections uses Deep Neural Networks to classify text. In this section we compare it to other machine learning based models used in literature.

6.1 Bernoulli Naïve Bayes

Prior work in textual spam detection with machine learning approaches includes the use of the Bernoulli Naïve Bayes algorithm [31]. This uses the Bayes Theorem but predicts only 1 of 2 possible answers. When the vocabulary is too large, the model's accuracy drops. By using a Bag of Words model, we were able to relatively improve this.

6.2 K – Nearest Neighbors

This clustering mechanism groups sentences with similar features into one of N groups based on the distance of each sentence from its K closest neighbors [6]. While the model performs better than the Naïve Bayes, it is nowhere as accurate as a Deep Neural Network. The K-NN approach has a very high F1 scores possible which is why a variation of the same, Model Based Clustering [21] is used in the 2PHM.

6.3 Support Vector Machines

Support Vector Machines [5] are excellent machine learning models that construct a hyperplane in N dimensional space. The loss function maximizes the margin between the data points and the hyperplane using the following equation:

$$c(x, y, f(x)) = \begin{cases} 0, & if\ y * f(x) \geq 1 \\ 1, & otherwise \end{cases} \tag{6}$$

The SVM model has a higher accuracy than K-NN and Naïve Bayes on the dataset in Sect. 3 but suffers from lower Precision and Recall.

6.4 Random Forests

Random Forests are relatively more complex Decision Trees [8] that can be used in classification of text. While standard Random Forests have issues with feature learning as the gradient flattens out, the use of Gradient Boosted RF greatly helps resolve this issue. This model produces substantially larger accuracy, recall and precision than the other models discussed but takes a lot more time to train on due to the complexity of the data structure.

The 2PHM model, in comparison to all these models, is superior in terms of accuracy while not losing out in terms of latency. Given the infrequency of training the dataset, we can safely say that the effect of low training speeds can be ignored in this comparison.

7 Experiments and Results

7.1 1D Convolution + LSTM

Training of the system is done only on Model 1 (Fig. 1). Testing and Validation on the other hand was performed for each module independently and for the system as a whole. Several machine learning models such as Naïve Bayes, Support Vector Machines, K-Nearest Neighbors and Random Forests [31] as discussed were compared along with our proposed LSTM based model.

Each model was trained with a batch size of 100. Binary cross entropy was used as a measure of loss while the Adam algorithm [32] was used as the optimizing function. The loss is defined as follows, where P is the predicted class and G is the actual class and N is the total number of training samples as given in Eq. (7)

$$Loss = \frac{1}{N} \times \sum_{i=1}^{N} (G \times \log(P) + (1 - G) \times \log(1 - P)) \tag{7}$$

The models were trained till the validation accuracy of 99% was attained. The performance of these models are summarized in term of recall, precision and accuracy on the validation set are as shown in Table 3. As shown, the model proposed, based on LSTM performed better than traditional machine learning based models in terms of accuracy. Spam sentences from the test set produced a much higher accuracy of 98.46% while ham data had much lower accuracy of 74.1%. The reason for a lower precision is to do with the existence of multiple false positives due to ham data being more generic in nature.

While the LSTM model took almost ten times as long to train over one epoch as the Naïve Bayes model, it learnt the patterns in just 2 epochs, reaching the expected amount of accuracy in validation much before the machine learning models did.

7.2 Word Group Extraction + Sentence Similarity

To compensate for the loss in precision due to the presence of false positives as well as attempting to increase the overall accuracy of the system, we weigh the output of the sentence similarity model and the 1D Convolution + LSTM model.

By applying a cosine-based similarity approach on the embeddings generated by the universal it to classify the call's intent. Table 3 shows the precision, recall and accuracy

Table 3. Comparative results on spam detection from a confusion matrix

Model	Precision (%)	Recall (%)	Accuracy (%)	F1-score
Naïve Bayes	91.21	87.78	77.01	0.89
SVM	93.54	90.33	82.11	0.91
K-NN	96.86	92.02	79.76	0.94
Random Forests	94.29	96.64	84.56	0.95
LSTM	90.53	90.90	91.58	0.91
1D Convolution + LSTM	90.89	97.90	89.18	0.94
Word Group Testing	96.42	98.57	85.52	0.97
2PHM	96.51	94.42	93.32	0.95

of this model. While the accuracy is not as high as others, both precision and recall are excellent. A combination of these two models produces even better accuracy by picking the more reliable result for each input from the test set based on the confidence generated by each one.

7.3 Test Results

The final prediction of the classifier is done by performing a weighted comparison between both models, assigning a 15 percent weightage for the Word Group Extraction + Sentence Similarity Model and 85 percent weightage (threshold value of 0.85) for 1D Convolution + LSTM model. The number (defined as threshold) was decided by trial and error as shown in Fig. 2.

