SkillDetective: Automated Policy-Violation Detection of Voice Assistant Applications in the Wild

Jeffrey Young*, Song Liao*, Long Cheng
Clemson University

Hongxin Hu
University at Buffalo

Huixing Deng
Clemson University

Abstract

Today's voice personal assistant (VPA) services have been largely expanded by allowing third-party developers to build voice-apps and publish them to marketplaces (*e.g.*, the Amazon Alexa and Google Assistant platforms). In an effort to thwart unscrupulous developers, VPA platform providers have specified a set of policy requirements to be adhered to by third-party developers, *e.g.*, personal data collection is not allowed for kid-directed voice-apps. In this work, we aim to identify policy-violating voice-apps in current VPA platforms through a comprehensive dynamic analysis of voice-apps. To this end, we design and develop SKILLDETECTIVE, an interactive testing tool capable of exploring voice-apps' behaviors and identifying possible policy violations in an automated manner. Distinctive from prior works, SKILLDETECTIVE evaluates voice-apps' conformity to 52 different policy requirements in a broader context from multiple sources including textual, image and audio files. With SKILLDETECTIVE, we tested 54,055 Amazon Alexa skills and 5,583 Google Assistant actions, and collected 518,385 textual outputs, approximately 2,070 unique audio files and 31,100 unique images from voice-app interactions. We identified 6,079 skills and 175 actions potentially violating at least one policy requirement.

1 Introduction

Smart speakers have become an intrinsic part of daily life for millions of people, largely due to the functionality made convenient through an on-board voice personal assistant (VPA). Today's VPA ecosystem (*e.g.*, Amazon Alexa and Google Assistant platforms) advertises hundreds of thousands of voice-apps[1] with functions such as locking a door, arming an alarm, or checking a credit card balance. For example, Amazon's Alexa platform currently boasts over 100,000 skills available in its skill store [5]. To accomplish this scale of product availability, both leading VPA platforms, *i.e.*, Amazon Alexa and Google Assistant, allow third-party developers to publish their own skills directly to the VPA's app store.

In an effort to thwart unscrupulous content, VPA platforms have defined a set of policy requirements [1–3, 9] to be adhered to by third-party developers. Skill publishing is overseen by a vetting process which rejects a skill if it violates any of these policies. In a previous work [23], the skill vetting processes of Amazon Alexa and Google Assistant platforms were put to the test. The results show the ease of policy-violating skills being certified by both platforms. Because of this lack-adaisical approach to skill vetting, it is almost inevitable that policy-violating content will be published. Therefore, there exists a need for a large-scale skill testing/analysis to determine if skills are within the policy guidelines.

Despite mounting evidence of multiple security/privacy flaws in VPA systems, little effort has been made to *comprehensively evaluate the policy compliance of skills*. In contrast to traditional smartphone platforms (*e.g.*, Android or iOS) where apps run on host smartphones, a skill's back-end code runs on the developer's server (*e.g.*, hosted by AWS Lambda under the developer's account or other third-party servers). Since the skill's code is hosted externally and is not available, using static code analysis to explore a skill's functionality is not an option for current VPA systems. As a result, dynamic analysis (by invoking and interacting with a skill) is currently the only option to understand a skill's actual behavior.

In this work, we seek to understand the range and scope of how existing skills conform to various policy requirements in the skill stores. To this end, we design and develop SKILLDETECTIVE, which is a scalable and robust testing tool to identify potential policy-violating skills. SKILLDETECTIVE significantly extends skill testing capabilities in a broader context.

In summary, we make the following contributions:

- **New tool development**. We designed and developed a dynamic testing tool, named SKILLDETECTIVE, with the capabilities to automatically test skill behaviors and report on any potential policy violations against various

*The first two authors contributed equally to this work.

[1] On the Amazon Alexa platform voice-apps are referred to as skills, while on the Google Assistant platform they are referred to as actions. Throughout this paper we refer to voice-apps as *skills* unless the need arises to differentiate the two platforms.

policy requirements. We shared the SKILLDETECTIVE tool and all datasets to facilitate future research.

- **A large-scale analysis of skills**. We conducted a comprehensive analysis of skills to detect if they are potentially in compliance with current policies of VPA platforms. After over a year of development and testing, we have tested 54,055 Amazon Alexa skills and 5,583 Google Assistant actions, and gathered 518,385 textual outputs, approximately 2,070 unique audio files and 31,100 unique images in total from skill interactions. Such a wide-range and large-scale policy violation detection of skills has not previously been reported.
- **Findings**[2] . We identified 6,079 skills and 175 actions potentially violating at least one policy requirement. 590 skills and 24 actions potentially violate more than one policy. In the Kids category, we identified 244 policy-violating skills. 80% of skills and 68% of actions in the Health category potentially violate at least one policy. 623 skills and 25 actions potentially violate policies related to personal data collection. We have reported the identified policy-violating skills to both vendors, and received their acknowledgments. Google had immediately removed 43 policy-breaking actions from their store, and awarded us a bug bounty for reporting these issues. The Amazon Alexa team appreciated our work which brings potential issues to their attention.

2 Background

2.1 Skill Structure and Interaction

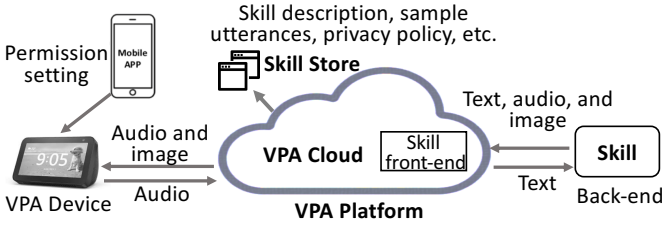

Figure 1: VPA platform and skill interaction.

Figure 1 shows a high-level view of the VPA platform and skill interaction flow. Skills, like smartphone apps, are mostly created by third-party developers and are available through a website known as the skill store. Each skill has a unique web-page that displays the skill's listing, consisting of the developer information, description, sample utterances, privacy policy, user ratings, user reviews, etc. Sample utterances are sets of likely spoken phrases by users. A skill's privacy policy on the skill store differs from enforcement policies by VPA platforms in that it should outline any data collected by the skill and the subsequent use of that data. Skills may request access to personal information from some users in order to provide customized information in skill responses. In this case, the user is supposed to use his/her VPA companion app (Android/iOS) to grant permission so that the skill can obtain the requested personal information.

A skill has a front-end interface and back-end code which manages how it responds to user requests. VPA platforms provide hosting for the front-end interface of a skill, but its back-end code is typically hosted on the developer's server (*e.g.*, hosted by AWS Lambda under the developer's account or other third-party servers). Skills are also unique in the way they are accessed. All skills are made to be verbally interactive and primarily return an audible response. The back-end code may also provide other media types such as pre-recorded audio streams and images (for VPA devices with displays). Modern VPA devices (*e.g.*, Amazon Echo Show in Figure 1) may have visual displays. These displays are used by skills to display such media as text, images and even movies.

For skill developers, VPA platforms offer a skill simulator for testing purposes. The simulators for both the Amazon Alexa and Google Assistant platforms are similar. They both consist of a virtual VPA device that can interact with other skills on the skill store. For ease of testing, the skill simulators offer a text-based interface that will accept a textual input, and provide a textual output as well as deliver any external content (*e.g.*, image files).

2.2 Policies Defined by VPA Platforms

In an effort to maintain content safety and privacy on their platforms, VPA platform providers claim to enforce a set of policies onto skill developers. These policies, which are checked during the skill certification process, are designed to limit the amount of potentially exploitable content allowed onto the skill store. We outline the policies and their requirements. Amazon Alexa has specified 7 privacy requirements [2] and content policy guidelines [1] which are categorized into 14 main sections. Google Assistant has 19 sections for content restrictions [9]. These policies/restrictions focus on the content being delivered to the user in a skill and the collection of data from users. For example, skills should not have promotions, advertisements, or promote alcohol or tobacco usage. All certified skills in the skill stores are expected to align with these policy requirements and guidelines. If a skill contains, facilitates, or promotes content that is prohibited by these policy guidelines, it will be rejected or suspended. The policies in place for developers are in no way comprehensive, but do attempt to enforce some protections for vulnerable VPA users.

3 System Overview

The diverse nature of content policies poses a challenge for policy violation detection. Due to the intrinsic challenges of skill testing, there does not currently exist a content detector capable of detecting various policy violations in skills. Therefore, it was necessary to design and implement a testing

[2]The details of our testing results, datasets, source code, and demos are available at https://github.com/skilldetective/skilldetective.

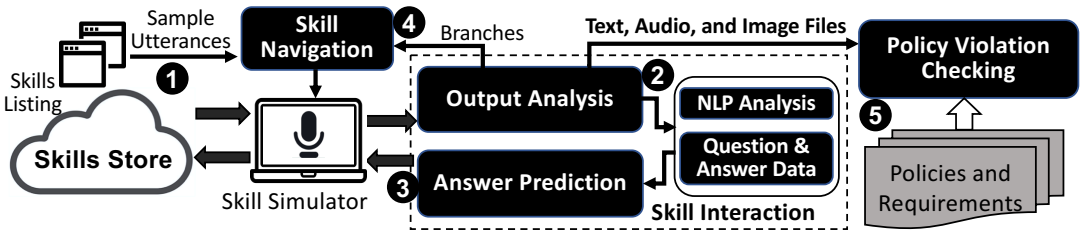

Figure 2: SKILLDETECTIVE overview.

system that not only communicates with skills, but also can detect and identify policy violating skill behaviors.

Figure 2 shows the design overview of SKILLDETECTIVE, an interactive testing tool capable of exploring a skill's comportment and identifying policy violations through the analysis of its outputs. We first collect sample utterances from a skill's description located in the skill store (❶), which are utilized to initiate the first interaction with a testing skill. In fact, the skill store requires developers to provide a selection of invocations for the purpose of letting end-users understand how to use the skill. These sample utterances are easily accessible on the skill's web-page located in the skill store. To interact with the skill, SKILLDETECTIVE employs a web driver tool (Selenium WebDriver [12]) to directly interface via text with the skill simulator provided by VPA platforms. Using this method, SKILLDETECTIVE is able to collect a skill's output as well as input a response to the skill. Once an output is received from the skill, the question analysis module (❷) uses a data-driven approach for classifying outputs by type. These classifications include 5 question types and 1 "not a question" type. When the question type is determined, the analysis tool predicts the appropriate response needed to provoke further interaction data (❸). The output is either answered by using a set of grammar rules or by utilizing a neural network model trained on a corpus of question-answer pairs. The neural model's training data was collected through the mining of GitHub repositories featuring skill source code. Subsequently, a dataset of question/answer pairs was extracted from skill interactions embedded in the source code. For questions that have multiple answers, the skill navigation module (❹) maintains a skill-tree where nodes represent the responses of the skill outputs (questions) and the branches represent the inputs (answers). The skill outputs (consisting of text, sound and image files) are stored and analyzed for possible policy violations (❺). We are particularly interested in the policy enforcement for child-directed and health-related skills, which require more stringent policies since 1) kids are more vulnerable to potential threats compared to adults, and 2) more sensitive details of users' lives and medical conditions are involved in health-related skills.

