Combating Robocalls with Phone Virtual Assistant Mediated Interaction

Sharbani Pandit
Georgia Institute of Technology

Krishanu Sarker
Georgia State University

Roberto Perdisci
University of Georgia, Georgia Institute of Technology

Mustaque Ahamad
Georgia Institute of Technology

Diyi Yang
Georgia Institute of Technology

Abstract

Mass robocalls affect millions of people on a daily basis. Unfortunately, most current defenses against robocalls rely on phone blocklists and are ineffective against caller ID spoofing. To enable detection and blocking of spoofed robocalls, we propose a NLP-based smartphone virtual assistant that automatically vets incoming calls. Similar to a human assistant, the virtual assistant picks up an incoming call and uses machine learning models to interact with the caller to determine if the call source is a human or a robocaller. It interrupts a user by ringing the phone only when the call is determined to be not from a robocaller. Security analysis performed by us shows that such a system can stop current and more sophisticated robocallers that might emerge in the future. We also conduct a user study that shows that the virtual assistant can preserve phone call user experience.

1 Introduction

The number of scam/spam phone calls people receive are increasing every day. Phone security company YouMail estimates that April 2022 saw 3.9 billion robocalls in the United States and the Federal Trade Commission (FTC) phone complaint portal receives millions of complaints about such fraudulent and unwanted calls each year [21]. Voice scams have become such a serious problem that people often no longer pick up calls from unknown callers. Robocalling, voice phishing, and caller-id spoofing are some of the techniques that are used by fraudsters and criminals in these scams.

A number of commercial applications are available for blocking unwanted calls (Hiya, Truecaller, Youmail etc.) but call blocking defenses used by such apps can be easily undermined with caller ID spoofing [52]. Such spoofing is easy to achieve, and robocallers have resorted to tactics like neighbor spoofing (i.e., the spoofed caller ID is similar to the targeted phone number) to increase the likelihood that the user will pick up the call. Both industry groups and regulatory bodies have explored stronger authentication with efforts such as SHAKEN/STIR [18], which enable the callee to verify the authenticity of the caller ID. Although the Federal Communications Commission (FCC) mandated US telecom companies to begin implementing SHAKEN/STIR starting June 30, 2021 [32], its slow adoption and global nature of phone scams will limit its effectiveness. Furthermore, as new phone numbers can be cheaply acquired and used to overcome blacklists, victims will continue to fall for robocall scams that do not use caller ID spoofing [19].

At a high level, the robocall problem resembles the email spam problem, in which information about the source of an email could potentially be spoofed. Over the years, the security community has been successful in developing effective spam filtering solutions [58, 65, 69]. However, most techniques used in such solutions rely heavily on the content of the email. This is not the case for phone spam, since content of the caller's message becomes available only after the call has been picked up. By this time, the user is already exposed to the spam call. In response to this, recently, a number of automated caller engagement systems that collect call content before forwarding a call have been proposed. The Call Screen feature available on the Google Pixel phone app provides call context to the user via a real time transcript of the call audio. However, distinguishing robocallers from humans is not addressed by such systems since their focus is on providing meaningful call context (e.g., caller name, call purpose etc.).

To protect users from mass, spoofed, targeted and evasive robocalls, we introduce a smartphone *virtual assistant* named RoboHalt, which uses a novel voice interaction model. It picks up incoming phone calls on behalf of the callee and aims to make a natural conversation with the caller to detect robocalls based on the conversation. RoboHalt asks questions that occur naturally among humans at the beginning of a phone call. The questions are designed in a way that it is easy and natural for humans to respond to; at the same time, it is difficult for robocallers to provide appropriate responses without incurring significant additional cost. Based on the responses provided by the caller, RoboHalt uses a combination of Natural Language Processing (NLP) based machine learning models to decide if the caller is a human or a robocaller. It only forwards a call to the callee once it has determined that the caller is a human. RoboHalt can stop mass robocalls even if caller ID spoofing is used because it does not solely depend on phone blocklists. Since it requires human-like interaction from callers, pre-recorded robocalls can be successfully blocked. Such interaction also allows RoboHalt to block robocalls when robocallers target their victims and use voice activity detection to bypass existing defenses. It aims to achieve this while preserving caller experience.

In summary, we make the following contributions:

- We design an interactive virtual assistant (RoboHalt) which can vet phone calls on behalf of the user by making a natural conversation with the caller.
- RoboHalt uses multiple NLP machine learning models to detect if the caller is a human or a robocaller. To the best of our knowledge, we are the first to develop such

a defense that detects robocalls even when robocallers use caller ID spoofing and voice activity detection.
- To conduct rigorous security evaluations, we recruited red team members to craft attacks to defeat RoboHalt. The security analysis showed that 95% of the mass robocalls, 84% evasive robocalls and 87% of targeted robocalls were not able to directly reach the callee, thus eliminating user interruption by unwanted robocalls.
- To demonstrate the usability of our system, we conducted an IRB approved user study. Its results demonstrate that users had a positive experience and would benefit from using such a system.

2 Related Works

Voice spam has grown significantly in recent years [25, 36, 49, 61, 68, 71]. It has been shown that phone blacklisting methods provided by smartphone apps (e.g., Truecaller [15], Nomorobo [8], Youmail [17], Hiya [7] etc.) or telephone carriers (e.g., " call protect services offered by AT&T [1], Verizon [16] etc.) can be helpful. However, these services typically rely on historical data such as user complaints or honeypot-generated information [42, 52, 67], and their overall effectiveness tends to be low due to caller ID spoofing [22]. Robokiller [34], a smartphone application, on the other hand, features an Answer Bot that detects spam calls by forwarding all incoming calls to a server, which analyzes their audio to determine if the call source is a recording. The audio analysis techniques used by Robokiller can be countered by sophisticated robocallers that use voice activity detection.

A number of research papers have explored caller ID authentication [51, 59, 60, 72]. The SHAKEN/STIR [10, 11] effort by the IETF and SIP Forum enables the callee to verify the correctness of the caller ID. However, caller ID authentication alone may not be sufficient if victims continue to fall for scams that don't use spoofing [20]. In recent work that conducted a large scale user study [73], it was shown that a significant fraction of users fall victim to telephone scams. Prasad et. al [56] use call audio and metadata to cluster robocall spam campaigns. Also, [64] evaluates the impact of user interface design elements on user decision-making for robocall blocking applications. In contrast, our work aims to handle incoming calls and provide a solution to block current and future robocalls from reaching recipients without user intervention, even in the presence of caller ID spoofing.

In the latest version of its Phone app, Google offers an automatic call screen feature where users can opt to have their calls from unknown callers automatically screened to block robocalls. Although this feature indicates that call content could be used for such blocking, our experiments reported in Section 4.2 found that this is not the case. Therefore, call screen does not currently stop robocallers that spoof benign caller IDs to deliver their spam content.

The questions asked by our phone virtual assistant during its interaction with a caller can be viewed as challenges which the caller must pass in order for the call to be forwarded. Audio captchas aim at achieving a similar goal where users need to prove that they are not robots. Typically they consist of a series of spoken words/numbers and some form of audio distortion or background noise [47]. However, several researchers have explored attacks that can easily break audio captchas [28, 29, 43, 70]. Solanki et. al [66] have demonstrated how off-the-shelf speech recognition systems can be used to break all commercially deployed audio captchas. Hence, currently used audio captchas are not effective when used against sophisticated robocallers. Fanelle et. al [37] explored cognitive audio captchas where users have to solve puzzles, answer math questions or identify sounds to pass the challenge. However, this disrupts the natural call experience. The interaction RoboHalt requires can be thought of as a NLP-based captcha, where the natural flow of the conversation is maintained to preserve usability. Moreover to pass the challenges posed by RoboHalt, robocallers need more advanced AI capabilities to truly comprehend what the virtual assistant is asking them to do.

RoboHalt is a conversational agent that detects and blocks robocalls. Typically dialog systems communicate with users in natural language (text, speech, or both), and fall into two classes: task-oriented [50] and chatbots. Task-oriented dialogue agents use conversation with users to help complete tasks. Dialogue agents in digital assistants (Siri, Alexa, Cortana, etc.), give directions, control appliances, find restaurants, or make calls. By contrast, chatbots [77] are systems designed for extended conversations, set up to mimic the unstructured conversations or 'chats' characteristic of human-human interaction [78]. Chatbots [39, 45, 75] are conversational agents designed to mimic the appearance of informal human conversation. These conversational agent systems are enormously data-intensive; Serban et. al. [63] estimate that training modern chatbots requires hundreds of millions or even billions of words. However, data on robocall messages and human assistant conversations is limited. Hence building such a system that has high accuracy is challenging. Also, the need for the system to be resistant to adversaries makes it even more challenging.

3 System Design

3.1 System Overview

RoboHalt picks up incoming calls on behalf of the user without user intervention. If the incoming call is from a safelisted caller, it immediately notifies the user by ringing the phone. A safelist can include the user's contacts and other legitimate caller IDs (public schools, hospitals etc.). On the other hand, if the call is from a blocklisted caller, RoboHalt blocks it and does not let the phone ring. However, if the caller ID belongs to neither a safelist nor a blocklist, RoboHalt picks up the call without ringing the phone and uses multiple techniques to detect robocallers. Upon picking up the call, it greets the caller and lets the caller know that he/she is talking to a virtual assistant. During the conversation, RoboHalt randomly chooses to ask a question from a predefined pool of questions that naturally occur in human conversations. It determines whether the caller

is a human or a robocaller based on the responses provided by the caller. The number of questions RoboHalt asks before making this decision depends on the responses provided by the caller and the confidence RoboHalt has in labeling them as appropriate/not appropriate. To strike a balance between usability and security, the maximum number of questions asked before making a decision is five. We explain this in detail in later sections. If the caller is deemed to be a human, the call is passed to the callee along with the transcript of the purpose of the call. Otherwise, the caller is deemed to be a robocaller and RoboHalt blocks the call and sends a notification to the user.