The test samples used 3000 call transcripts to test the hybrid system as a whole to determine the final test accuracy. This shows a significant increase from earlier while not compromising on either precision or recall. The results obtained are as shown in Table 3. The system achieves an overall test accuracy of over 93%, precision over 96% and recall over 94% respectively. As shown in Table 3, our proposed weighted model performs significantly better than existing LSTM based spam-text classification models [33] adopted to our dataset. The time taken for a decision is also pretty small as shown in Table 4.

Table 4. Measuring the latency to classifying a sentence

Iteration	Number of words processed	Time taken for classification
1	10	2.76
2	14	2.81
3	12	2.83
4	13	2.69

As shown above, the number of words does not have a major impact on the task of classification. For practicality reasons, only one sentence having enough context has been tested for in each of these iterations. Given this, we can conclude that on an average of around 3 s, the classifier is able to predict the intent of the call and can thus transfer the call to the customer which lies within the typical 10 s ring tone range. This makes the delay acceptable in real world scenarios.

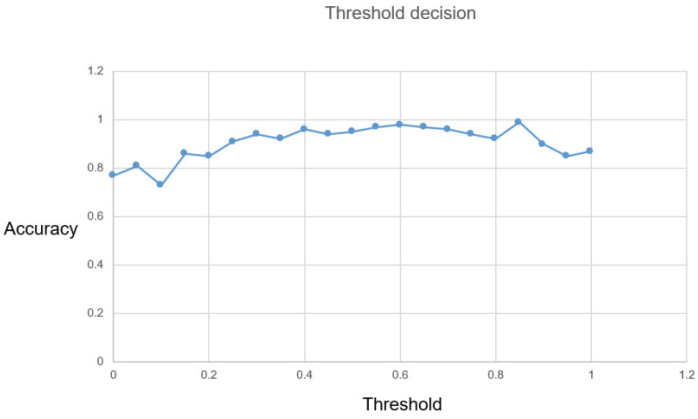

Fig. 2. Threshold decision based on maximizing accuracy

8 Conclusion and Future Work

The 2 Phase Hybrid Model thus developed classifies calls with an accuracy of 93.32% over a variety of data inputs across various domains. The system provides for excessive customization and correction of the model by incorporating the user feedback for each call. In order to adjust the weights, we can add the call data back into the corpus and retrain the models over a short period of time.

The use of two models – the LSTM network and the sentence similarity network in making a decision proves to be very effective in the elimination of false positives against a model consisting of only the LSTM network. This system works in real-time, and hence can be included in the native call application of a mobile device to identify a spam call from its intent. A pipeline can be constructed to replicate this model to support any language.

Future work in this area involves other use cases that this model can be extended to. These include identifying and dynamically generating other actionable items from the transcript such as setting reminders, making appointments, etc. based on Natural Language Understanding. Including this feature in our external call application can greatly enhance user experience in cases where users are unable to manually answer phone calls. Future work also encompasses integrating On-Device Automatic Speech Recognition with the Spam detection model ported onto the device. This will ensure that the user's data and preferences are stored locally on the device and are not transmitted back to the cloud, preventing any privacy compromises.

Acknowledgment. First and foremost, we would like to express our gratitude to the team at Samsung R&D Institute India-Bangalore, for providing us with the problem statement to work on. This work would not have been possible without the constant motivation and help of our mentor, Mr. Nandan at Samsung R&D Institute India-Bangalore. We would also like to thank them for extending their support towards the creation of the datasets used in this work. We would like to thank Dr. K N Subramanya, Principal RV College of Engineering for facilitating an Industry Institute Interaction and MoU with Samsung R&D Institute India-Bangalore.