Distinction from prior work. SKILLDETECTIVE distinguishes itself from existing work [27, 37] in four ways. 1) We adopt a data-driven approach for classifying question types and generating their answers. 2) Our detection is not only for text but also includes skill media data, such as audio and image files. Some skills play audio files that are hosted externally and are not in the voice of the device. These external files do not get translated by the skill simulators and must be processed separately in order to extract any policy violations as well as to navigate some skills. Also, many newer skills contain image files, which must be processed to extract any content violations. 3) We consider a wide range of policies when analyzing skill outputs. The policy violation detector in SKILLDETECTIVE currently checks for 52 different policies defined by VPA platform providers. 4) We improve upon skill navigation by dynamically exploring skill paths and adding them to a stack in the navigation model, which improves the ability of SKILLDETECTIVE to handle complex skills. In addition, we conduct a large-scale dynamic and static analysis of skills to detect policy non-compliant skills in both the Amazon Alexa and Google Assistant platforms.

4 SKILLDETECTIVE Design

4.1 Data-Driven Question Analysis

Typically, skill outputs can be answers to previous questions/requests or may ask further questions. To continue the conversation, our goal is to first identify whether a skill's response is a question. If yes, we need to understand the question and provide a valid response. We define a *valid response* as any response capable of being understood by the VPA and that allows the conversation to continue.

Question Types. We mainly consider five types of questions. Table 1 lists the question types and corresponding example questions. 1) Binary questions (Yes/No) have only two possible answers: yes or no. 2) Instruction questions give the user assistance in determining how to answer them. Structurally, this question type usually tells the user how to respond by using directive keywords (*e.g.*, "tell" or "say"). Authors in [27] found that over 96% of instruction questions asked through skills use the words "ask" and "say". 3) Selection questions contain multiple answers and are also referred to as "choice" questions if the answers are connected by the keyword "or". The question gives the user a list of possible answers either explicitly or implicitly. 4) Open-ended questions (also known as free-form or "wh" questions) require the user to answer in a free manner. An example of this type is "What is your mother's name?". As long as we provide any valid name, we can continue the conversation with the skill. 5) Mixed questions contain elements from more than

one of the above question types. For example, “There are A, B, and C to choose from. Which one do you want?” The question “Which one do you want?” is a Free-form question, but the answers “A, B, and C” provided in the previous statement are of the selection type. Structurally, this question type has mixed attributes that match more than one of the other question types. If a skill’s output does not represent any type of inquiry and demands no response, we consider it as a statement and classify it as NULL.

Question Type	Example	Valid Answers
Binary	Are you in the car?	Yes, No
Instruction	Say your name.	George
Explicit Selection	Do you want to eat, run, or watch TV?	Eat, Run, Watch TV
Implicit Selection	Choose a number between 1 and 3?	1, 2, 3
Open-ended	What is your mother’s name?	Amy
Mixed	There are A, B, and C to choose from. Which one do you want?	A, B, C
NULL	Zebras are black and white.	N/A

Table 1: Different types of questions from skills.

Question Classification. To identify question types, we adopt a data-driven methodology and build a classifier trained on a corpus of labeled questions and statements. A sampling of 1,000 questions (200 per type) were randomly chosen from template skill source code mined from GitHub and hand labeled by type to be used by the question type classifier. Another random sampling of 200 statements were added to the dataset and hand labeled NULL. Labeling the NULL values allows the classifier to correctly classify statements that are not questions and demand no further communication during the skill testing. Utilizing the data-driven classifier gives us distinct advantages over the grammar-rule based approach in SkillExplorer [27]. Evaluation results in Section 6.2 show that our approach improves the questions classification of Mixed questions by 11% and statements by 26.3%.

Classification is performed by utilizing the structural differences of each question type via the creation of a parts-of-speech (PoS) signature that is used to differentiate between types. PoS tagging assigns labels to tokens (words), signifying whether they are nouns, verbs, adjectives, etc. Every token in a sentence has a tag applied. We use the Stanford Core-NLP package [14] to extract the PoS tags from the questions. For example, if analyzing the sentence "Alex was raised in the U.S.", we would assign an NNP (proper noun, singular) tag for Alex indicating that the name “Alex” is a singular proper noun. Tags would also be applied to the remaining tokens in the sentence creating a pattern.

Algorithm 1: Question Classification

Input: Question corpus (C); Target question (α)
Output: Question type of α;

```
μ = 0;   q = Null;
foreach n ∈ C do
    if Similarity(PoS(α), PoS(n)) > μ then
        μ = Similarity(PoS(α), PoS(n));
        q = QuestionType(n)
return q;
```

Algorithm 1 shows the question classification algorithm. The classifier is provided a corpus of labeled data (C) and a target question (α). The labeled corpus contains two attributes (Q and T), which represent the questions and their types. To start, $\mu = 0$ and q = NULL. For each labeled question n in C, the similarity of n to α is calculated. If the similarity is greater than μ, then μ is set equal to the similarity value and q is set equal to the question type of n. Finally, q is returned as the type of the question with the highest similarity value. To calculate the similarity of n to α, we permute through all conjoined PoS patterns created from n and count the amount of times they show up in the target question α. Details of the similarity calculation can be found in Appendix B.

4.2 Skill Interaction

Answer Prediction. Given a skill’s response and after identifying its question type, we generate a valid answer to continue the conversation. We adopt similar approaches used in SkillExplorer [27] to generate answers for the Binary, Instruction, Selection and Open-Ended types of questions. The Binary type of question is the easiest to answer. Upon classification, the answer is either "yes" or "no". To extract the answer from an Instruction type, we identify specific patterns associated with the commands "ask" and "say". To answer the Selection type, the answers appear connected by conjunctions or clearly marked by identifiers such as numbers or letters. To answer an Open-ended question, first a dataset of virtual user profiles was created. These profiles contain such information as name, age, address, etc. Keywords from the questions are searched within the dataset and the corresponding answer is returned.

The SkillExplorer methodology [27] is not suited for answering Mixed type questions as accurately as the other types. This is likely due to the fact that the grammar-rules developed for the Mixed type question were created using a small (for this type of methodology) labeled set of data (only 2,000 questions sampled from skill responses in SkillExplorer). Another possible factor is the fact that the grammar-rules are fixed and not dynamic. We found through testing that SkillExplorer’s accuracy in answering the Mixed type questions is around 87% (Details in Section 6.2). To improve upon this, we develop a different method for answering the Mixed types. We utilize an FNN (Feed-Forward Neural Network) model [26], which is trained on a corpus of question/answer pairs and predicts the most probable answer based on question similarity. We chose an FNN model because they can achieve advanced results on a range of language processing tasks while being considerably less resource expensive than deep recurrent models [18].

To utilize the FNN for answer prediction, we first vectorize all questions in our corpus. To this end, we utilize a variation of the Count Vectorizer technique [26]. We identify representative keywords from the complete corpus of question/answer pairs, by extracting the nouns of each question. We choose the nouns of each question because they represent the subject of the question. These nouns form the representative keywords, which together form a bag of words (BoW). We next create

our feature vector by calculating the average Levenshtein Distance (LD) [44] between each individual keyword in the BoW to every word in our target question (Details are provided in Appendix B). By using this method, the corpus is vectorized to be used as a training set for the neural model. The system classifies each input question by using the Sigmoid Function for activation and subsequently predicts the most probable answer. Different parameters were tested such as the learning rate and number of hidden layers against a test set of data using cross validation. In SKILLDETECTIVE, the optimal learning rate was set to 0.85 with 35 nodes in the hidden layers and a tolerance of 0.05.

Skill Navigation. Structurally, skill interaction can be represented as a dynamically growing tree. The nodes of the tree represent the responses of the skill (*e.g.*, questions) while the branches represent the inputs (*e.g.*, answers). We use this skill-tree to keep track of what nodes SKILLDETECTIVE has interacted with. Because skill testing is an iterative process that requires a fresh start for each branch, we must keep track of all branches that have and have not been explored in order to maximize skill coverage while minimizing testing latency.

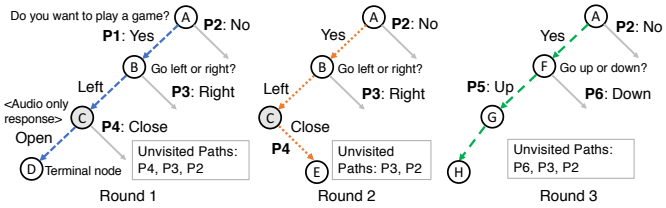

Figure 3: Example of the skill-tree for skill navigation.

Figure 3 illustrates an example of how the skill-tree is used to navigate during skill testing. Skill navigation begins with the instantiation of a single node (*i.e.*, node A), which represents the initial skill output following an activation term (*e.g.*, "Alexa, open [skill]."). Processing this output, SKILLDETECTIVE determines what content is contained such as questions, question types, and statements. Using this information, the skill tree can begin to grow and the first of the skill's paths are created. A skill path represents a single pathway through the skill tree to a terminal node (*i.e.*, the NULL type).

Take the example in Figure 3 (Round 1), suppose the skill output is "Do you want to play a game?" The possible answers expected to be returned are "yes" or "no". These two answers are represented by branches coming from node A. SKILLDETECTIVE would generate two skill paths, and each path begins by recording the first node A and the subsequent answers "yes" and "no". The initial paths are represented as P1: A->"yes" and P2: A->"no". To continue, SKILLDETECTIVE always follows the leftmost path (*e.g.*, P1) in the skill tree, and pushes the other path candidates (*e.g.*, P2) onto a stack *Unvisited_Paths*. It goes to node B with a skill output "There is a road. Do you go left or right?". The answer prediction module generates two possible responses "left" and "right". Thus, a forking path P3: A->"yes"->B->"right" is pushed onto *Unvisited_Paths*. For this turn, we respond "left" which takes us to node C. Suppose node C contains an external audio file, where the testing console outputs the terms <Audio only response> or <Short audio>. When one of these terms are received by the navigation module, the audio file is fetched, transcribed using the speech-to-text software, and finally the transcription is used as input to the answer prediction module. In Figure 3, node C contains a question "There are two doors. Do you want to open the closed door or close the opened door?". As a result, another forking takes place and path P4: A->"yes"->B->"left"->C->"close" is pushed onto *Unvisited_Paths*. Following P1, we choose "open", and the skill responds with "You are attacked and died. The end!" which is a statement and makes node D terminal. If the skill output is a terminal node, SKILLDETECTIVE returns "Exit" and "Stop" to bring the interaction to an end.