3.2 Challenges

3.2.1 Availability of Datasets

Building natural language models presents numerous challenges. First, a large annotated dataset is required to build highly accurate NLP models. However, large datasets consisting of phone call conversations are difficult to find or collect. Similarly, datasets of real robocalls are not easy to collect as they typically require setting up a large telephone honeypot. To overcome this challenge, we used datasets generated for other NLP tasks (for instance, speech-to-text) and adapted them to suit our purposes.

3.2.2 Real-time Performance

Second, models that have the capability of fully understanding natural language and user intent tend to be very complex; this is still an area of ongoing research in the field of natural language processing [33, 40]. Also, we intend RoboHalt to be used in real-time. Therefore, the models should be light-weight and have low latency. To achieve our goal of high accuracy in real time while keeping the overall system light-weight, instead of relying on one single large model we use an ensemble of multiple light-weight ML models.

3.2.3 NLP Challenges

Finally, most of the work on conversational agents has focused on usability and how the conversation can be made more human-like [23, 44, 78]. However, we need to strike a balance between usability and security, since RoboHalt is designed to face both human callers and robocallers. Having the conversational agent succeed in an adversarial environment while at the same time being user-friendly to human callers is an additional challenge [76].

3.3 Threat Model

3.3.1 In-scope Threats

In this section we describe the scope of threats that our system is designed to protect against.

Non-targeted and Targeted Robocalls: Non-targeted robocalls are mass calls that are not directed at specific victims. Most current robocalls fall in this category. In contrast, in targeted robocalls, the attacker knows both the name and phone number of the intended victims (there have been several data breaches which could provide associations between phone numbers and names). Hence, the attacker could generate pre-recorded robocalls that start by asking to speak with a specific person before continuing with the actual robocall message. Since RoboHalt requires human-like answers to multiple questions, it can stop such unwanted targeted robocalls.

Evasive Robocalls: We assume an evasive robocaller does not have AI capabilities, hence cannot comprehend what RoboHalt is saying in real-time; however, we assume the attacker has knowledge of how RoboHalt works and can try to bypass it. Such calls are currently not common, but may emerge in the future in an attempt to defeat intelligent phone call defense tools such as the one we propose. An evasive robocaller might use voice activity detection to identify interruptions from the callee's side and pause accordingly to give the impression of liveness. It may also learn common questions and try to provide prerecorded responses in a certain order in an attempt to bypass RoboHalt. However, since we randomize the order of RoboHalt's questions and use multiple challenges to detect a robocaller, the likelihood of providing a reasonable response and overcoming all challenges without understanding the questions is low. We discuss this in detail in Section 4.

3.3.2 Out-of-scope Threats

RoboHalt does not protect against the following attacks:

Unwanted calls from an AI equipped attacker: Attacks in which robocallers emulate human conversations, to the extent that even humans have difficulty distinguishing between an automated call and a human caller, are out of scope. Such AI equipped robocall systems are not common in the phone fraud ecosystem and building them is expensive and requires a substantial amount of resources. Since robocallers aim at making cheap mass calls so that they can reach a vast number of targets, crafting such AI equipped attacks will add significant cost for them.

Unwanted live calls from humans: Unwanted calls can be made by telemarketers, debt collectors or humans working for a scam campaign. Since our system looks for human-like interaction from the callers, such unwanted live calls from humans cannot be stopped. Since malicious actors who craft these campaigns aim at decreasing their cost and increasing the profit by fooling victims, having a human caller instead of a robocaller increases their cost significantly.

3.4 RoboHalt Use Cases

There can be two example use cases of RoboHalt. It can be embedded with the Phone app where all incoming calls are handled locally. With each incoming call, RoboHalt examines the caller ID and interacts with the caller without making the callee aware. Once the call is deemed to be from a human, RoboHalt makes the phone ring and lets the caller reach the callee. One other scenario is where RoboHalt is hosted at a network carrier. In this case, all incoming calls go through RoboHalt hosted by the carrier and it performs the above mentioned activities. Once the call is deemed to be from a human, the carrier then forwards the call to the user (i.e., the callee, in this case).

3.5 RoboHalt Workflow

RoboHalt handles all incoming calls and first makes a decision based on the caller ID of the incoming call. There can be three cases: (i) the caller ID belongs to a predefined safelist, (ii) the caller ID belongs to a predefined blocklist, or (iii) the caller ID does not belong to these predefined lists and is an unknown caller. If the caller ID is safelisted, RoboHalt immediately passes the call to the callee. RoboHalt blocks calls from blocklisted caller IDs and does not ring the phone. Additional analysis is performed for calls from unknown callers to understand the nature of the call. The following steps are taken to handle this case.

- RoboHalt picks up the call, greets the caller and informs them that they are talking to a virtual assistant. RoboHalt then randomly chooses to ask the caller to hold or continue the conversation.
- Once the caller has responded to the previous question, RoboHalt then asks a question from the question pool described below. The question is chosen by RoboHalt according to rules defined later in this section. The questions are easy and natural for humans to answer, but without comprehending what the question is, it is difficult for robocallers to answer. RoboHalt then determines if the response is appropriate for the question asked and assigns a label (appropriate, not appropriate). RoboHalt also assigns a confidence score with each label.
- RoboHalt might ask another question or make a decision on whether the caller is a human or robocaller. The number of questions RoboHalt asks the caller before making this decision depends on the responses provided by the caller earlier and the confidence RoboHalt has in labeling each of them. For example, if RoboHalt is highly confident that the caller is a human or a robocaller after asking two questions, it chooses not to ask a third question. Otherwise, it asks the next question. The minimum and maximum number of questions RoboHalt asks before labeling a caller human or robocaller is two and five respectively.
- RoboHalt asks for the purpose of the call before completing the conversation if it has not already been asked. The response to this question provides additional context about the incoming call. It is important that RoboHalt asks questions in a random order to avoid a robocaller simply playing pre-recorded responses in a certain order to fool it.
- Based on the steps discussed above and following the algorithm discussed in Section 3.7.1, RoboHalt determines whether the caller is a human or not. If the caller is deemed to be a human, the call is forwarded to the user along with the context captured during the initial interaction and other relevant information such as the name of the caller. Calls from robocallers are blocked and a notification is provided to the user.

3.6 RoboHalt Questions

RoboHalt chooses questions to ask from Table 1. The questions are designed to be easy and natural for a human caller to respond to during a phone call. However, the responses to these questions are specific enough that they do not typically appear in robocall messages. While designing the questions, we aimed to balance between usability and security. The trade-off we intend to make depends on the following factors.

- *Number of questions:* Asking too many questions might annoy a human caller and degrade call experience. On the other hand, asking too few questions can make it easy for the attackers to evade the system.
- *Type of questions:* To preserve usability, we are limited to questions that are reasonably common at the beginning of a phone conversation between people who may not know each other. However, at the same time there should be enough variations in the questions so that the system is robust against attacks.
- *Gathering call context:* Besides preserving usability, determining the context of the call is another important goal. Therefore, questions should be asked so that meaningful information can be captured (e.g., the purpose of the call, the caller's name, etc.).

The list of question types that RoboHalt can ask includes:

Hold: As seen very commonly in phone conversations, RoboHalt asks the caller to hold briefly.

Context Extractor: This type of questions is asked to extract context (e.g., purpose of the call). While pre-recorded robocalls contain spam messages (free cruises, vehicle warranty, etc.), human callers will provide legitimate context.

Name Recognizer: If the caller knows the callee, they would be capable of saying who they are trying to reach.

Relevance: These questions commonly occur in natural human conversations and allow RoboHalt to determine if the caller can actually comprehend the questions and provide meaningful responses.

Repetition: It is reasonable to expect that, if asked, a human can repeat a statement by either saying it again or providing a semantically similar statement. However, without understanding the question, robocallers won't be able to perform this task.

Speak up: It is natural in a human conversation to ask a caller to speak up. Our system asks this question to determine if the caller can indeed speak up when asked to do so.

Follow up: RoboHalt may choose to ask a follow up question, after asking certain questions. For example, it might ask "Can you please tell me more about it?" as a follow up question after asking "How can I help you?". Moreover, after asking, "Who are you trying to reach?", RoboHalt might ask "Did you mean [name]?" as a follow up question. For example, if the name of the callee is *Taylor*, RoboHalt can ask "Did you mean Taylor?" or ask "Did you mean Tiffany?", where Tiffany is not the name of the callee.