References

1. Tu, H., Doupé, A., Zhao, Z., Ahn, G.: SoK: everyone hates robocalls: a survey of techniques against telephone spam. In: 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, pp. 320–338 (2016)
2. Kim Fai Kok: Truecaller Insights: Top 20 Countries Affected By Spam Calls & Sms. https://truecaller.blog/2019/12/03/truecaller-insights-top-20-countries-affected-by-spam-calls-sms-in-2019/ (2019). Accessed 01 Oct 2020
3. Roy, P.K., Singh, J.P., Banerjee, S.: Deep learning to filter SMS spam. Futur. Gener. Comput. Syst. **102**, 524–533 (2020)
4. Dada, E.G., Bassi, J.S., Chiroma, H., Adetunmbi, A.O., Ajibuwa, O.E.: Machine learning for email spam filtering: review, approaches and open research problems. Heliyon **5**(6) (2019)
5. Evgeniou, T., Pontil, M.: Support vector machines: Theory and applications. In: Paliouras, G., et al. (eds.) Machine Learning and its Applications, pp. 249–257. Springer, New York (2001)
6. Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: KNN model-based approach in classification. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) OTM 2003. LNCS, vol. 2888, pp. 986–996. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39964-3_62
7. Pouria, K., Sunita, D.: Short survey on naive Bayes algorithm. Int. J. Adv. Res. Comput. Sci. Manage. **4**(11), 607–611 (2017)
8. Jehad, A., Rehanullah, K., Nasir, A., Imran, M.: Random forests and decision trees. Int. J. Comput. Sci. Issues **9**(5), 272–278 (2012)
9. Wu, Y., Bagchi, S., Singh, N., Wita, R.: Spam detection in voice-over-IP calls through semi-supervised clustering. In: 2009 IEEE/IFIP International Conference on Dependable Systems & Networks, Lisbon, pp. 307–316 (2009)
10. Chaisamran, N., Okuda, T., Blanc, G., Yamaguchi, S.: Trust-based VoIP spam detection based on call duration and human relationships. In: 2011 IEEE/IPSJ International Symposium on Applications and the Internet, Munich, Bavaria, pp. 451–456 (2011)
11. Kolan, P., Dantu, R.: Socio-technical defense against voice spamming. ACM Trans. Auton. Adapt. Syst. **2**(1), 2–7 (2007)
12. Mathieu, B., Niccolini, S., Sisalem, D.: SDRS: a voice-over-IP spam detection and reaction system. IEEE Secur. Privacy **6**(6), 52–59 (2008)
13. Li, H., et al.: A machine learning approach to prevent malicious calls over telephony networks. In: 2018 IEEE Symposium on Security and Privacy (SP), pp. 53–69 (2018)
14. Dashti, S.M.: Real-word error correction with trigrams: correcting multiple errors in a sentence. Lang. Resour. Eval. **52**(2), 485–502 (2017). https://doi.org/10.1007/s10579-017-9397-4
15. Prabhu, A.D., Arora N., Vatsal S., Ramena G., Moharana S., Purre N.: On-Device Sentence Similarity for SMS Dataset. ArXiv, abs/2012.02819 (2020)
16. Cloud Speech to Text. https://cloud.google.com/speech-to-text/. Accessed 21 Jan 2020

17. vor der Brück, E.S., Alexander, T.M.: Lexicon-assisted tagging and lemmatization in Latin: a comparison of six taggers and two lemmatization models. In: 2015 Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH), pp. 105–113 (2015)
18. Zhu, X., Yang, X., Huang, Y., Guo, Q., Zhang, B.: Measuring similarity and relatedness using multiple semantic relations in WordNet. Knowl. Inf. Syst. **62**(4), 1539–1569 (2019). https://doi.org/10.1007/s10115-019-01387-6
19. Daniel, C., et al.: Strope Brian. Universal Sentence Encoder, Kurzweil Ray (2018)
20. Shen, J., Lyu, R., Ren, X., Michelle, V., Brian, S., Han, J.: Mining entity synonyms with efficient neural set generation. In: 2018 AAAI (2018)
21. Zhang, W., Di, Y.: Model-based clustering with measurement or estimation errors. Genes **11**, 185 (2020)
22. Akogul, S., Erisoglu, M.: An approach for determining the number of clusters in a model-based cluster analysis. Entropy **19**(9), 452 (2017). https://doi.org/10.3390/e19090452
23. Banfield, J.D., Raftery, A.E.: Model-based Gaussian and non-Gaussian clustering. Biometrics **49**(3), 803–821 (2013)
24. Schwarz, G.: Estimating the dimension of a model. Ann. Stat. **6**(2) 461–464 (2018). www.jstor.org/stable/2958889. Accessed 10 June 2020
25. Christophe, B., Gilles, C., Gerard, G.: Assessing a mixture model for clustering with the integrated classification likelihood. In: 2017 IEEE Transactions on Pattern Analysis and Machine Intelligence – PAMI (2017)
26. Cavanaugh, J.E.: A large-sample model selection criterion based on Kullback's symmetric divergence. Stat. Probab. Lett. **2009**(42), 333–343 (2009)
27. Katsuki, C., Katsuhito, S., Satoshi, N.: Training Neural Machine Translation using Word Embedding-based Loss. https://arxiv.org/pdf/1807.11219.pdf (2018)
28. Lee, G.Y., Manski, S., Maiti, T.: Actuarial applications of word embedding models. ASTIN Bull. **50**(1), 1–24 (2020). https://doi.org/10.1017/asb.2019.28
29. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: ICLR Workshop Papers (2013)
30. vor der Brück, T., Pouly, M.: Text similarity estimation based on word embeddings and matrix norms for targeted marketing, pp. 1827–1836 (2019). https://doi.org/10.18653/v1/N19-1181
31. Nguyen, N.-T., Manolopoulos, Y., Iliadis, L., Trawiński, B. (eds.): ICCCI 2016. LNCS (LNAI), vol. 9875. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45243-2
32. Yi, D., Ahn, J., Ji, S.: An effective optimization method for machine learning based on ADAM. Appl. Sci. **10**, 1073 (2020)
33. Raj, H., Yao, W., Banbhrani, S.K., Dino, S.P.: LSTM based short message service (SMS) modeling for spam classification. In: Proceedings of the 2018 International Conference on Machine Learning Technologies (ICMLT 2018), pp. 76–80. Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3231884.3231895