SKILLDETECTIVE must restart once per skill path. This process is repeated until there are no more skill paths left to explore. Figure 3 (Round 2) shows the process to explore the path P4 in *Unvisited_Paths*. It follows the pre-defined path A->"yes"->B->"left"->C->"close" and explores any downstream nodes after returning the answer "close" to the skill. Suppose the next node SKILLDETECTIVE encounters is node E which is terminal. Subsequently, P4 is removed from *Unvisited_Paths*.

Next, we discuss dynamic content, where skill outputs are not fixed in different rounds of testing. In Figure 3 (Round 3), SKILLDETECTIVE restarts the conversation to explore path P3: A->"yes"->B->"right". After node A, SKILLDETECTIVE responds "yes" to the skill. However, this time the skill output contains a new question "Do you go up or down?". The corresponding answers are "up" and "down". In this case, two new paths P5 and P6 are created. For practical consideration when conducting a large-scale testing, to avoid SKILLDETECTIVE exploring a skill tree infinitely (due to the path explosion), we introduce a threshold for limiting the depth of individual paths in our dynamic testing of skills. Therefore, SKILLDETECTIVE interacts with a skill in a best-effort manner.

4.3 Policy Violation Detection

We mainly focus on four types of policies related to privacy and content safety: 1) policies for specific categories (*i.e.*, the Kids and Health categories); 2) policies on data collection and privacy notification for general categories; 3) policies on skill descriptions; and 4) policies on content safety (*e.g.*, toxic or inappropriate content). Table 12 and Table 13 in Appendix D list the detailed policy violations that we target to detect in SKILLDETECTIVE. Note that collecting personal data from users is forbidden for skills in the Kids and Health categories. For the other categories, skills may collect user data but they need to provide a privacy policy outlining data usage. There are three categories of policies that we do not consider because to do so would be difficult without human intervention: 1) policies about trademarks and brands; 2) policies on whether a skill uses in-skill purchasing or web search correctly; and 3) misleading information in skills.

Through the mining of skill stores, we were able to attain a skill's name, sample utterances, category, description, privacy policy, permissions, etc. During dynamic testing, SKILLDETECTIVE collects the skill's outputs by exploring possible interaction paths. Although the skill simulator provides textual outputs for most skills, some skills provide additional pre-recorded audio streams and images which may contain policy violations. This means that SKILLDETECTIVE also must check audio and image outputs from skills.

SKILLDETECTIVE first detects potential violations of policies specific to skills in the Kids and Health categories. For all categories, it detects personal data collection in a skill, and then captures any inconsistencies among privacy notice/policy, description and run-time skill behavior. Since Google's VPA platform has defined several data types that are protected by their permission models (*i.e.*, collecting these specific types of data should be through the permission APIs), SKILLDETECTIVE also identifies Google actions that collect these data without using the required permission APIs. There are more than 40 specific policies defined by VPA platforms about skill outputs for content safety. SKILLDETECTIVE detects policy violations of content safety in skill outputs.

4.3.1 Detecting violations of policies specific for skills in the Kids and Health categories.

Kids category. The Amazon Alexa platform has defined three specific policies for skills in the Kids category. Skills are not allowed to 1) collect any personal information from end users; 2) direct end users to engage with content outside of Alexa; and 3) include content not suitable for all ages. The Google Assistant platform has specified the first and third policies, but does not explicitly prohibit skills from directing users to external websites.

To detect personal data collection in skills, we selected 21 types of common PII (personal identifiable information) from a NIST (National Institute of Standards and Technology) report [33], as shown in Table 2. Given a skill's output, we use the `Spacy` library [6] for analysis and to check whether any keyword is used as a noun. This is because some words such as "address" or "email" can be a verb in a sentence instead of a noun. Since skill developers might provide their own information such as email or phone number in the conversation, we only detect the personal data with the keyword "your". Specifically, we limit the name as "name", "first name", "last name" and "full name" since there may be other types of names in a skill's output, such as "your shopping name" or "your group name". In addition, we manually collect a list of common sentences of personal data collection such as "what can I call you", "how old are you", and "where do you live". It improves the detection accuracy by checking for the existence of any sentence in the list.

For the second kids-specific policy (only for Amazon Alexa skills), we detect whether there exists a website URL (other than Amazon and Alexa's domains) in skill outputs to direct users to external websites. For the third kids-specific policy, since some skills explicitly mention they include mature content not suitable for all ages in the outputs, we detect keywords such as "mature content" in skill outputs. We also conduct toxic content detection, since toxic content such as violence, profanity and sex are forbidden for all categories.

Personally Identifiable Information (PII) [33]	Address, Name, Email, Birthday, Age, Gender, Account, Location, Contact, Phonebook, Profession, Income, Zipcode, Postal code, Phone number, Passport number, Driver license number, Bank account number, Debit card number, Credit card number, SSN
Common Health Information	Height, Weight, Blood group, Blood pressure, Blood glucose, Blood Oxygen, Heart rate, Body temperature, Sleep data, Fat percentage, Mass index, Waist circumference, Menstruation, Period
Protected Health Information (PHI) [16]	Name, Phone number, Address, SSN, Email address, Account, Internet protocol address, Age, Gender, Birthday, Medical record number, Health plan beneficiary number, Driver license number
Verb Set Related to Data Collection	Access, Ask, Assign, Collect, Create, Enter, Gather, Import, Obtain, Observe, Organize, Provide, Receive, Request, Share, Use, Include, Integrate, Monitor, Process, See, Utilize, Retain, Cache, Delete, Erase, Keep, Remove, Store, Transfer, Communicate, Disclose, Reveal, Sell, Send, Update, View, Need, Require, Save

Table 2: Keywords related to personal data collection.

Health category. Both Amazon and Google have defined specific policies for skills in the Health category. They restrict data collection about health information from users, and also require skills that provide health-related information to include a disclaimer in their descriptions. For data collection in the Health category, Amazon doesn't allow skills to collect information relating to any person's physical or mental health or condition while Google clearly defines that actions cannot collect "information that could be considered protected health information (PHI) [16] under the Health Insurance Portability and Accountability Act (HIPAA)". But, Google does allow actions with fitness functions to collect some common health information such as calories burned, steps taken, weight data, BMI, etc. Therefore, SKILLDETECTIVE detects the data collection of PHI for both platforms and conservatively detects the data collection of common health information only for the Amazon platform. We use the same method of detecting kids' data collection to detect health data collection in skills and actions, where the keywords are listed in Table 2.

4.3.2 Detecting policy violations in the privacy notice

For all categories, the Amazon Alexa platform requires skills with data collection to provide a privacy policy/notice while the Google Assistant platform requires every action to have one. In a privacy policy document, developers should clearly disclose the data collection practices of a skill. SKILLDETECTIVE mainly detects four types of potential violations related to privacy policies: 1) a skill doesn't provide a privacy policy (*i.e.*, missing a privacy policy) although it is required; 2) a skill has a privacy policy, but does not disclose all data collection practices in the privacy policy (*i.e.*, incomplete privacy policy); 3) a skill explicitly mentions it does not collect data in its privacy policy, but it actually does collect data (*i.e.*, deceptive privacy policy); and 4) for Google actions, an action does not request permissions to access certain types of user data, but collects these data through the conversational interface.

We detect whether there exists data collection behavior in a skill's outputs by using the same NLP-based method described in Section 4.3.1. For detecting missing privacy policies, we check whether the privacy policy of a skill is provided and if the skill contains data collection. To detect incomplete privacy policies, we adopt a simple yet effective keyword-based approach. We maintain a verb set [19, 32, 42] related to data collection, which includes 40 commonly used verbs in privacy policies as listed in Table 2. For each sentence in a privacy policy, we check whether any PII data type listed in Table 2 is collected. To improve the detection accuracy, we use a data type ontology defined in [30]. Specifically, "address", "location", "geolocation" and "position" will be treated as data types in the same level. While "zip code" and "postal code" are data types in the lower level, which can be covered by the higher level data types. For example, a skill asking for the location permission but only claiming the zip code collection in a privacy policy would be considered as having an incomplete privacy policy. On the contrary, if it asks for the zip code but claims collecting location data, it has a complete privacy policy. For detecting deceptive privacy policies, we use the `PolicyLint` [17] (a privacy policy analysis tool) to obtain negative statements about whether or not certain types of user data are collected. Some privacy policies would claim they do not collect user data, but we observed data collection from their skill outputs. For example, the skill "Reggie Birthday Reminder" claims that "We never collect or share personal data with our skills" but asks for the user's birthday through the conversation channel.

In addition, both the Amazon Alexa and Google Assistant platforms provide permission requesting APIs for skills collecting specific types of data from users. Specifically, Google requires that an action must "Request all sensitive user data (location and name) via the Permissions API". The Amazon Alexa platform allows developers to request permissions for collecting device address, customer name, customer email address, customer phone number and location for skills in general categories. If a skill collects these data through permission APIs, Amazon would display the permission information in the skill's introduction page, and we would check whether its privacy policy has disclosed such data practices or not. Google doesn't provide the permission information on the action's introduction webpage. For our detection, if an action collects user name or location through the conversational interface, it is flagged as a possible policy violation since the skill should use permission APIs for such data collection. Note that we do not detect this policy violation for the Amazon Alexa platform since it doesn't explicitly require developers to use Permission API when collecting the permission protected data types.

4.3.3 Detecting policy violations in skill descriptions

A unique policy for the skill description is that both Amazon and Google platforms require health-related skills to have a disclaimer in their descriptions. Amazon policy requires "a skill that provides health-related information, news, facts or tips should include a disclaimer in the skill description stating that the skill is not a substitute for professional medical advice". Google also has a similar policy. Amazon goes so far as to provide an example description disclaimer, "This tool does not provide medical advice, and is for informational and educational purposes only, and is no substitute for professional medical advice, treatment or diagnosis."