It is important to note that RoboHalt uses multiple variations of each question. For example, the question "How are you?" can have multiple variations with the same meaning such as "How are you doing?", "How's it going?" etc. This enables us to defend against robocallers that can use the audio length

Table 1: RoboHalt Question Examples

Question type	Example
Hold	Please hold briefly.
Context Extractor	How can I help you?
Name Recognizer	Who are you trying to reach?
Relevance	How are you doing?
	How do you like the weather today?
Repetition	Can you please say that again?
Speak up	Can you speak up please?
Follow up	Can you tell me more about it?
	Did you mean [name]?

of a question to determine what question was asked. With multiple variations of the same question, a robocaller truly needs to comprehend what RoboHalt is saying to provide an appropriate response. An AI equipped attacker might be able to automatically learn the questions. However, it is going to be difficult for the attackers considering the lack of significant amount of labeled data of such type of human assistant-like conversations. The effectiveness of these questions is discussed in detail in the later sections.

Question Order RoboHalt uses the following rules to choose a question at each turn:

- After the announcement and initial greeting, our system randomly chooses to ask the caller to hold or not.
- RoboHalt then randomly asks the Context Extractor or Name Recognizer question with equal probability.
- At this point RoboHalt might choose to continue the conversation or block/forward the call based on the previous responses. If RoboHalt decides to continue the conversation, it randomly chooses one of the Follow up, Relevance, Repetition, Name Recognizer, Hold questions with high probability or Speak up with low probability.
- If RoboHalt decides to ask a fourth or fifth question, it randomly chooses one of the following questions with equal probability, Context Extractor, Repetition, Name Recognizer, Hold, Relevance, Speak up.

RoboHalt asks a specific question only once during its interaction with the caller. The rules are designed to keep the conversation similar to a typical phone call while increasing the entropy for the attacker so that there is no specific pattern that the attacker can exploit. An example conversation is shown below:

```
VA: Hello, you've reached the
    virtual assistant. How can I help you?
Caller: I'm calling to make an appointment.
VA: I'm sorry, I didn't get that.
    Can you please repeat?
Caller: I said I wanted to make an appointment.
VA: Please wait while I forward your call.
```

3.7 System Architecture

In this section, we describe our system architecture, depicted in Figure 1, which is independent of the implementation environment. With each incoming call, the Metadata Detector module determines if the caller ID is present in the safelist or blocklist. Calls from unknown callers are passed to the Controller of the Robocall Detector.

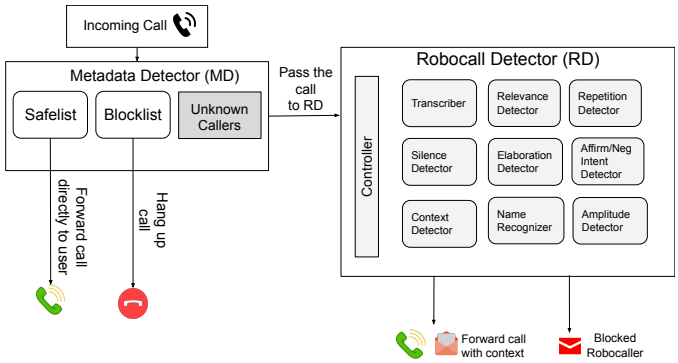


Figure 1: System Architecture

3.7.1 Controller

The Controller determines which question to ask the caller at every turn. After asking each question, the controller records the response from the caller. The audio from the caller is recorded until the caller finishes speaking or 20 seconds elapse, whichever is the minimum. The audio recording is then transcribed through the Transcriber module. The controller uses individual modules to label the transcripts (of responses) to determine if it is an appropriate response. For example, Relevance Detector determines if the response is an appropriate one to the relevance question, Repetition Detector determines if the response is an appropriate one to the repetition question etc. Each of these modules (Relevance Detector, Repetition Detector etc.) predicts a label (appropriate/ not appropriate) and a confidence score with it. After every prediction, RoboHalt calculates a score S_i (with $1 \leq i \leq 5$) inspired by the sequential probability ratio test (SPRT) [74], according to the following equation:

$$S_i = S_{i-1} + min((i/\lambda), 1) \times log\, A_i \quad (1)$$

Here, $log\ A_i = log\ (C_i/1 - C_i)$ where C_i is the confidence assigned by the corresponding module and λ is a tunable parameter that determines the weight of the i^{th} prediction. In our implementation we set $\lambda = 3$. S_i determines the stopping rule of RoboHalt, i.e when to stop asking questions. As in classical hypothesis testing, SPRT starts with a pair of hypotheses, H_0 and H_1. We specify H_0 and H_1 as follows.

H_0 : Caller is human

H_1 : Caller is robocaller

The stopping rule is a simple threshold scheme:

- $a < S_i < b$: continue asking questions.
- $S_i \geq b$: Accept H_1
- $S_i \leq a$: Accept H_0

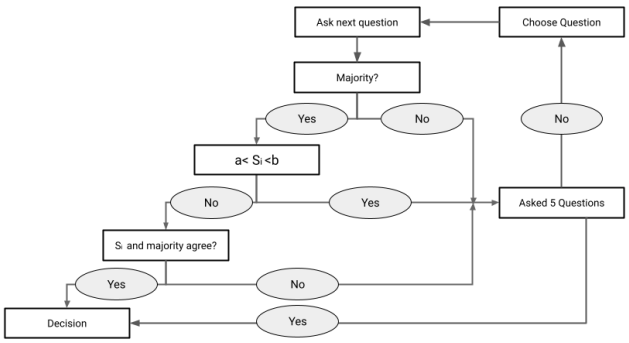


Figure 2: RoboHalt Controller Algorithm

where a and b depend on the desired type I and type II errors, α and β. We choose α and β to be 5%.

$$a \approx \log\frac{\beta}{1-\alpha} \quad and \quad b \approx \log\frac{1-\beta}{\alpha} \tag{2}$$

RoboHalt requires at least two and at most five predictions to make a decision.The controller implements the algorithm depicted in Figure 2 to make this decision. At any given point, if a majority does not exist in the prediction labels, RoboHalt chooses to ask the next question. If a majority exists but the S_i score is between a and b, RoboHalt chooses to continue the conversation and asks the next question; otherwise it checks if the majority labels and the label supported by SPRT (according to the stopping rule specified above) are in agreement. If yes, RoboHalt finalizes the label and makes a decision to forward the call if the caller is labeled human, and blocks the call if the caller is labeled as *robocaller*. If not, RoboHalt chooses to ask the next question. If RoboHalt is unable to make a decision after five predictions, it makes a decision of passing or blocking the call supported by the majority labels.

3.7.2 Transcriber

This component transcribes the responses from the caller for other modules, which then determine if the responses are appropriate or not. Moreover, when the user is notified of an incoming call, the transcript of the conversation between the caller and the virtual assistant (VA) is shown on the screen for additional context. For blocked calls, call context is saved for the user to review later. We chose Google Cloud Speech API [6] because it has high accuracy and transcription can be performed in minimal time, unlike Kaldi [55] and Mozilla deep speech [4].

3.7.3 Silence Detector

This module is invoked by the controller when RoboHalt asks the caller to hold. It randomly picks a hold time, t_s, ranging between five to ten seconds, asks the caller to hold and comes back to the caller to continue the conversation after t_s seconds. The Silence Detector module transcribes everything said by the caller during the t_s seconds and calculates the average number of words said per second, w_{ps}. If w_{ps} is less than the threshold θ_s, the response is labeled as appropriate by the Silence Detector module and vice versa. We set θ_s in the following way. We calculate the average number of words spoken per second, aw_{ps}, from our collection of pre-recorded robocall recordings and set $\theta_s = (t_s \times aw_{ps})/2$.

3.7.4 Context Detector

This module is invoked after the virtual assistant says "How can I help you?" To identify *not appropriate* responses, we build a robocall campaign detector using a dataset of phone call records collected at a large phone honeypot provided by a commercial robocall blocking company. This dataset contains 8,081 calls (having an average duration of 32.3 seconds) coming into a telephony honeypot during April 23, 2018 and May 6, 2018. The audio recordings of these calls are used to generate call transcripts. Using LSI topic modeling, we extract 30 topics from our corpus of transcripts where each topic represents a spam campaign. We construct a similarity matrix by computing the cosine similarity between each transcript. We then convert the similarity matrix into a distance matrix by inverting the elements of the similarity matrix. We then performed DBSCAN clustering on the distance matrix. At the end of this, 72 clusters were created where each cluster represents groups of highly similar transcripts of robocalls. We then take one representative robocall from each cluster and calculate the vector representations by projecting the robocall messages onto the pre-computed LSI topic model. To classify a response from a user, the Context Detector, after pre-processing the text, calculates the vector representation by projecting the response onto the pre-computed LSI topic model. It then computes the cosine similarity of the user response with the pre-computed 72 roboball feature vectors. If the cosine similarity is greater than a threshold, C_s, it is labeled as an inappropriate response. In other words, if the content of the caller response matches with any previously known robocall message, it is labeled as a not appropriate response; otherwise it is labeled as an appropriate response.

3.7.5 Elaboration Detector

A follow up question of "How can I help you?" is "Tell me more about it". This module determines if the response provided by the caller for this follow up question is appropriate. There has been a lot of work on text summarization in the area of natural language processing [24, 38, 46]. However, a large amount of data and a complex architecture is required to train models accurate enough to be useful in real-life scenarios. Therefore, to keep this component simple, we take the following approach. We count the number of words in the caller's response. If the number of words is higher than the number of words in the previous response, it is labeled as an appropriate response and vice versa. We understand this is a simple approach but it is important to note that RoboHalt uses labels from multiple modules to make a final decision. Hence mislabeling by an individual component should be compensated by the other components.