For detecting if a disclaimer is missing in a skill description, we first check whether the words "medical advice", "educational purpose" or "information purpose"(keywords taken from the sample description disclaimer) show up in the description. This is because a lot of skills copy the sample disclaimer. For other cases, we compare the similarity of each sentence in the skill description to each sentence in the example disclaimer using the `Spacy` library [6]. If the similarity score is greater than a threshold, we consider the sentence as a disclaimer. After checking the descriptions of randomly selected 500 skills (as testing data), we found that choosing the similarity threshold to be 0.93 yields the best result. Our method achieved an accuracy of 98% when applying this threshold to all the skills (Details in Section 6.1.3).

We also check whether the description violates other policies, such as data collection or requesting a positive rating. Some skills would mention their data collection in the description. For example, the skill "Omron Health" says they would "Monitor and track your blood pressure". Amazon requires that skills cannot "Explicitly request that users leave a positive rating of the skill". We search for keywords such as "5 star review" or "five star rating" in a skill description and check whether the keywords are associated with verbs "give" or "leave". We found a large number of skills requesting a positive rating in their skill descriptions (Details can be found in Section 6.1.3).

4.3.4 Detecting policy violations for content safety

Most policies defined by VPA platforms are about the content safety of skill outputs. For these policies, we first detect toxic or inappropriate content in the skill outputs. Several of Amazon's policies mention that skills should not provide content about sex, violence, anything illegal, hate or profanity. Google also prohibits such content. To detect toxic content, we use the `Perspective` tool (an inappropriate content detection tool) [8] to analyze the skill outputs, and it calculates the severe toxicity score given an input sentence. In SKILLDETECTIVE, we set a high score threshold (*i.e.*, 0.9), which means the flagged sentences contain inappropriate content with a very high probability. Since the toxicity ratings can be subjective in the real world, we then organize an internal focus group of three researchers, which includes two native English speakers, to manually verify that the contents detected by `Perspective` are indeed toxic or not. An output would be considered as toxic only if two or more researchers labeled it as toxic.

In addition to the above policies, there are more than 40 policies describing specific content/activities that are not allowed, such as to "claim to cure all diseases", "predict gender", or "build a bomb". These policies provide detailed examples thus enhancing our ability to detect violations. For these detailed policies, we first partition them into short phrases. For example, as to the policy "Includes references to or information regarding forced marriages or purchasable husbands and/or wives", we obtain three short phrases: "forced marriages", "purchase husband" and "purchase wife". For the policy "Purports to be able to predict gender", we generate a phrase "predict gender". Then, we use the `Spacy` library [6] to check whether these content or activities exist in skill outputs. For each skill output, we obtain nouns and verbs and compare whether they are similar to a policy. If the similarity is over a pre-defined threshold of 0.9, we consider the skill to contain a policy violation.

5 Implementation and Testing

SKILLDETECTIVE was implemented mostly in two different programming languages: Java and Python. The data-driven question analysis model was trained using a corpus of 18,641 question/answer pairs gathered from over 60,000 skill source code files taken from 1,437 skill templates mined from GitHub repositories. The bulk of the question/answer pairs was self-labeled by extracting the answers from the source code, *e.g.*, answers are nested within the "if" statements in the code. Over 1,300 questions from the corpus were hand labeled to train the classifier. Using the repository data, we were able to ascertain a set of representative skill structures that were used to enhance our skill navigation. The instantiations of SKILLDETECTIVE were run on a 2.6 GHz Quad-Core Apple Mac running macOS Catalina, as well as in Linux environments for concurrent skill testing. The development and experiments were conducted from Jan 2020 to Jan 2021.

Data collection and communication between SKILLDETECTIVE and the testing console (*i.e.*, skill simulator) was performed using the Selenium WebDriver [12]. Through SKILLDETECTIVE, we were able to test 54,055 Amazon Alexa skills and 5,583 Google actions. We found there are a large number of skills with the same name in the Alexa skill store (10,847 skills with duplicate names with others), so, we need to figure out which skill we are testing instead of invoking another skill with same name. For doing that, we extracted the unique Amazon product ID for each skill from the URLs of skill webpages. When testing skills, we recorded the skill ID at the same time. Based on the skill ID, we could obtain the skill's accurate developer, description, category and privacy policy information.

We gathered 518,385 total skill interactions. From this data, we applied our tool to infer the question type distribution, and the tool identified that 7.8% are Selection, 15.8% are Open-Ended, 8.7% are Binary, 8.2% are Instruction, 9.8% are Mixed, and 49.7% are Statements (NULL). We gathered approximately 2,070 unique audio files and 31,100 unique image files. We used the Python library `Speech-Recognition` [13] to translate the audio file into text. For audio processing during testing, we used the `Sphinx4` [39] library to transcribe audio files into text. To process image files, we used the Python library `pytesseract` [10] to extract texts from images.

Skill testing is a time consuming process due to the fact that every time it requires a fresh start for testing each interaction branch. We first tested all (as of Sep. 2020) skills in the Kids and Health categories. In our testing, we found a large number of skills asking for personal data in their first outputs (Details can be found in Figure 4). It makes sense because a skill tends to establish a rapport with the user in the earlier stages of the interaction and it is during this information gathering session that many policy violations occur. For the purpose of optimizing the total testing time, we designed SKILLDETECTIVE with two speeds: Fast and Slow. The Slow speed takes more time and explores as many nodes in the skill as possible, which was used to test skills in the Kids and Health categories, while the Fast speed spends less time on a skill and was used for the bulk of the analysis. We limit the depth SKILLDETECTIVE is allowed to go to 7 nodes down in the Fast mode. We have found that most policy violating materials show up within the first few nodes. Overall, SKILLDETECTIVE using the Slow speed spends an average of 14 minutes and 33 seconds exploring a skill, while SKILLDETECTIVE using the Fast speed spends an average of 5 minutes 13 seconds exploring a skill. The overall average skill exploration time is approximately 10 minutes per skill.

6 Evaluation Results

In this section, we present our results of policy violation detection and the performance of SKILLDETECTIVE. Table 14 in the Appendix presents a summary of our detection results. The detection results presented in tables in this section are true positives after our manual verification. With SKILLDETECTIVE, we identified 6,079 unique skills and 175 actions potentially violating at least one policy.

6.1 Identifying Policy-Violating Skills

6.1.1 Skills in Kids and Health categories

Based on the method described in Section 4.3.1, we tested all 3,617 Amazon skills and 108 Google actions in the Kids category (as of Sep. 2020). We identified 244 suspected policy-violating skills in the Kids category of the Amazon Alexa's skill store, and we did not find any policy-violating child-directed actions. Table 3 lists the breakdown of these policy-violating skills.

Since data collection is not allowed for skills in the Kids category, we first checked whether any kid skills collect any personal information and we found 34 skills that do collect personal information. 26 of the policy-violating skills ask for a user name while the others ask for age, birth date and

Policy Violation	# of Skills	Example
Collecting personal data	34	So, first, what is your name?
Directing users to outside of Alexa	21	Please support us by visiting www.oneoffcoder.com.
Explicit mature content	12	My bestie contains mature content that may not be suitable for all ages.
Requesting for positive rating	177	If you enjoyed playing kid chef, leaving us a five star review will help us add more content.
Toxic content	4	If I had a face like yours, I'd teach my ass to talk.
Violation in audios/images	4	Happy holidays santa's little helper here. Tell me your name to begin. (in audio)

Table 3: Detailed breakdown of 244 policy-violating skills in the Kids category of the Amazon Alexa's skill store.

location. For example, the skill "Spare Parts" asks "How old are you?", and the skill "Cake walk" would ask "When is your birthday?". We also found that 19 skills ask for user data in their first replies. It is puzzling that how these skills got approved because the policy violations can easily be captured during the skill vetting process. Next, we identified 21 kid skills that direct users to content/website outside of Alexa, and 12 skills explicitly claim they contain mature content. For examples, the skill "Math Whiz" replies that "please support us by visiting www.oneoffcoder.com", and the skill "Random picker" says "Random picker contains mature content that may not be suitable for all ages".

We also checked kid skills against other policies defined for general categories. We found 177 skills requesting a positive rating in the skill output or description. For example, the skill "Kids Aimal Sounds" says "If you liked this skill, please give us a 5 star rating". We identified 4 skills containing toxic content for kids by using the `Perspective` tool [8] and human verification. The skill "My Burns" will output "Are you always so stupid or is today a special occasion?" or "You're so ugly you'd scare the crap out of the toilet". Interestingly, we found 1 skill with data collection and 3 skills directing users to external websites in audio/image files.

For the Health category, we detected 146 skills out of 2,162 skills and 13 actions out of 227 actions with health-related data collection. Table 4 shows our detection results. 20 skills and 7 actions would collect data in their first replies. 82 skills request permissions (*i.e.*, using the provided permission APIs) to collect data. In addition, we found 13 skills not in the Health category but asking users for health information. For example, the skill "HealthDataGatherer" would ask "Hello, what is your blood pressure now? I will try to remember this", but the skill is in the "Productivity" category. After manually checking all the flagged skills/actions, we found 12 false positives (*i.e.*, achieving a detection accuracy of 92%). The false positives are because of some skills providing tips or facts such as "lower your blood pressure", which actually doesn't contain a policy violation.

6.1.2 Skills with data collection for general categories

For the general (*i.e.*, non-Kids and non-Health) categories, we identified 480 skills and 61 actions collecting personal data without using permission APIs (*i.e.*, they collect user data through the conversation interface). There are also 1,369 skills collecting personal data using permission APIs in the Amazon Alexa platform. Among all these skills with data collection, 623 skills and 25 actions potentially violate at last one policy, such as lacking a privacy policy, having an incomplete or deceptive privacy policy. Table 5 summarizes the results.

Policy Violation	# of Skills	# of Actions
Collecting health data	146	13
Collecting health data but not in the Health category	13	0
Lacking a disclaimer	1,709 (79%)	149 (66%)

Table 4: Policy violations in health-related skills. We tested all 2,162 skills and 227 actions in the Health category.

	Skills collect data through permission APIs	Skills collect data without using permission APIs	Action
Lacking a privacy policy	1	171	0
Having an incomplete privacy policy	330	104	8
Having a deceptive privacy policy	38	12	2
Should ask for permission	-	-	17
Total policy-violating skills	623		25

Table 5: Policy violations related to data collection for skills in general categories.