3.7.6 Relevance Detector

This module determines whether the response from the caller is an appropriate response for the Relevance questions asked by RoboHalt. We build a binary classifier that, given a (question, response) pair, labels the response *appropriate* if the response is a reasonable answer to the question selected by the controller and *not appropriate* if not. Human callers are expected to provide *appropriate* responses and robocallers are expected to provide *not appropriate* responses.

To build such a classifier, a large labeled dataset of relevant question answer pair during phone conversations would be required. However, due to the unavailability of such datasets, we use the "Fisher English Training Part 2, Transcripts" dataset [5], which was collected to build Automatic Speech Recognition(ASR), and modify it to fit our use case. This dataset represents a collection of conversational telephone speech. Under the Fisher protocol, a large number of participants each make a few calls of short duration speaking to other participants, whom they typically do not know, about assigned topics.

We tailor the Fisher dataset to build our Relevance Detector model. We take the conversation between each speaker pair (speaker A and B), convert it into (comment, response) pairs and label them as appropriate. To generate the irrelevant examples, for each comment by speaker A we randomly pick a response which is not the response provided by the speaker B from the Fisher dataset and label that pair as not-appropriate. As a result, we generated 300,000 appropriate and 300,000 not-appropriate (comment, response) pairs to construct our dataset. We then perform sentence embedding on each data point to convert the text into a vector. Similar to word embeddings (like Word2Vec [48], GloVE [53], Elmo [54] or Fasttext [27]), sentence embeddings embed a full sentence into a vector space. We use Infersent [35] to perform sentence embedding on our data points. InferSent is a widely used sentence embedding method that provides semantic sentence representations. The (comment embedding, response embedding) pairs are then passed to the binary classification model we built.

We used a Multilayer Perceptron (MLP) as our base model. The train, validation and test accuracy of the base model is 83%, 70% and 70% respectively. To test with robocalls we take the following approach. We treat the questions asked by RoboHalt as a comment and the transcripts from robocall recordings as a response. However the base model performs poorly when tested with robocalls. Since the model is not specifically trained to recognize robocalls, the testing accuracy decreases in this case.

We further fine-tune our base model to specifically recognize robocalls and legitimate (human) calls. We assume that human callers will be able to provide appropriate responses to the questions whereas robocallers will not. Therefore we label (question, robocall response) pairs as *not appropriate* and (question, human response) pairs as *appropriate* to fine tune our base model. To generate our *not appropriate* responses, we use the dataset of robocalls described in Section 3.7.4. We take the first 30 words (as we let each response to be of at most 20 seconds) from each robocall transcript and pair it with both relevance questions to form our *not appropriate* responses. In this way we get 67 unique (question, robocall response) pairs. Since this dataset is too small to fine tune a model and the number of unique robocall messages is limited, we perform data augmentation on the 67 unique robocall responses. For each robocall response, we generate two more augmented texts using the techniques in [31]. To generate the appropriate pairs, for each question from the Relevance question pool we use Quora [9] to collect appropriate human responses to these questions. We augment the (question, human response) pairs in the same way. Upon generating the appropriate and not appropriate pairs we generate the sentence embedding pairs, which are then passed to fine tune our base model. Table 2 shows the test accuracy of the fine-tuned model.

Table 2: Relevance Detector Results

Accuracy	First 10 words	First 20 words	First 30 words	First 40 words
Overall	91.02%	97.18%	98.32%	97.21%
Robocall	92.5%	98.46%	100%	98.49%
Human Call	87.23%	93.62%	93.62%	93.62%

3.7.7 Repetition Detector

To build a repetition detector, a labeled dataset of repetition pairs of human responses are required. However due to the unavailability of such datasets, we collect segments of conversation (current response, last response) pairs from Lenny [62] recordings. Lenny is a bot (a computer program) which plays a set of pre-recorded voice messages to interact with human callers. Lenny is surprisingly effective in keeping the conversation going for tens of minutes. There are more than 600 publicly available call recordings where Lenny interacts with human spammers (for e.g. telemarketers). During the conversation, Lenny asks the callers to repeat themselves multiple times. Among 600+ publicly available call recordings, we randomly selected 160 call recordings and manually transcribed the parts where the callers have repeated themselves. Specifically, we created 160 (current response, last response) pairs and assigned them with the "appropriate" label. Since the telemarketers talking to Lenny are humans, when asked to repeat themselves, they provide a semantically correct answer, if not the exact repetition of their last statement. We expect most legitimate human callers to behave in a similar way. Robocallers on the contrary are not expected to provide an appropriate response when asked to repeat. To generate our "not appropriate" pairs, for each last response, we randomly pick a response from the Lenny transcripts which is not an appropriate repetition. In this manner, we generate 160 "not appropriate" pairs. We extract the following three features from the data points generated:

Cosine similarity: We calculate the cosine similarity between current response and last response.

Word overlap: Upon removing stop words and punctuation,

Table 3: Repetition Detector Results

	Test Accuracy	Robo Test Accuracy	FPR
SVM	94%	82%	12.5%
Logistic Regression	93.6%	85%	12.5%
Random Forest	95%	86%	6.25%
XG Boost	95%	85%	9.4%
Neural Network	93.7%	83%	12.5%

we calculate the number of words that overlap between current response and last response.

Named entity overlap: In information extraction, a named entity is a real-world object, such as persons, locations, etc., that can be denoted with a proper name. We use Spacy [13] to extract the named entities and then calculate the number of overlapping named entities between current and last response.

These simple yet effective features allow us to determine if a $statement_1$ is a semantic repetition of $statement_2$ without using resource intensive machine learning models. We train 5 different classifiers using the above mentioned three features. Table 3 shows the test accuracy and false positive rates for each classifier. Table 3 also shows how the classifier performs on the robocall test set. To generate the robocall test set we take 72 representative robocalls messages and generate (current response, last response) pairs by setting the first sentence and second sentence from the robocall messages as current response and last response respectively. In this dataset, none of the current responses are semantic repetitions of last responses and hence these pairs should be labeled as "not appropriate". Since Random Forest has the highest robocall test accuracy and lowest false positive rate, it is chosen to be the most suitable classifier for the RD module.

3.7.8 Name Recognizer

This module is invoked by the controller when RoboHalt asks the caller to provide the name of the callee. Name Recognizer (NR) determines whether the correct name was spoken. The user is allowed to set the name(s) that should be accepted as correct by RoboHalt. Users may set multiple correct names as a valid recipient of phone calls coming to their device. The backbone of the NR module is a keyword spotting algorithm. In our implementation we use Snowboy [12] to recognize the name. We treat Snowboy as a blackbox, which when provided with 3 audio samples, creates a model to detect the keyword. We embedded the downloaded trained model with the NR module to recognize the correct name(s). Since Snowboy does not provide a confidence score, we set the accuracy of Snowboy (0.83) as its fixed confidence score for every label.

3.7.9 Affirmative/Negative Intent Recognizer

A follow up question of "Who are you trying to reach?" is asking the caller to confirm the name. RoboHalt does this in two ways, asking the caller to confirm by saying the correct name or saying an incorrect name. For example, if the correct name is Taylor, the virtual assistant will say, "Did you mean Taylor?" and expect an affirmative answer from a human caller. An alternative question is asking the caller to confirm the name by intentionally saying an incorrect name, such as, "Did you mean Tiffany?". In this case RoboHalt expects a negative answer from an human caller. Based on the question and the expected response from the caller, Affirmative/Negative Intent Recognizer labels a response from the caller as inappropriate or appropriate. To detect if the response is affirmative or negative, we manually compile a list of affirmative (e.g. yes, yeah, true etc.) and negative (e.g. no, not etc.) words. If an affirmative answer is expected and the caller's response contains any of the affirmative words, it is labeled as an appropriate response. Similarly, if a negative answer is expected and the caller's response contains any of the negative words, it is also labeled as an appropriate response. All other cases are labelled as inappropriate responses.

3.7.10 Amplitude Detector

The Amplitude Detector module is invoked after RoboHalt asks a caller to speak up. If the average amplitude of the response audio is higher than the caller's previous response by an error margin, Amplitude Detector labels it as an appropriate response and vice versa.

4 Evaluation

In this section, we report results of the evaluation we performed to assess the accuracy of decisions made by RoboHalt. We also conduct a red team style security analysis and discuss attacks the red teams crafted in an attempt to evade RoboHalt. By performing this analysis, we demonstrate RoboHalt's effectiveness against robocalls in our threat model. We also discuss a user study that was conducted to evaluate its usability.

4.1 Security Analysis

We first report the effectiveness of RoboHalt against current robocalls and a baseline attack in which the robocaller randomly responds to its questions. We recruited a group of graduate students with varying expertise in security analysis who play the role of a robocaller. We had three red teams, Red Team A, B and C, working independently of each other. Red Team A consisted of two MS students and one PhD student. Red Team B consisted of one MS student with expertise in voice-based attacks. Red Team C consisted of a PhD student who is an expert security tester. Red teams A and B attempted black-box attacks. They were provided with unlimited access to RoboHalt but not its implementation details. Red Team C had knowledge of how the system works and crafted a grey box attack discussed in the later sections. All red team attacks were automated and no human interaction was present when making calls to RoboHalt. Our red teams mainly crafted three types of attacks discussed below.

Current Mass Robocall Attacks: We define mass robocalls as automated calls made by attackers who don't have any specific information, such as name of their target victim. We provided Red Team A with 72 representative robocall recordings from a corpus of 8000 real robocalls. The sample

was selected using the method described in Section 3.7.4. Red Team A used the robocall recordings in various scenarios to craft the black-box attacks on RoboHalt.