All of these skills and actions should provide a privacy policy to clearly disclose any involved data practices. However, for skills and actions collecting data without using permission APIs, we found 171 Alexa skills do not provide a privacy policy. All the 61 Google actions with data collection have provided a privacy policy. Next, we found 104 skills and 8 actions with an incomplete privacy policy. Finally, we checked deceptive privacy policies where skills claim they do not collect a specific type of data, but actually do. We found 12 skills and 2 action with deceptive privacy policies. For example, the skill "Diaper Duty" says "you can say your name" but claims "We never collect or share personal data with our skills" in its privacy policy. For the actions with data collection, 17 actions should have asked for permissions through APIs, but they do not. 9 actions ask for user names and another 8 actions ask for addresses.

For the 1,369 skills using permission APIs to collect user data, we also checked whether their privacy policies are accurate or not. Surprisingly, a skill named "Adopt A Pet" uses permission APIs to access "Device Country and Postal Code" data but doesn't provide a privacy policy (A screenshot is shown in Figure 5 in Appendix C). We also found that 330 skills request permissions but provide an incomplete privacy policy, and 38 skills provide a deceptive privacy policy. Another surprising observation is that, 2 skills in the Kids category and 82 skills in the Health category collect data with permissions, which is apparently not allowed in these two categories. An example is shown in Figure 6 in Appendix C. After manually checking the permissions and privacy poli-

cies of these 1,369 skills, our method correctly identified 891 skills with a complete privacy policy and 330 skills with an incomplete privacy policy. Other 147 skills were wrongly classified as false positives or false negatives. As a result, our approach achieves 90.5% precision, 94.3% recall, 92.4% F1 score and 89% accuracy.

6.1.3 Policy violations in skill descriptions

Both VPA platforms require health-related skills to provide a disclaimer in the description. However, we found only 453 Amazon skills among 2,162 skills (21%) provide a disclaimer. 78 Google actions among 227 actions (34%) provide a disclaimer and 66% of actions do not provide one. Interestingly, we also noticed that Amazon provides a sample disclaimer, and we found 89 skills and 16 actions using the exact sample in their descriptions. Other skills may use a disclaimer in their own words similar to the sample. After manually labeling all the disclaimers in health-related skills, we confirmed that our method achieved 95% precision, 97% recall, 96% F1-Score and 98% accuracy, which is higher than the results reported in VerHealth [37]. One possible reason is that VerHealth only uses 59 skills with a disclaimer for training and the number might be too small for training a neural network for detection.

As to the policy prohibiting "Explicitly requests that users leave a positive rating of the skill" in the Amazon Alexa platform, we found 3,452 violations in descriptions while only 21 potential violations appeared in the skill outputs. Due to the prevalence of this issue, we verified whether the Amazon Alexa platform really enforced restrictions against this policy by submitting skills violating this policy for certification. We submitted two new skills that requested a positive rating in the skill description, one was immediately rejected and the reason was "We do not allow skills to request positive reviews or 5-star ratings from users. Skills may only request reviews in a neutral manner". It is worth mentioning that the Amazon Alexa platform allows developers to choose "certify and publish now" or "certify now and publish later". In our submissions, we always chose "certify now and publish later" to get the certification results without publishing the skills. Therefore, our skills were never published in the skill store causing no harm to users. We also manually checked these skills and confirmed that all our results are correct. Thus, we identified 3,473 policy violations in 3,464 skills (9 skills have violation in both descriptions and output). We also found the developer “ButtonPushApps” developed 305 skills requesting a positive rating in the descriptions of all these skills.

6.1.4 Policy violations of content safety in skill outputs

For policies on content safety, we identified 177 Alexa skills with 208 potential toxic outputs after manually checking 554 outputs which were labeled as high toxicity by the Perspective tool [8]. We did not find any action with this issue. For example, the skill "My flames" outputs a steady stream of profanity that contains the phrases "teenage mutant ninja frog looking ass dirt eating ass dude", "beetle that collect and roll poop all day looking ass dude", "pee bathing in ass dude", "pissy Lizard looking ass dude", and "kangaroo looking ass dude". Another skill, "Sex Facts", only provides sexual content. It is surprising that these skills got approved through the vetting process and are now still available to the public. Three skills "Name Genie - (The Gender Predictor)", "Mayan Gender Predictor" and "Gender Predictor" try to predict the gender, which is not allowed by the Amazon Alexa platform. These skills are in different categories such as "Education & Reference", "Novelty & Humor" or "Games & Trivia".

6.1.5 Policy violations in audio/images

We have detected 8 skills with suspected policy violations hidden in the audio or image files. 4 of the skills (in the Kids category) were discussed earlier. Four non-kid skills contain data collection, while 2 of them lack a privacy policy and 2 more have incomplete privacy policies. The skill "Shape Game" asks "what is your name" in the audio file output, but does not provide a privacy policy. Although we have found few violations in external media, this type of analysis holds significant implications. First, dynamic media is becoming the new trend on skill marketplaces [15]. Popular content such as podcasts, pre-recorded audio streams, and image files are becoming common fixtures on VPA devices and provide many new hiding places for unscrupulous developers to hide policy-breaking content. This media type is dynamic and is almost always stored externally on the developer’s server. This makes testing this media a challenge and therefore it has been mostly ignored by existing work.

6.1.6 Potential influences of policy-violating skills

We investigate the potential influence of these policy-violating skills to users by measuring their usage and popularity (*i.e.*, ratings). Overall, we detected 6,079 skills and 175 actions violating different policies in total. 42% of these Alexa skills have at least one rating and it suggests they might have been invoked by real users. The average number of ratings is 63 for all the policy-violating skills. 13% skills have 5-star ratings and it shows some popular skills are violating policies. For example, the skill "Annoying sounds" has 12,796 ratings (it asks for a 5-star rating). The skill "Relaxing Sounds: Spa Music" in Health category lacks a disclaimer in its description and it has 9,255 ratings. As for the 175 Google actions, 67% of them have been rated by users and 21% actions have 5-star ratings. The action "Nutrition Facts", which is developed by Google and lacks a disclaimer in the Health category, has 1,435 user ratings and gets 4.6-star rating on average.

These popular suspected policy-violating skills could potentially influence thousands of users. It is interesting that we found there are critical user reviews complaining about skills that request for positive rating, *e.g.*, one user review "the program forced me to give a five star review" for skill

"Sleep Sounds: Box Fan Sounds", and another one "would have been 5 stars if it didn't keep asking me to review it" for skill "White Noise by Sleep Jar".

6.2 Performance Comparison

We conducted a performance comparison between SKILLDETECTIVE and SkillExplorer [27] on the question classification, answer prediction, and navigational capacity. We have fully implemented SkillExplorer's methodology from scratch. We did this using the paper as a road-map and made all attempts to accurately represent their work.

Performance of Question Classification. Our classifier in SKILLDETECTIVE was tested against the classification abilities of the SkillExplorer methodology. We conducted the test using 1,800 randomly chosen labeled questions with 300 questions per type including the NULL (statement) type. These 1,800 questions were randomly chosen from a dataset of hand labeled questions gathered from Github repositories. Each of the 300 per-type questions were fed into both SkillExplorer's and SKILLDETECTIVE's classifiers and measured for accuracy of classification. The test results concluded that our methodology performed more accurately overall, but especially on the Mixed and NULL classifications performing 11% and 26.3% better respectively. Table 6 shows the comparison results for each question-type followed by the percentage of accurate predictions achieved by both classifiers.

Question Type	SkillExplorer [27]	SKILLDETECTIVE
Open-Ended	97.3%	98%
Instruction	89.7%	99%
Binary	100%	100%
Selection	99%	99.3%
Mixed	87%	98%
NULL	71.3%	97.6%

Table 6: Performance of question classification.

Performance of Answer Prediction. We tested the performance of SKILLDETECTIVE's answer prediction against that of SkillExplorer as well as three other popular pre-built chatbots (Mitsuku [7], ChatScript [4], and Blender [11]). We found that the pre-built chatbots did not perform very well for most types of questions. The pre-built chatbots were able to answer Binary types of questions and to answer arithmetic questions such as "what is 2 + 2?" in many cases. However, they fell short in answering the other question types effectively. Note that we define an effective answer as one that continues the conversation with a skill, and an ineffective answer as one that stops or redirects the conversation of the skill. Ineffective answers are determined by specific skill outputs (*e.g.*, "Sorry, I don't understand. Can you please repeat that?"). Most existing chatbots today are designed to converse with people and to maintain context throughout the conversation, which is not suitable for skill interactions. SKILLDETECTIVE was designed to interact with skills, therefore it performed much better than the pre-built chatbots. Table 7 reports the question prediction results for SKILLDETECTIVE, SkillExplorer, and the max value of three popular chatbots conducted using 1,000 labeled questions with 200 representatives of each type. The questions were classified using the classifier and extracted from the dataset of questions gathered from GitHub. As shown in the table, the popular chatbots did not attain a max accuracy greater than 61%, while SkillExplorer did achieve over 90% on the Open-ended, Instruction, and Binary types. Only SKILLDETECTIVE was able to attain an accuracy over 90% on 4 types and 89% on the remaining type.

Question Type	SkillExplorer [27]	SKILLDETECTIVE	Popular Chatbots
Open-Ended	90%	90%	29%
Instruction	93%	95%	46%
Binary	100%	100%	61%
Selection	89%	94%	54%
Mixed	63%	89%	11%

Table 7: Performance of answer prediction.

Performance of Skill Navigation. To test the navigational capacity of SKILLDETECTIVE in the Slow mode, we used two sets of skills for this evaluation: 1) We traced the source-code of 64 skills taken from GitHub repositories to actual deployed skills on the Amazon skill store by manually checking the skill's functionality and comparing it to the source-code. We found that the average number of nodes in each of the 64 skills is 24 with a standard deviation (STD) of 9.48. 2) 100 skills with data collection behavior provided by the SkillExplorer research team [27]. We first manually traversed the skills for the purpose of mapping the interactions. We found through manual checking that the average number of nodes per skill is 40 with a STD of 13.26.

For the 64 skills with source code, SkillExplorer was able to traverse an average of 20.77 nodes (86.5% coverage rate) with a STD of 8.68. Using SKILLDETECTIVE, we were able to traverse an average of 20.95 nodes (87.3% coverage rate) with a STD of 9.22. All of the 64 skills tested did not contain any external audio sources. Also, these skills were found to be very simple (*i.e.*, fact skills). We would expect to have similar results when testing these simple skills. However, SKILLDETECTIVE outperforms SkillExplorer in handling complex skills. For the 100 skills with data collection behavior, SkillExplorer was able to traverse an average of 31.41 total nodes (78.5% coverage rate) with a STD of 13.55, and SKILLDETECTIVE was able to traverse an average of 36.83 nodes (92.1% coverage rate) with a STD of 12.54 using the Slow mode.