In the first experiment, Red Team A used the robocall recordings and played them as soon as RoboHalt picks up the call. Each recording was played two times to make two independent calls. Red Team A found that 95% of the 144 mass robocalls were successfully blocked. In the second experiment, Red Team A used the same robocall recordings. However, they did not play the recordings as soon the call is picked up. Instead the recordings were played after RoboHalt finishes saying the greetings and asks the first question. This was done to simulate the evasion technique many current robocallers use, where they speak only after being spoken to, once the call is picked up. Red Team A similarly used every recording two times to make two independent calls and found that 94.5% of the 144 mass robocalls were successfully blocked. Red Team A then conducted additional experiments by increasing and decreasing the playback speed of the audios and did not find any difference in RoboHalt's performance. Moreover, they played the shorter robocall recordings in a loop and found similar success rate. Red Team A reported that they could not find any pattern of current mass robocalls which can be exploited to attack RoboHalt.

Random Response Robocalls Attack: An attacker can learn the questions asked by RoboHalt by interacting with it multiple times. We measure the effectiveness of RoboHalt against an adversary who randomly chooses a response to the asked questions. To build this random response adversary, we take the following approach. We create appropriate responses for the questions from the question pool. For example, we create the response "I am fine." for the question "How are you doing?". We assume that the random response adversary does not craft targeted attacks, hence, does not know the correct name. Therefore, we create the response, "I am trying to reach Mike." as an answer to the question "Who are you trying to reach?", where Mike is a common name in the US. As an appropriate response for the *Hold* question, the adversary randomly chooses to pause for 5 to 10 seconds. Also, we don't create any response for the *Repetition* and *Speak up* questions as these responses are related to the previous response. Once the response pool is created, the adversary makes a call to RoboHalt and then randomly chooses the number of questions to answer. We assume that the adversary knows RoboHalt asks 2 to 5 questions and chooses between 2 to 5 responses.

The adversary set up by us made 300 calls to RoboHalt where the followings choices were made randomly by the adversary during each call: (i) the number of questions to answer, (ii) the response to a certain question, and (iii) the time interval between each response. 273 out of 300 calls were blocked, yielding a blocking rate of 91% for attackers who randomly guess responses to RoboHalt questions.

Baseline Evasive Robocall Attacks: We define baseline evasive robocallers as attackers who use information extracted from interacting with RoboHalt to craft black-box attacks. Red Team B was asked to launch the baseline evasive robocall attacks. To do this, Red Team B interacted with RoboHalt several times to extract information about which questions are asked, the order and frequency of the questions. Based on this, they created audios of appropriate responses to the questions. For the Name Recognizer question, they used the callee names "Jessica" and "William". For the questions "What's the weather like?" and "How are you?", they used "Great" as a general answer. For the Context Detector question, the sentence pattern they used is "I want to talk to CALLEE NAME". Red Team B realized that once the call is picked up, RoboHalt will either ask the caller to hold briefly or ask questions directly. They created several audios where some start with a pause and others start with the responses to the questions. Red Team B sorted the answers in different orders and created 10 different audios. They used each audio 10 times to make 100 calls in total and found that RoboHalt was successful in blocking 84% of the evasive attacks. The only time the attack was successful was when the order of the responses aligned perfectly with the questions which happened 16% of the times. Since the attacks were only successful by random chance, it can be said that RoboHalt is effective against evasive attacks.

Targeted Robocall Attacks: In targeted attacks, we assume that robocallers know the name of the callee and use that information to craft attacks. We conducted a limited scale targeted attack on RoboHalt. First, we appended the audio, "I want to reach Taylor" to the beginning of all of the 72 representative robocall recordings. We used these newly created targeted robocall recordings and made two calls for each recording to RoboHalt. Out of the 144 targeted robocalls we made, 87% of the calls were successfully blocked by RoboHalt.

Red Team B also crafted a targeted attack on RoboHalt. A red team member used the same audio recordings used in the evasive attack discussed above. For conducting targeted attacks, we shared the correct name of the callee, "Taylor", with red team B. Therefore, they used the correct name "Taylor" instead of "Jessica" and "William" for the Name recognizer question and kept the rest of the audios same as the ones they used during the evasive attacks. Similar to the evasive robocall attacks, they used the 10 audio recordings 10 times to make 100 calls in total and found that RoboHalt was successful in blocking 82% of the targeted attacks.

Successful Attacks: Red team B found that if a short generic response such as "I want to talk to Jessica" is repeated for every question, in most cases RoboHalt fails to block the call. This occurs because such a short generic response is applicable for many questions RoboHalt asks in the beginning of the call, such as "How can I help you?", "Can you please hold?", "Can you repeat?". This attack can be countered by adding a simple check to see if the caller is saying the same thing over and over again.

NLP Equipped Evasive Robocall Attacks: Red Team C crafted a grey box attack on RoboHalt using NLP techniques. The red team member had knowledge of the categories of questions asked and how the individual components and RoboHalt

as a whole work. Hence, we define this as a grey box attack. To craft this attack, red team C used a Python websocket client to automatically initiate calls with the RoboHalt web app and download all of the audio files returned from the VA server. After about 30 minutes of continuous running, Red Team C was able to harvest a large portion of the audio prompts by the VA. The red team member then ran all the prompts through AWS Transcribe [3] to get the transcripts of the audio files and manually clustered the transcripts and labeled the clusters based off of the categories mentioned in Table 1 (Hold, Context Extractor, Name Recognizer, etc). After this, short and plausible text-based responses were crafted for each cluster of questions (such as "I am fine", "The weather is nice today," etc.). They then ran the text responses through AWS Polly [2] to build a database of corresponding voice responses. After this initial learning phase, calls were attempted. During the attack, a call is initiated and the questions from RoboHalt are transcribed in real-time using AWS Transcribe. After generating the transcript, the attacker labeled the given question with their respective question category, based on keywords and phrase matching. Finally, the attacker returned the appropriate response for each question from the database of voice responses. The red team member made 100 such calls and was able to evade RoboHalt 67% of the times. The attack was not successful the remaining 33 times because the attacker was not able to provide the appropriate response within reasonable time due to latency caused by the transcription service. The red team member then crafted a second attack where the attacker selects an appropriate response as soon a keyword/phrase is matched, instead of waiting for the whole transcription to finish. Using this optimization, the red team member was able to achieve a 96% success rate. The attack was not successful the remaining 4% of the times because of transcription error caused by background noise.

Countermeasures: Red team C noticed that the success of the attack heavily relied on the quality of the transcription. Adversarial examples have been explored in the research community where inputs can be specially designed to cause a machine learning algorithm to produce a misclassification [26]. Recent work has showed that any given source audio sample can be perturbed slightly so that an automatic speech recognition (ASR) system would transcribe the audio as any different target sentence [30, 57]. Hence, such techniques could be used to perturb the audio prompts by the VA where the audio is perceived by humans correctly while being incorrectly transcribed by ASR systems. Moreover, since robocall hit rates are extremely low, robocallers have to make a massive number of calls for their scams to be profitable. Therefore, adding advanced NLP capabilities at such a scale will add significant cost and complexity to their infrastructure.

4.2 Comparison with Google Call Screen

Google Pixel phones include an automatic call screening feature that allows Google Assistant to pick up calls on the user's behalf and filter out spam calls. Since the goal of this feature is similar in principle to our RoboHalt's goal, we conducted an experiment to better understand how Google Pixel's call screening works and to evaluate its performance in blocking robocalls. To the best of our knowledge, Google Pixel's call screening features is a closed-source system with no detailed documentation available to the public. Therefore, we performed an initial exploratory black-box analysis to understand how the system works. After some experimentation, we suspected that the spam call filtering functionality relies primarily on caller ID blocklists. To test this hypothesis, we devised an experiment in which both benign and robocall-like calls were made to a Google Pixel device from both legitimate and known-spam phone numbers. We describe these experiments below.

Calls from legitimate phone numbers: Since no ready-to-use phone safelist is available, we collected a set of phone numbers that can be considered as legitimate (i.e. non-spam) by crawling the YellowPages.com phone book. We gathered around 200 phone numbers listed across 5 different major US cities and 5 different business categories including doctors, plumbers, restaurants, etc. The assumption is that the vast majority of these businesses are legitimate entities unlikely to engage in phone spam activity. We used Spoofcard [14] to spoof these benign phone numbers. For each of the 200 phone numbers, we made 3 calls to our own Google Pixel, each simulating one of 3 different scenarios: *benign conversation*, *robocall message*, and *silence*. Each of the calls we made using the benign phone numbers were picked up by the Google Assistant, as expected, which asked for the purpose of the call. To simulate *benign conversation*, we randomly selected and played an audio recording from a set of pre-recorded generic phrases, such as "I am calling to make an appointment". For *robocall message* we randomly selected and played a robocall recording from a set of 72 representative robocall recordings chosen from a honeypot containing 8000 real robocall recordings (same dataset used to analyze mass robocall attacks in 4.1). For the last scenario, we stayed silent after the purpose of the call was asked. We found out that none of these calls were blocked by the Google Assistant, and were instead forwarded to the user along with the transcription of the recordings. For the silent calls, no transcription was shown. It is important to notice that none of the 200 robocalls made from benign phone numbers were blocked by Google Assistant, even though the content of the calls was obviously spam. The phone call records were not shared with Google's server to ensure low risk for the benign phone numbers we used.

Calls from spam phone numbers: To confirm our hypothesis that Google Call screen only relies on the caller ID to block unwanted calls, we then conducted the same experiment with known spam phone numbers. To gather a list of known spam phone numbers, we extracted the most complained about 200 phone numbers reported by the FTC during the first week of March 2022, and spoofed these numbers to make calls to our Google Pixel during the first week of April 2022. We conducted the same experiment with 3 call scenarios of *benign conversa-*

tion, *robocall message*, and *silence* for each of the 200 spam numbers. We obtained the exact same result across the 3 different scenarios. In each scenario, 52.2% of calls were blocked, 31.5% calls were forwarded to the user with a spam warning and 16.5% of the calls were forwarded without any warning.

Conclusion: The above results demonstrate that Google Pixel's call screening feature makes decisions on blocking or forwarding a call based primarily on the caller ID, independently of the content of the call. Therefore, robocalls with spoofed benign or previously unseen caller IDs can evade the Google Assistant without any additional effort. On the other hand, our RoboHalt system does not solely rely on the caller ID and can successfully block robocalls even when they originate from a benign or unknown caller ID.

4.3 Usability Study

To explore the usability of RoboHalt, we conducted an Institutional Review Board (IRB) approved user study. In the following, we first describe the study setup, its participants and then discuss the results.

Study Setup: Our study participants were 40 users who were sampled from a population of grad and undergrad students, their parents and other family members. All participants were required to be above 18 years old and fluent in English. 20% of our users were aged between 18-25 old, 30% were aged between 25-35 years old, 24% were aged between 35-50 and the rest were above 50 years old. 45% of users were female and the rest were male. We collected information about phone usage and previous experience regarding robocalls from our users. 56% of our users use Android and the rest use iPhone. Moreover, 35% of our users reported that they use some sort of call blocking applications (e.g. Truecaller, Youmail etc.). 82% of the users who use call blocking applications reported that they never pick up calls labeled as suspicious/spam. Regarding their previous experience of receiving robocalls, 39% of our users reported that they receive one or more robocalls every day and 53% of the users receive one or more robocalls every week. However, the majority of these users don't use call blocking applications and are unprotected from robocallers.

We briefed the participants about the experiment and explained the purpose of RoboHalt. Due to the ongoing pandemic, it was not safe to conduct an in-person user study. To ensure that the user study avoids physical contact, we hosted RoboHalt on a AWS server which can be accessed via a web interface. We provided a URL to users which directed them to a web interface in their browser. By clicking on a Start Call button, users initiate an interaction with RoboHalt. The users are able to talk to RoboHalt through the microphone of their own device. In this study, all users played the role of a caller and made four calls on various given topics. Such a call took at most one minute. Upon completing each call, users were provided with a set of survey questions that focus on evaluating the user experience. At the end of the user study, each user was asked three generic questions about their overall experience with RoboHalt.

We performed two experiments, one where the callers know the name of the callee and one where the caller doesn't know the name of the callee. During the first experiment, we preset the correct name to be *Taylor* instead of having each user set a name. We make this choice because the purpose of the user study is to get insights about call experience in the presence of a phone assistant, rather than testing the accuracy of the keyword spotting algorithm. We recruited 30 out of 40 users for this experiment and provided the following four topics to make the four simulated phone calls: (i) make a movie plan, (ii) make an appointment, (iii) plan a cruise vacation, and (iv) make a call about car warranty. The topics are selected such that it is natural for a phone call setting and common in real life scenarios. We choose the last two topics to overlap with robocall topics (free cruise and car warranty).

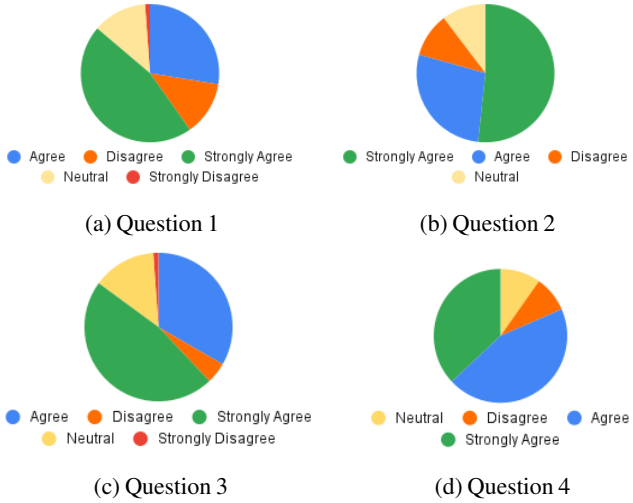
During the second experiment, the caller is either given an incorrect name or no name at all. We recruited the remaining 10 users for this experiment. Since in a real life scenario most legitimate human callers know the callee's name, we have a lower number of users playing the role of a caller who does not know the callee. The following are the call topics of the three calls made by the callers during the experiment: (i) play role of telemarketer to sell a computer, (ii) conduct a survey on robocalls, (iii) call friend Robert to meet for lunch, and (iv) call Jordan about car warranty. The caller is not given any name in the first two topics and is given an incorrect name in the last two topics. Since RoboHalt requires human-like interaction to forward a call, it is expected that the calls will be forwarded even if the caller doesn't know the correct name. This ensures that calls from first-time legitimate callers who don't know the name of the callee are not blocked.

4.3.1 User Study Survey Results

In this section we present the results obtained from the user responses to the survey questions. After each call the users were asked the following four questions. The user responses to these questions are summarized in Figure 3

- Question 1: The conversation with RoboHalt felt natural.
- Question 2: I was able to answer the questions asked by RoboHalt without difficulty.
- Question 3: The number of questions I answered was acceptable.
- Question 4: The time I spent interacting with RoboHalt before my call was forwarded/blocked is acceptable.

Figure 3(a) demonstrates that most of the users reported that the conversation with RoboHalt felt natural. Only 13.5% of the times users reported that the conversation did not feel natural. We collected additional feedback about their experience and asked for suggestions from the users during the study. The users who felt that the conversation was not natural, mentioned that when they responded to how they were or how the weather was, they also asked the virtual assistant the same question. However, RoboHalt did not respond to their question and moved on to ask the next question. It is understandable because



(a) Question 1 (b) Question 2

(c) Question 3 (d) Question 4

Figure 3: User Responses

RoboHalt is not designed to respond to the caller's questions. This feedback from the users was useful and could be incorporated in future work. Figure 3(b) demonstrates that most of the users (80%) were able to answer the questions asked by RoboHalt without difficulty. The additional feedback collected from our users showed that 4 out of 50 users mentioned that they felt unfamiliar with the system and had difficulty answering the questions during the first call. However, after making one or two calls, they became familiar and were able to answer the questions with ease. We also asked users if the number of questions they had to answer was acceptable. Only 5% of users reported that it was not acceptable (Figure 3(c)). We found out that these users had to answer five questions before a decision about their call was made. Moreover, we computed the number of questions RoboHalt asked during its interaction with our users and found that in 70% of the cases, RoboHalt made a decision by asking up to three questions. Hence, 80% of the users reported that the number of questions they had to answer was acceptable. Figure 3(d) further shows that only 8.7% of users felt that the time they spent interacting with the virtual assistant before their call was forwarded/blocked is not acceptable.

After the end of the experiment, each user was asked the following three questions regarding their overall experience with RoboHalt. The user responses to these questions are summarized in Figure 4.

- Question 5: It was easy to interact with the RoboHalt.
- Question 6: I felt comfortable with RoboHalt intervening phone calls.
- Question 7: I think I would like to use a system equipped with such a virtual assistant frequently.

As shown in Figure 4(a), 83.8% of user study participants reported that it was easy to interact with RoboHalt. Moreover, 85% of participants reported that they felt comfortable with RoboHalt intercepting the calls (Figure 4(b)) and 88.7% reported that they would like to use the system frequently (Figure 4(c)).

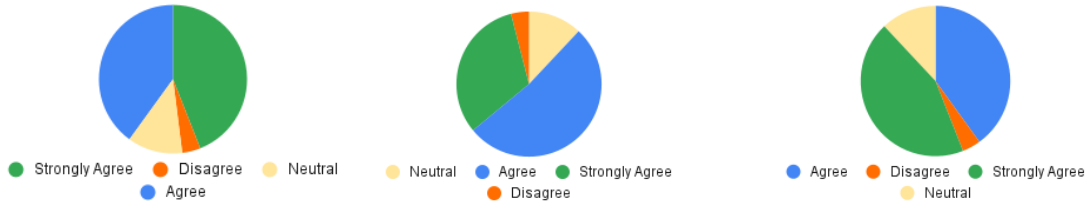
4.3.2 RoboHalt Performance During User Study

During the user study, we computed the number of questions users had to answer and the time they spent interacting with RoboHalt before a decision about the call was made. We found that for 30% calls, users answered two questions. Also, users answered three questions for 40% calls. In only 15% of the cases, RoboHalt asked five questions before a decision was made. Moreover, users spent an average of 23.8 seconds with a median of 19 seconds interacting with RoboHalt. The maximum latency a user experienced before a decision about their call was made is 90 seconds. We further investigated which questions performed better during the user study. We found that most users were asked the *Hold* question during their interaction and the Silence Detector module labeled the responses correctly every time. The *Relevance* questions also performed well during the user study, which were only mislabeled for 7% of the calls. Moreover, *Context Extractor* and *Repetition* questions incurred an error rate of 12% and 13%, respectively. We noticed that mislabeling in *Repetition* only occurred when the statement from the user produced an incorrect transcription. During the user study, the Name Recognizer performed the worst, yielding an error rate of 23%. This is expected since we utilized Snowboy's personal model in our implementation, where the model is trained using only one person's audio recordings. However, since RoboHalt does not heavily rely on one specific module, it can correctly label human callers despite mislabeling by individual modules. We further describe the effectiveness of each module in the following sections.