While testing the 100 skills with data collection behavior, we found 9 skills with policy violating content within the nodes traversed by SKILLDETECTIVE but missed by SkillExplorer [27]. For example, the skill "Baton Rouge Foodie" asks for the user's phone number after 5 interactions which includes an audio file. This policy violation was not detected using SkillExplorer due to it lacking the ability to process audio files. However, SKILLDETECTIVE was able to encounter this node and report the violation.

We also conducted an execution time performance comparison with SkillExplorer. We found that during testing of

the 164 skills, SkillExplorer on average spends approximately 11 minutes on each skill, while SKILLDETECTIVE spends approximately 14 minutes on each skill using Slow mode. This difference in time is due to the fact that SKILLDETECTIVE must perform tasks not applicable to SkillExplorer (*i.e.*, transcribe audio files). As to the performance statistics of tree traversal in SKILLDETECTIVE, the average depth of skill trees is 8.25 nodes, the average breadth of skill trees is 6.75 nodes, and the average fan of a skill tree is 40.25 nodes.

Performance of Policy Violation Detection. We tested both SkillExplorer and SKILLDETECTIVE on 100 skill names that were provided by the SkillExplorer research team and identified as containing data collection. We found through manual testing that only 61 of the 100 skills still contain data collection. This is most likely due to the fact that some of the skills have been removed from the skill store, while others may have had their content changed. We found that our policy violation detector can identify 54 of the 61 data collecting skills using the outputs from the manual testing. We also found that SKILLDETECTIVE detects 50 of the 61 skills that collect data using the interaction model in a fully automated manner, while the SkillExplorer methodology only found 41 of the 61 skills collecting data.

Skill-Tree Depth of Policy Violation Occurrences. For all the identified policy violations, we calculated the number of rounds of interactions it took for the policy-violating content to appear (*i.e.*, the skill-tree depth of the policy violation). We found that 244 skills respond with the policy-violating content in their first reply. There are 19 skills in the Kids category that ask for personal data within the first round of interactions. Also, there are 20 skills and 7 actions in the Health category that ask for personal data within the first round of interactions. The distribution of the skill-tree node depths of policy violations is shown in Figure 4.

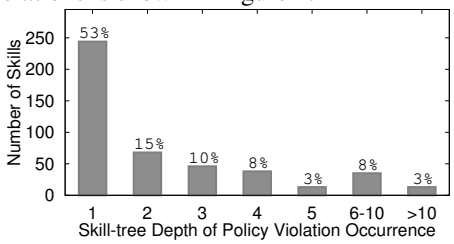

Figure 4: 78% of policy violations were captured within the first three rounds of interactions in our testing.

7 Limitation

SKILLDETECTIVE has several limitations. First, the concept of a chatbot is an ongoing research problem. Our methodology cannot handle all possible situations therefore limiting the traversal of some skills. We recognize that not all branches of a skill-tree can be mapped and as a consequence, some policy violations will be missed. Next, because of the intrinsic latency of skill testing we were unable to check every skill at full capacity using our Slow method. We did however find through testing that most policy violations happen within the first few nodes of skill interaction thereby reducing the effects of this limitation. We also conducted a manual validation of 100 skills which were randomly sampled from all categories in our dataset. We found 11 skills with policy violations after the manual analysis and our tool could detect 9 of these policy violations. A second limitation is that our policy violation detector is vulnerable to adaptive adversary evasion attacks [31, 40], *e.g.*, unscrupulous developers may rephrase policy-violating outputs without changing the semantic meaning to bypass SKILLDETECTIVE's detection. As for our future work, we would like to improve the robustness of SKILLDETECTIVE to detect stealthy policy violations.

A third limitation is that there are other policies (*e.g.*, providing misleading information in skills) that SKILLDETECTIVE does not check for. This is due to the complexity needed to accurately ascertain theses types of violations. In addition, SKILLDETECTIVE does not check skills that require an account linking to be accessed (*e.g.*, skills in the categories of SmartHome). Another limitation is that we implemented the chatbot using an FNN and a bag-of-words approach. We would like to recreate our methodology using more advanced sentence embedding such as BERT [24] in the future.

Lastly, there is a noticeable difference between the number of Amazon skills and Google actions tested. This is due to many difficulties put in place by Google to hinder automated interactions with their web-sites. First, Google prohibits the use of automated tools (like web drivers) to log into users accounts directly. To circumvent this issue, we used a third party site (Stackoverflow.com) that has a Google sign in option. We set the web driver to log into the the third party site under our developers credentials and subsequently navigate to the Action testing terminal. We were able to use this method for only a short time until Google patched this loop hole. As of the writing of this paper, no work around has been discovered thereby hindering future automated Google testing.

We hope to address these issues in future builds of the project by adding more robust navigational features and bettering the model in skill traversal as well as speed. We were able to find many different policy violations during our testing of Amazon's and Google's VPA platforms. The main reason is that the vetting process for skill certification was weak for both platforms. This weakness in vetting allows for developers (either knowingly or unknowingly) to get policy violating content certified and deployed to the public. Another reason might be that the content of live skills can be changed with no need to re-certify the skill. This ability to change content again allows developers to update their skills with policy-violating content.

8 Related Work

There has been considerable research on attacks [20, 21, 34, 35, 38, 41, 43, 45] against speech recognition systems and corresponding defenses [22, 25, 45, 46]. We mainly summarize recent research on security and privacy of skills.

Skill Invocation Hijacking via Speech Misinterpretation. Kumar *et al.* [29] presented the skill squatting attack. This attack utilizes linguistic ambiguities to exploit speech interpretation for the purpose of routing users to a malicious skill. In addition, paraphrased invocation names ("Citi Bank" vs "Citi Bank please") can be used to hijack sensitive skills [47]. This hijacking is due to the fact that the longest string match is used by the VA platform to invoke the skill. The masquerading attack was also presented in [47], where a malicious skill mimics the exit intent tricking the user into believing the skill has terminated, while still collecting user's input. LipFuzzer [48] is a black-box mutation-based fuzzing tool to systematically discover misinterpretation-prone voice commands in existing VPA platforms. LipFuzzer focused only on skill commands and their invocations. This differs greatly from our SKILLDETECTIVE in that our testing tool focuses on all layers of the skill and all outcomes of skill interaction.

Analysis of Voice Applications. Shezan *et al.* [36] developed a tool to identify sensitive voice commands by analyzing skills' descriptions. Liao *et al.* [32] found that many skills do not comply with the requirements defined by VPA platforms on the privacy policy by conducting a privacy policy analysis. Lentzsch *et al.* [30] conducted a large-scale analysis of Alexa skills, including identifying flaws in the vetting process, evaluating the efficacy of voice squatting techniques and privacy policies of skills. As to policy violations, the authors vetted privacy policies of 1,423 skills that request permissions for data collection, and found 23.3% of the privacy policies are not fully disclosing the data types associated with permissions requested by the skill. These works [30, 32, 36] are based on NLP analysis of skills' introduction pages on the store, without dynamically testing skills for the purpose of security and privacy analysis. They consider limited types of policies, and cannot detect policy violations related to skills' actual behavior. In SKILLDETECTIVE, 1,362 policy violations are owing to the dynamic analysis of skills. Cheng *et al.* [23] and Hu *et al.* [28] measured the skill vetting process on VPA platforms. Their results revealed that Amazon Alexa and Google Assistant platforms have not strictly enforced policy requirements. In [23], the authors also conducted a manual analysis to identify existing policy-violating skills in the current stores. Out of 755 Alexa skills under the Kids category, 31 skills were identified as problematic skills with policy violations.

More recently, to facilitate automated analysis of skills on a large scale, dynamic testing tools [27, 37] have been developed. SkillExplorer [27] is a grammar-rule based testing tool to automatically explore skills' behaviors. The authors tested 28,904 Amazon skills and 1,897 Google actions, and identified 1,141 skills request users to provide personal information without disclosing in their privacy policies. SkillExplorer mainly focuses on identifying skills that collect private information, without evaluating skills' conformity to other policy requirements. The authors in VerHealth [37] only analyzed 813 health-related skills on the Amazon Alexa platform, and VerHealth adopts a rather simple interaction model to collect limited responses from skills. We also achieved a higher detection accuracy compared with prior works. When detecting incomplete privacy policies for skills with permissions, our method achieves 90.5% precision, which is higher than the result reported in [30] (the authors in [30] only provided the precision result). For detecting whether health-related skills lacks a disclaimer or not, our methods achieved 95% precision, 97% recall, 96% F1-score and 98% accuracy, which are higher than results reported in VerHealth [37].

Research Work	Skills Analyzed	Actions Analyzed	Policies Considered	Problematic Skills Released?
SkillExplorer [27] (USENIX SEC'20)	**28,904**	**1,897**	1 (Data collection)	Released 100 skills with data collection
VerHealth [37] (UbiComp'20)	**813**	**0**	5 (Health-specific)	✗
SKILLDETECTIVE (Our work)	**54,055**	**5,583**	50+	Fully released

Table 8: Comparison of SKILLDETECTIVE with related works that detect policy violations through dynamic testing.

Distinctive from SkillExplorer and VerHealth, our work systematically evaluates skills' conformity to more than 50 policy requirements through a comprehensive dynamic analysis of skills. Table 8 shows the comparison of SKILLDETECTIVE with SkillExplorer [27] and VerHealth [37]. We tested 54,055 Alexa skills and 5,583 Google actions (much larger than their analysis), and SKILLDETECTIVE's policy violation detector covers more than 50 policies. In addition, both works do not release the detailed list of the identified policy-violating skills. The lack of such information makes it hard to conduct a sound performance comparison among them. We share details of our testing results to facilitate future research.

9 Conclusion

In this work, we designed and implemented an interactive testing tool named SKILLDETECTIVE, which aims at identifying policy-violating skills in current VPA platforms. Using SKILLDETECTIVE, we tested 54,055 Amazon Alexa skills and 5,583 Google Assistant actions, and collected 518,385 different pieces of skill interaction data. We were also able to capture 2,070 audio files and 31,100 image files from skills during our testing. We identified 6,079 skills and 175 actions that contain at least one suspected policy violation. SKILLDETECTIVE can also be deployed by VPA platform providers to proactively identify policy-violating skills at the submission phase and prevent them from being published. Our findings in this paper could help Amazon and Google improve their skills' policy compliance, and we believe this work will have a great potential to positively impact the VPA ecosystem.

Acknowledgment

We thank our shepherd, Dr. Anupam Das, and the anonymous reviewers for their valuable comments. This work is supported in part by National Science Foundation (NSF) under the Grant No. 2120369, 2114920, 2114982, 2031002, and 1846291.

References

[1] Alexa Skills Policy Testing. https://developer.amazon.com/fr/docs/custom-skills/policy-testing-for-an-alexa-skill.html. [Accessed: 25-Nov-2020].

[2] Alexa Skills Privacy Requirements. https://developer.amazon.com/fr/docs/custom-skills/security-testing-for-an-alexa-skill.html#25-privacy-requirements. [Accessed: 25-Nov-2020].

[3] Alexa Skills Security Requirements. https://developer.amazon.com/fr/docs/alexa-voice-service/security-best-practices.html. [Accessed: 25-Nov-2020].

[4] Brillig Understanding, Inc. http://brilligunderstanding.com/rosedemo.html.

[5] Celebrating 100,000 Alexa Skills. https://developer.amazon.com/en-US/blogs/alexa/alexa-skills-kit/2019/09/congratulations-alexa-skill-builders-100-000-skills-and-counting. [Accessed: 25-Nov-2020].

[6] Industrial-Strength Natural Language Processing. https://spacy.io.

[7] Mitsuku Chatbot. http://www.square-bear.co.uk/mitsuku/home.htm.

[8] Perspective. https://www.perspectiveapi.com/home.

[9] Policies for Actions on Google. https://developers.google.com/actions-/policies/general-policies. [Accessed: 25-Nov-2020].

[10] pytesseract 0.3.7. https://pypi.org/project/pytesseract/.

[11] Recipes for building an open-domain chatbot. https://parl.ai/projects/recipes/.

[12] Selenium WebDriver. https://www.selenium.dev.

[13] SpeechRecognition 3.8.1. https://pypi.org/project/SpeechRecognition/.

[14] Stanford CoreNLP – Natural language software . https://stanfordnlp.github.io/CoreNLP/. [Accessed: 25-Nov-2020].

[15] Virtual Personal Assistants (VPA) and Smart Speaker Market: Artificial Intelligence Enabled Smart Advisors, Intelligent Agents, and VPA Devices 2020 - 2025. https://www.researchandmarkets.com/reports/5125038/virtual-personal-assistants-vpa-and-smart. [Accessed 6-8-2021].

[16] What is Considered PHI Under HIPAA? https://www.hipaajournal.com/considered-phi-hipaa/.

[17] Benjamin Andow, Samin Yaseer Mahmud, Wenyu Wang, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Tao Xie. Policylint: Investigating internal privacy policy contradictions on google play. In *Proceedings of the 28th USENIX Conference on Security Symposium*, page 585–602, 2019.

[18] Jan A Botha, Emily Pitler, Ji Ma, Anton Bakalov, Alex Salcianu, David Weiss, Ryan McDonald, and Slav Petrov. Natural language processing with small feed-forward networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2879–2885, 2017.

[19] Travis D Breaux, Hanan Hibshi, and Ashwini Rao. Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements. *Requirements Engineering*, 19(3):281–307, 2014.

[20] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *USENIX Security Symposium (USENIX Security)*, pages 513–530, 2016.

[21] Guangke Chen, Sen Chen, Lingling Fan, Xiaoning Du, Zhe Zhao, Fu Song, and Yang Liu. Who is real bob? adversarial attacks on speaker recognition systems. In *IEEE Symposium on Security and Privacy (SP)*, 2021.

[22] S. Chen, K. Ren, S. Piao, C. Wang, Q. Wang, J. Weng, L. Su, and A. Mohaisen. You can hear but you cannot steal: Defending against voice impersonation attacks on smartphones. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 183–195, 2017.

[23] Long Cheng, Christin Wilson, Song Liao, Jeffrey Young, Daniel Dong, and Hongxin Hu. Dangerous skills got certified: Measuring the trustworthiness of skill certification in voice personal assistant platforms. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2020.

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[25] Huan Feng, Kassem Fawaz, and Kang G. Shin. Continuous authentication for voice assistants. In *Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 343–355, 2017.

[26] Yoav Goldberg. *Neural Network Methods for Natural Language Processing*. Morgan & Claypool, 2017.

[27] Zhixiu Guo, Zijin Lin, Pan Li, and Kai Chen. Skillexplorer: Understanding the behavior of skills in large scale. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 2649–2666, 2020.

[28] Hang Hu, Limin Yang, Shihan Lin, and Gang Wang. A case study of the security vetting process of smart-home assistant applications. In *Proceedings of IEEE Workshop on the Internet of Safe Things (SafeThings)*, 2020.

[29] Deepak Kumar, Riccardo Paccagnella, Paul Murley, Eric Hennenfent, Joshua Mason, Adam Bates, and Michael Bailey. Skill Squatting Attacks on Amazon Alexa. In *27th USENIX Security Symposium (USENIX Security)*, pages 33–47, 2018.

[30] Christopher Lentzsch, Sheel Jayesh Shah, Benjamin Andow, Martin Degeling, Anupam Das, and William Enck. Hey Alexa, is this skill safe?: Taking a closer look at the Alexa skill ecosystem. In *Proceedings of the 28th ISOC Annual Network and Distributed Systems Symposium (NDSS)*, 2021.

[31] J Li, S Ji, T Du, B Li, and T Wang. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium*, 2019.

[32] Song Liao, Christin Wilson, Long Cheng, Hongxin Hu, and Huixing Deng. Measuring the effectiveness of privacy policies for voice assistant applications. In *Annual Computer Security Applications Conference (ACSAC)*, page 856–869, 2020.

[33] Erika McCallister. *Guide to protecting the confidentiality of personally identifiable information*, volume 800. Diane Publishing, 2010.

[34] Nirupam Roy, Sheng Shen, Haitham Hassanieh, and Romit Roy Choudhury. Inaudible voice commands: The long-range attack and defense. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pages 547–560, 2018.

[35] Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. In *Network and Distributed System Security Symposium*, 2019.

[36] Faysal Shezan, Hang Hu, Jiamin Wang, Gang Wang, and Yuan Tian. Read between the lines: An empirical measurement of sensitive applications of voice personal assistant systems. In *Proceedings of The Web Conference (WWW)*, 2020.

[37] Faysal Hossain Shezan, Hang Hu, Gang Wang, and Yuan Tian. Verhealth: Vetting medical voice applications through policy enforcement. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2020.

[38] Tavish Vaidya, Yuankai Zhang, Micah Sherr, and Clay Shields. Cocaine noodles: Exploiting the gap between human and machine speech recognition. In *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, 2015.

[39] Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. Sphinx-4: A flexible open source framework for speech recognition. Technical report, USA, 2004.

[40] Boxin Wang, Hengzhi Pei, Boyuan Pan, Qian Chen, Shuohang Wang, and Bo Li. T3: Tree-autoencoder regularized adversarial text generation for targeted attack. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6134–6150, 2020.

[41] Qiben Yan, Kehai Liu, Qin Zhou, Hanqing Guo, and Ning Zhang. Surfingattack: Interactive hidden attack on voice assistants using ultrasonic guided wave. In *Network and Distributed Systems Security (NDSS) Symposium*, 2020.

[42] Le Yu, Xiapu Luo, Xule Liu, and Tao Zhang. Can we trust the privacy policies of android apps? In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 538–549. IEEE, 2016.

[43] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, XiaoFeng Wang, and Carl A. Gunter. Commandersong: A systematic approach for practical adversarial voice recognition. In *USENIX Conference on Security Symposium (USENIX Security)*, pages 49–64, 2018.

[44] L. Yujian and L. Bo. A normalized levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1091–1095, 2007.

[45] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. Dolphinattack: Inaudible voice commands. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 103–117, 2017.

[46] Linghan Zhang, Sheng Tan, Zi Wang, Yili Ren, Zhi Wang, and Jie Yang. Viblive: A continuous liveness detection for secure voice user interface in iot environment. In *Annual Computer Security Applications Conference (ACSAC)*, page 884–896, 2020.

[47] Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFeng Wang, Yuan Tian, and Feng Qian. Understanding and mitigating the security risks of voice-controlled third-party skills on amazon alexa and google home. In *IEEE Symposium on Security and Privacy (SP)*, 2019.

[48] Yangyong Zhang, Lei Xu, Abner Mendoza, Guangliang Yang, Phakpoom Chinprutthiwong, and Guofei Gu. Life after speech recognition: Fuzzing semantic misinterpretation for voice assistant applications. In *Network and Distributed System Security Symposium (NDSS)*, 2019.

Appendix A Similarity Calculation for Question Classification

To calculate the similarity of a labeled question *n* to a target question α, we count the frequency of all specific conjoined PoS (parts-of speech) patterns created from *n* that appear in α. Table 9 shows an example of a corpus of questions (with labeled question types). Suppose we want to classify the question "How old is your mother?", which has a PoS representation of WRB (Wh-adverb), JJ (Adjective), VBZ (Verb, 3rd person singular present), PRP (Personal pronoun), and NN (Noun, singular or mass). We start with the first PoS entry in the corpus, which is WRB, JJ, VBP, PRP ((Wh-adverb), (Adjective), (Verb, non-3rd person singular present), (Personal pronoun)), and corresponds to the question "How old are you?". Next, we count the number of occurrences of every pattern permuted from each conjoined tag in the corpus entry. Let's say the tag WRB is located at index 1 and JJ is located at index 2 and so on. We make our patterns by first starting off with 1, 2, 3, and 4 individually. Next we use (1,2), (2,3), (3,4). We then use (1, 2, 3), (2, 3, 4) and so on until all conjoined patterns have been created. In total from this corpus entry we get 10 different patterns. We count the number of times that each pattern shows up in our target question. For this example, we get 4 patterns {WRB, JJ, PRP, and (WRB, JJ)} that show up in the target question. We then calculate the similarity by taking the aggregate by dividing the number of corpus entry patterns located inside the target question by the total number of patterns that can be derived from the target question. This process is continued until all entries in the corpus have been compared with the target question. Finally, the class of the corpus entry with the highest similarity score will be chosen as the classification of the question. For this example, we get the similarity scores of 0.4, 0.07, and 0.1. The question with the highest similarity is "how old are you?", so we return the classification as "open-ended" which is correct. We determine from this classification how we will answer the question "how old is your mother?"