4.4 Measuring False Positives

To compute false positives, we use the data collected during the user study. 40 users made 160 calls in total and only 13 calls were blocked, yielding an overall false positive rate of 8.1% . We further investigated calls that were mistakenly blocked and found that in 10 out of the 13 (76.9%) calls, the user kept silent and didn't answer all the questions, hence the calls were mislabeled. This is expected because RoboHalt blocks calls when human-like interaction is not detected. We also investigated if not knowing the right callee name incurs a higher false positive rate. Among the 40 calls made by users who did not know the correct name, 3 calls were blocked. Hence the false positive rate (7.5%) remained similar. Thus, RoboHalt is not heavily dependent on a caller knowing the name of the callee. We also investigated if call topics had any effect on their blocking. Two of our call topics in the user study overlapped with robocall topics (free cruise and vehicle warranty). 70 calls were made by users regarding these topics and only 6 calls were blocked, yielding a false positive rate of 8.6%. It is important to note that these 6 calls are the same calls where the users did not respond to all questions. Therefore, it can be concluded that RoboHalt is not biased towards specific keywords and does not block calls when it detects such keywords.

While 8.1% false positives may seem high for security applications, we find that most of the false positives occur when callers do not respond to the questions asked by RoboHalt.



(a) Question 5 (b) Question 6 (c) Question 7

Figure 4: User Responses (contd.)

We found that in all the cases where users kept silent, it was their first or second time interacting with the system. Since each user made 4 calls, users who were initially silent became familiar with the system by the third call and answered all questions. If we only consider the third and fourth calls, false positive rate drops to 5% (4 out of 80 calls). Thus, as users become accustomed to RoboHalt, we expect the false positive rate to further decrease. Also, it is important to note that false positives only impact calls from numbers that are not in the callee's contact list. Given that false positive calls are redirected to voicemail (which is similar to the callee not picking up the call), the user will be notified of the call with a transcript. Since users often do not pick up calls from unknown caller IDs, the cost of a RoboHalt false positive is expected to be low.

4.5 Effectiveness of RoboHalt's Components

RoboHalt uses an ensemble of ML models. We explore the performance of the 8 sub-modules during RoboHalt's handling of both legitimate and robocalls. For legitimate callers, we utilized the data collected during our user study and observed how each individual component performs in labeling the responses (see Section 4.3.2). For interaction with robocallers, we made 72 representative robocalls to RoboHalt and measured the performance of each individual sub-module in labeling them. Table 4 summarizes the accuracy of each sub-module. Silence Detector (SD) always labels human responses very accurately, however it incurs a lower accuracy for robocallers. This is caused by short length robocalls where the robocaller stays silent when the *Hold* question is asked. Since the SD module finds silence as a response, it mislabels the robocaller. Repetition Detector incurs a higher error rate for legitimate callers because of transcription errors. However, it has a higher accuracy for robocallers, as they do not repeat themselves. Name Recognizer has a 100% accuracy for robocallers, since robocall messages didn't include the correct name. However, it has lower accuracy for legitimate callers. It is important to note that, RoboHalt's ensemble design ensures that it does not solely rely on one component. Therefore, its overall performance remains high even when individual components mislabel callers.

Table 4: Accuracy of Sub-modules

	Legitimate Caller	Robocaller
Silence Detector	100%	85%
Context Detector	88%	89%
Elaboration Detector	98%	87%
Relevance Detector	93%	90%
Repetition Detector	87%	93%
Name Recognizer	77%	100%
Aff./Neg. Detector	97%	100%
Amplitude Detector	92%	84%

5 Discussion and Limitations

Our evaluation shows RoboHalt is effective against both current mass robocallers and more sophisticated robocallers that might emerge in the future. However, there are some limitations. RoboHalt cannot block calls from an AI-equipped robocaller who comprehends its questions and responds accordingly. However, deploying such AI-enabled robocallers at scale can be expensive and resource-intensive. RoboHalt is also not designed to protect against unwanted human callers. Therefore, spam campaigns that use human callers cannot be stopped by RoboHalt. Again, hiring human callers can also be expensive.

RoboHalt does not intercept calls from *safelisted* callers. It may be possible for robocallers to spoof a phone number from the victim's contact list to bypass our defense. However, to craft such an attack, robocallers need to obtain a large number of contact lists and target each individual user. Since robocall conversion rates are extremely low, robocallers have to make a massive number of calls to be profitable. Such targeted attacks at scale will increase their cost significantly.

Our user study was limited to 40 users. We hope the results of the study would be applicable to the general population but further studies may be needed. Our experiments were conducted with a specific name set as the correct name. Since keyword spotting algorithm evaluation is out of our scope, we did not conduct experiments with a broader range of names. Also, RoboHalt is not designed to answer questions from callers and the natural flow of the conversation may not always be maintained, as was pointed out by some of our users.

6 Conclusion

RoboHalt is a smartphone virtual assistant that aims to automatically detect and block robocalls before they reach the user. We presented the design of RoboHalt and explored if it can be effective against mass, targeted, and evasive robocallers. We developed a proof-of-concept system, hosted it on an AWS server and conducted a user study to assess its usability. The results from the user study demonstrate that RoboHalt can preserve user experience and at the same time keep the false positive rate low. We recruited multiple red

teams with varying levels of expertise who crafted attacks against RoboHalt. Our red teams reported that 95% of the mass robocalls were successfully blocked. The red teams also tried more sophisticated robocalls and found that RoboHalt was successful against 82% of evasive attacks.

In future work, we plan to explore if RoboHalt can ask questions dynamically, instead of choosing from a fixed question pool. Similar to a chatbot [41], RoboHalt can carry a more natural conversation with the caller. The responses can be created using text generation. We anticipate that such a method would further improve robustness and enhance user experience.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments and suggestions on how to improve this paper, and the red team members for their valuable contributions to the evaluation of our system. This material is based in part upon work supported by the National Science Foundation (NSF) under grants No. CNS-1514035 and CNS-1514052. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] At&t. https://www.att.com/features/security-apps.html. Accessed: 2021-09-19.

[2] Aws polly. https://aws.amazon.com/polly/. Accessed: 2022-05-29.

[3] Aws transcribe. https://aws.amazon.com/transcribe/. Accessed: 2022-05-29.

[4] Deep speech. https://github.com/mozilla/DeepSpeech. Accessed: 2020-09-26.

[5] Fisher english training part 2, speech. https://catalog.ldc.upenn.edu/LDC2005S13. Accessed: 2022-05-29.

[6] Google speech. https://cloud.google.com/speech-to-text/. Accessed: 2020-09-26.

[7] Hiya. https://www.hiya.com/. Accessed: 2021-09-19.

[8] Nomorobo. https://www.nomorobo.com/. Accessed: 2021-09-19.

[9] Quora. https://www.quora.com/. Accessed: 2022-05-29.

[10] Secure Telephony Identity Revisited, IETF Working Group. https://tools.ietf.org/wg/stir/. [accessed: 2021-09-19].

[11] Shaken/Stir. https://transnexus.com/whitepapers/understanding-stir-shaken/. [accessed: 2021-09-19].

[12] Snowboy. https://github.com/Kitt-AI/snowboy. Accessed: 2022-05-29.

[13] Spacy. https://spacy.io/. Accessed: 2020-09-26.

[14] Spoofcard. https://www.spoofcard.com/. Accessed: 2022-05-29.

[15] Truecaller. https://www.truecaller.com/. Accessed: 2021-09-19.

[16] Verizon. https://www.verizon.com/support/residential/homephone/calling-features/stop-unwanted-calls. Accessed: 2021-09-19.

[17] Youmail. https://www.youmail.com/. Accessed: 2021-09-19.

[18] Combating spoofed robocalls with caller id authentication. https://www.fcc.gov/call-authentication, Apr, 2021. [accessed: 2021-09-19].

[19] Perspectives: Why we're still years away from a robocall-free future. https://www.cnn.com/2019/04/10/perspectives/stop-robocalls-shaken-stir/index.html, April 10, 2019. [accessed: 2021-09-19].

[20] Shaken/Stir CNN. https://www.cnn.com/2021/07/02/tech/robocall-prevention-stir-shaken/index.html, July 02, 2021. [accessed: 2021-09-19].

[21] U.S. Phones Received Over 3.9 Billion Robocalls in April, Says YouMail Robocall Index. https://www.prnewswire.com/news-releases/us-phones-received-over-3-9-billion-robocalls-in-april-says-youmail-robocall-index-301540784.html, May 05, 2022. [accessed: 2022-05-11].

[22] Neighbor scam moves on to spoofing just area codes. https://hiya.com/blog/2018/05/23/neighbor-scam-moves-on-to-spoofing\-justarea-codes/, May 23, 2018. Accessed: 2021-09-19.

[23] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.

[24] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*, 2017.

[25] Morvareed Bidgoli and Jens Grossklags. "hello. this is the irs calling.": A case study on scams, extortion, impersonation, and phone spoofing. In *2017 APWG*

Symposium on Electronic Crime Research (eCrime), pages 57–69. IEEE, 2017.

[26] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[27] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[28] Elie Bursztein, Romain Beauxis, Hristo Paskov, Daniele Perito, Celine Fabry, and John Mitchell. The failure of noise-based non-continuous audio captchas. In *2011 IEEE symposium on security and privacy*, pages 19–31. IEEE Computer Society, 2011.

[29] Elie Bursztein and Steven Bethard. Decaptcha: breaking 75% of ebay audio captchas. In *Proceedings of the 3rd USENIX conference on Offensive technologies*, volume 1, page 8. USENIX Association, 2009.

[30] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *25th USENIX security symposium (USENIX security 16)*, pages 513–530, 2016.

[31] Jiaao Chen, Yuwei Wu, and Diyi Yang. Semi-supervised models via data augmentationfor classifying interactive affective responses. *arXiv preprint arXiv:2004.10972*, 2020.

[32] Catalin Cimpanu. Fcc tells us telcos to implement caller id authentication by june 30, 2021. https://www.zdnet.com/article/fcc-tells-us-telcos-to-implement-caller-id-authentication-by-june-30-2021/, Mar 2020.

[33] Leigh Clark, Nadia Pantidi, Orla Cooney, Philip Doyle, Diego Garaialde, Justin Edwards, Brendan Spillane, Emer Gilmartin, Christine Murad, Cosmin Munteanu, et al. What makes a good conversation? challenges in designing truly conversational agents. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[34] Meir Cohen, Eliyahu Finkelman, Ethan Garr, and Bryan Moyles. Call distribution techniques. U.S. Patent 9,584,658, issued February 28, 2017.

[35] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[36] Andrei Costin, Jelena Isacenkova, Marco Balduzzi, Aurélien Francillon, and Davide Balzarotti. The role of phone numbers in understanding cyber-crime schemes. In *2013 Eleventh Annual Conference on Privacy, Security and Trust*, pages 213–220. IEEE, 2013.

[37] Valerie Fanelle, Sepideh Karimi, Aditi Shah, Bharath Subramanian, and Sauvik Das. Blind and human: Exploring more usable audio captcha designs. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, pages 111–125, 2020.

[38] Mahak Gambhir and Vishal Gupta. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66, 2017.

[39] Karthik Gopalakrishnan, Behnam Hedayatnia, Qinglang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, Dilek Hakkani-Tür, and Amazon Alexa AI. Topical-chat: Towards knowledge-grounded open-domain conversations. In *INTERSPEECH*, pages 1891–1895, 2019.

[40] Ting-Hao Kenneth Huang, Walter S Lasecki, Amos Azaria, and Jeffrey P Bigham. " is there anything else i can help you with?" challenges in deploying an on-demand crowd-powered conversational agent. In *Fourth AAAI Conference on Human Computation and Crowdsourcing*, 2016.

[41] Vlado Keselj. Speech and language processing daniel jurafsky and james h. martin (stanford university and university of colorado at boulder) pearson prentice hall, 2009, xxxi+ 988 pp; hardbound, isbn 978-0-13-187321-6, 2009.

[42] Huichen Li, Xiaojun Xu, Chang Liu, Teng Ren, Kun Wu, Xuezhi Cao, Weinan Zhang, Yong Yu, and Dawn Song. A machine learning approach to prevent malicious calls over telephony networks. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 53–69. IEEE, 2018.

[43] Shujun Li, S Amier Haider Shah, M Asad Usman Khan, Syed Ali Khayam, Ahmad-Reza Sadeghi, and Roland Schmitz. Breaking e-banking captchas. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 171–180, 2010.

[44] Weixin Liang, Youzhi Tian, Chengcai Chen, and Zhou Yu. Moss: End-to-end dialog system framework with modular supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8327–8335, 2020.

[45] Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study

of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*, 2016.

[46] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*, 2019.

[47] Hendrik Meutzner, Santosh Gupta, Viet-Hung Nguyen, Thorsten Holz, and Dorothea Kolossa. Toward improved audio captchas based on auditory perception and language understanding. *ACM Transactions on Privacy and Security (TOPS)*, 19(4):1–31, 2016.

[48] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[49] Najmeh Miramirkhani, Oleksii Starov, and Nick Nikiforakis. Dial one for scam: A large-scale analysis of technical support scams. *arXiv preprint arXiv:1607.06891*, 2016.

[50] Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*, 2016.

[51] Hossen Mustafa, Wenyuan Xu, Ahmad Reza Sadeghi, and Steffen Schulz. You can call but you can't hide: detecting caller id spoofing attacks. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 168–179. IEEE, 2014.

[52] Sharbani Pandit, Roberto Perdisci, Mustaque Ahamad, and Payas Gupta. Towards measuring the effectiveness of telephony blacklists. In *NDSS*, 2018.

[53] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[54] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[55] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

[56] Sathvik Prasad, Elijah Bouma-Sims, Athishay Kiran Mylappan, and Bradley Reaves. Who's calling? characterizing robocalls through audio and metadata analysis. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 397–414, 2020.

[57] Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *International conference on machine learning*, pages 5231–5240. PMLR, 2019.

[58] Anirudh Ramachandran and Nick Feamster. Understanding the network-level behavior of spammers. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 291–302, 2006.

[59] Bradley Reaves, Logan Blue, Hadi Abdullah, Luis Vargas, Patrick Traynor, and Thomas Shrimpton. Authenticall: Efficient identity and content authentication for phone calls. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 575–592, 2017.

[60] Bradley Reaves, Logan Blue, and Patrick Traynor. Authloop: End-to-end cryptographic authentication for telephony over voice channels. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 963–978, 2016.

[61] Merve Sahin, Aurélien Francillon, Payas Gupta, and Mustaque Ahamad. Sok: Fraud in telephony networks. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 235–250. IEEE, 2017.

[62] Merve Sahin, Marc Relieu, and Aurélien Francillon. Using chatbots against voice spam: Analyzing lenny's effectiveness. In *Thirteenth Symposium on Usable Privacy and Security ({SOUPS} 2017)*, pages 319–337, 2017.

[63] Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. A survey of available corpora for building data-driven dialogue systems. *arXiv preprint arXiv:1512.05742*, 2015.

[64] Imani N Sherman, Jasmine Bowers, Keith McNamara Jr, Juan E Gilbert, Jaime Ruiz, and Patrick Traynor. Are you going to answer that? measuring user responses to anti-robocall application indicators. In *NDSS*, 2020.

[65] Camelia Simoiu, Ali Zand, Kurt Thomas, and Elie Bursztein. Who is targeted by email-based phishing and malware? measuring factors that differentiate risk. In *Proceedings of the ACM Internet Measurement Conference*, pages 567–576, 2020.

[66] Saumya Solanki, Gautam Krishnan, Varshini Sampath, and Jason Polakis. In (cyber) space bots can hear you speak: Breaking audio captchas using ots speech recognition. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 69–80, 2017.

[67] Bharat Srinivasan, Payas Gupta, Manos Antonakakis, and Mustaque Ahamad. Understanding cross-channel abuse with sms-spam support infrastructure attribution. In *European Symposium on Research in Computer Security*, pages 3–26. Springer, 2016.

[68] Bharat Srinivasan, Athanasios Kountouras, Najmeh Miramirkhani, Monjur Alam, Nick Nikiforakis, Manos Antonakakis, and Mustaque Ahamad. Exposing search and advertisement abuse tactics and infrastructure of technical support scammers. In *Proceedings of the 2018 World Wide Web Conference*, pages 319–328, 2018.

[69] Nadeem Ahmed Syed, Nick Feamster, A Gray, and Sven Krasser. Snare: Spatio-temporal network-level automatic reputation engine. *Georgia Institute of Technology-CSE Technical Reports-GT-CSE-08-02, Tech. Rep.*, 2008.

[70] Jennifer Tam, Jiri Simsa, Sean Hyde, and Luis V Ahn. Breaking audio captchas. In *Advances in Neural Information Processing Systems*, pages 1625–1632, 2008.

[71] Huahong Tu, Adam Doupé, Ziming Zhao, and Gail-Joon Ahn. Sok: Everyone hates robocalls: A survey of techniques against telephone spam. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 320–338. IEEE, 2016.

[72] Huahong Tu, Adam Doupé, Ziming Zhao, and Gail-Joon Ahn. Toward authenticated caller id transmission: The need for a standardized authentication scheme in q. 731.3 calling line identification presentation. In *2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT)*, pages 1–8. IEEE, 2016.

[73] Huahong Tu, Adam Doupé, Ziming Zhao, and Gail-Joon Ahn. Users really do answer telephone scams. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 1327–1340, 2019.

[74] Abraham Wald. Sequential tests of statistical hypotheses. *The annals of mathematical statistics*, 16(2):117–186, 1945.

[75] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[76] Yi Wu, Xiangyu Xu, Payton R Walker, Jian Liu, Nitesh Saxena, Yingying Chen, and Jiadi Yu. Hvac: Evading classifier-based defenses in hidden voice attacks. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 82–94, 2021.

[77] Zhao Yan, Nan Duan, Junwei Bao, Peng Chen, Ming Zhou, Zhoujun Li, and Jianshe Zhou. Docchat: An information retrieval approach for chatbot engines using unstructured documents. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 516–525, 2016.

[78] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93, 2020.