Question	PoS Tags	Question Type
How old are you?	WRB, JJ, VBP, PRP {(Wh-adverb), (Adjective), (Verb, non-3rd person singular present), (Personal pronoun)}	Open-ended
Do you want to go?	VBP, PRP, VB, TO, VB {(Verb, non-3rd person singular present), (Personal pronoun), (Verb, base form), (to), (Verb, base form)}	Binary
Can you stay for a while?	MD, PRP, VB, IN, DT, NN {(Modal), (Personal pronoun), (Verb, base form), (Preposition or subordinating conjunction), (Determiner), (Noun, singular or mass)}	Binary

Table 9: Example of PoS tagging and similarity calculation for question classification.

Appendix B Vector Representation of Questions in FNN Model

To vectorize our data and extract their features, we utilize PoS tagging of the corpus to extract the nouns of each question. These nouns form the representative keywords, which together form a bag of words (BoW). We next create our feature vector by calculating the average Levenshtein Distance (LD) [44] between each individual keyword in the BoW to every word in our target question. In other words, we read the first keyword in the BoW and find the average LD of that keyword to every word within the question we wish to vectorize. When the similarity is calculated, the vector is updated and the process is continued. An example is provided in Table 11, which shows a vectorized version of the corpus in Table 10. The BoW for this corpus are [number, letters, D, letter, right, way] as determined by the Stanford Core-NLP Package (the word "left" was not recognized as a none by Core-NLP). Each entry in Table 11 shows the value of the average LD from the keyword to every word in the corresponding question in the corpus. By using this method, the corpus is vectorized to be used as a training set for the neural model.

Question	Answer
Pick a number between 1 and 4.	[1, 2, 3, 4]
Choose from letters A – D. Which one do you want?	[A, B, C, D]
Pick a letter from A to D.	[A, B, C, D]
You can go left or right. Which way do you want to go?	[right, left]

Table 10: Corpus of question/type pairs.

Number	Letters	D	Letter	Right	Way
0.16	0.10	0.05	0.11	0.03	0.07
0.05	0.13	0.15	0.12	0.04	0.10
0.05	0.14	0.14	0.17	0.03	0.10
0.01	0.07	0.04	0.08	0.15	0.16

Table 11: Vector representation of the corpus in Table 10.

Category	Policy Index	Policy Violation	Policy Defined by VPA Platforms
Kids	P1	Collecting kids data	It collects any personal information from end users.
	P2	Directing users to outside of Alexa	It promotes any products, content, or services, or directs end users to engage with content outside of Alexa.
	P3	Explicit mature content	It includes content not suitable for all ages.
Health	P4	Collecting health data	Collects information relating to any person's physical or mental health or condition.
	P5	Lacking a disclaimer in the skill description	Is a skill that provides health-related information, news, facts or tips and does not include a disclaimer in the skill description stating that the skill is not a substitute for professional medical advice.
Personal data collection	P6	Lacking a privacy policy	If your app collects personal data from end users, your app must display a legally adequate privacy policy within the app and on the app detail page.
	P7	Incomplete privacy policy	The privacy policy must comprehensively disclose what personal data your app collects, how it is used, and the types of parties with whom it is shared.
	P8	Deceptive privacy policy	Ensure that your collection and use of that data complies with your privacy policy.
	P9	Should ask for permission	Google: Request sensitive user data via the Permissions API (All requests for a user's location and name).
General (non-Kids, non-Health)	P10	Requesting for positive rating	Explicitly requests that users leave a positive rating of the skill.
	P11	Toxic content	It includes content not suitable for all ages.
	P12	Predicting gender	Purports to be able to predict gender.

Table 12: The types of policy violations in existing skills captured by SKILLDETECTIVE (*i.e.*, there exists at least one skill violating one of these 12 policies).

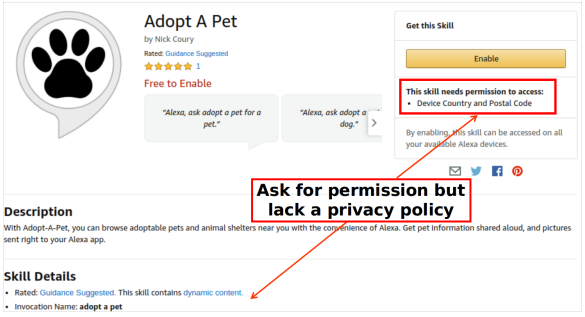

Figure 5: A skill named "Adopt A Pet" collects user data through the permission APIs but lacks a privacy policy.

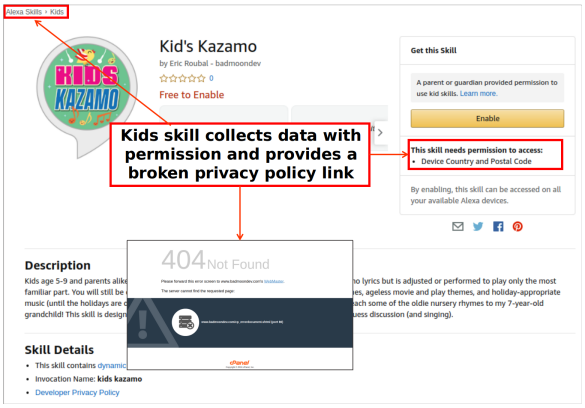

Figure 6: A kid skill named "Kids's Kazamo" collects user data through the permission APIs and also provides a broken privacy policy. Apparently, data collection in a kid skill is not allowed according to Amazon Alexa's policy requirements.

Appendix C Examples of Policy Violations

Appendix D Types of Policy Violations

Provide life-saving assistance, Cure all diseases, Black market sale, Prescription drugs, Offers a separate skills store, Recommend skills, Offer compensation for using skills, Solicit donations, Extreme gore, Decapitations, Unsettling content, Excessive violence, Organized crime, Terrorism, Illegal activities, Forced marriages, Purchasable husbands, Purchasable wives, Promote hate speech, Incite racial hatred, Incite gender hatred, Nazi symbols, Promote Ku Klux Klan, Contact emergency responder, Contact 911, Illegal downloading, Pirated software, Join illegal organization, Illegal lifestyle, Prostitution, Create dangerous materials, Build bomb, Build meth lab, Build silencer, Promote terrorism, Praise terrorism, Recruit members for terrorist, Promote gambling, Excessive alcohol, Underage alcohol

Table 13: Policies that are considered in SKILLDETECTIVE, but we didn't find any skill violating them in our testing. This table only shows keywords of these polices. The complete policy descriptions can be found in [1–3, 9].

	Policy Violation Type	# of Skills	# of Actions	Example of Policy Violation		
				Skill ID	Skill Name	Skill Output
Kids	P1: Collecting kids data	34	0	B07DFCXXM5	Say Please	Please tell me your name.
	P2: Directing users to outside of Alexa	21	-	B087Z4F96J	Math Whiz	Goodbye! Please support us by visiting www.oneoffcoder.com.
	P3: Explicit mature content	12	0	B07R97WNFR	My bestie	My bestie contains mature content that may not be suitable for all ages.
	P10: Requesting for positive rating	177	-	B07BF956XW	Name, Place, Animal, Thing	Don't forget to leave a 5 star review in the Alexa app store.
	P11: Toxic content	4	0	B0837HWNY5	My burns	You're so ugly you'd scare the crap out of the toilet.
	Total policy-violating skills	244	0			
Health	P4: Collecting health data	146	13	B07N626993	Blood donation helper	Please tell your blood group followed by rh factor.
	P4: Collecting health data (Not in health category)	13	0	B0829RDY2T	HealthDataGatherer	Hello, what is your blood pressure now?
	P5: Lacking a disclaimer	1,709	151	B01JG73CJI	spare the air	Description: Spare the air. Improve the quality of air by using these simple tips.
	Total policy-violating skills	1,790	155			
Collecting data protected by permission models	P6: Lacking a privacy policy	1	-	B0746FHXLY	Adopt A Pet	Permission: Device Country and Postal Code. No privacy policy
	P7: Incomplete privacy policy	330	-	B07CLN9Q8T	Sneeze Forecast	Permission: Device Country and Postal Code.
	P8: Deceptive privacy policy	38	-	B07NGT7LG4	You Choose	Permission: Device Address. Privacy policy: We never collect or share personal data with our skills.
	Total policy-violating skills	360	-			
Collecting data not protected by permission models	P6: Lacking a privacy policy	171	0	B07N41L2FK	paul's demo	Hello, what is your name?
	P7: Incomplete privacy policy	104	8	B01MTD0S8B	Genio	Hey yes, what is your phone number?
	P8: Deceptive privacy policy	12	2	B08DXQJWM7	Diaper Duty	Output: You can say your name. Privacy policy: We never collect or share personal data with our skills.
	P9: Should ask for permissions	-	17	B07Y73QN1M	seven wonders	What is your name again?
	Total policy-violating skills	280	25			
General	P10: Requesting for positive rating	3,464	-	B07DC3C4SN	Marriage Jokes	Leaving us a five star review in the Alexa store, will help us add more jokes.
	P11: Toxic content	177	0	B082FR65LQ	Roast me	you are so fat and you look like a mommas boy.
	P12: Predicting gender	3	-	B07JPLTR31	Gender Predictor	Would you like Alexa to predict the gender of your baby?
	Total policy-violating skills	3,640	0			
Media (audios or images)	P1: Collecting kids data	1	0	B077LC3R91	Santa's Helper	Tell me your name to begin.
	P2: Directing users to outside of Alexa	3	-	B085CMV7ZB	Junior Science Bee	Please register for the regional contest in www.northshore.org 124 players can play.
	P6: Lacking a privacy policy	2	0	B07TZM51S4	Shape Game	What's your name?
	P7: Incomplete privacy policy	2	0	B07WVPNJ3C	My Biorhythms	When were you born?
	Total policy-violating skills	8	0			
Total number of policy violations		**6,079**	**175**			

Table 14: Summary of policy violations identified by SKILLDETECTIVE. Based on the 13 policies listed in Table 12, we detected 6,079 unique skills and 175 actions violating at least one policy. "-" means Amazon or Google doesn't have this policy and "0" means platforms have defined the policy but we did not find a skill/action violating this policy. An example of policy-violating skill output (with skill ID and skill name) in each category is also shown in this table. The detection results are true positives after our manual verification. The complete list of policy-violating skills can be found at https://github.com/skilldetective/skilldetective.