

WhisperVoiceTrace: A Comprehensive Analysis of Voice Command Fingerprinting

Minji Jo*
minlit@ewhain.net
Ewha Womans University
Seoul, South Korea

Hyojin Kim*
222aig14@ewhain.net
Ewha Womans University
Seoul, South Korea

Jiwoo Hong
hjiwoo0914@ewhain.net
Ewha Womans University
Seoul, South Korea

Hosung Kang
kanghsung717@ewhain.net
Ewha Womans University
Seoul, South Korea

Nate Mathews
nate.mathews@mail.rit.edu
Rochester Institute of Technology
Rochester, NY, USA

Se Eun Oh†
seoh@ewha.ac.kr
Ewha Womans University
Seoul, South Korea

ABSTRACT

Smart speakers such as Amazon Alexa or Google Home, have significantly enhanced the convenience and efficiency of our daily lives, leading to widespread adoption globally. Subsequently, the voice commands directed at these devices contain a wealth of private information, encompassing users' daily routines, details about clinic visits, and even shopping habits.

Security researchers have already identified vulnerabilities in encrypted network traffic, particularly in the context of traffic analysis, enabling the network-level adversaries to infer the websites visited by users. This security threat, common in traditional web communication, extends to voice query communication between the smart speaker and the server, giving rise to the concept of Voice Command Traffic Fingerprinting (VCF).

In this paper, building upon recent literature on VCF, we strive to conduct a more comprehensive study. Our contributions include proposing a variant of WF features that empowers adversaries to outperform state-of-the-art models. Additionally, we consider variations in voice command traffic influenced by factors such as device locations and user characteristics. Furthermore, we introduce a command category fingerprinting approach capable of inferring the voice command category even for unknown commands, presenting a more practical and scalable model.

Through these contributions, we aim to advance the understanding of VCF by considering various practical factors influencing voice command traffic, and provide insights for the development of more robust defenses.

CCS CONCEPTS

• **Security and privacy** → **Network security**; • **Computing methodologies** → **Machine learning**; **Neural networks**; • **Networks** → **Network privacy and anonymity**;

*Equally credited authors.
†A corresponding author.



This work is licensed under a Creative Commons Attribution International 4.0 License.

ASIA CCS '24, July 1–5, 2024, Singapore, Singapore

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0482-6/24/07.
https://doi.org/10.1145/3634737.3657017

KEYWORDS

Traffic Analysis, Voice Assistant Systems, Voice Command Fingerprinting

ACM Reference Format:
Minji Jo, Hyojin Kim, Jiwoo Hong, Hosung Kang, Nate Mathews, and Se Eun Oh. 2024. WhisperVoiceTrace: A Comprehensive Analysis of Voice Command Fingerprinting. In *ACM Asia Conference on Computer and Communications Security (ASIA CCS '24), July 1–5, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3634737.3657017

1 INTRODUCTION

Voice assistant systems have gained widespread adoption across diverse sectors, including electronics, healthcare, automotive, and retail, owing to their inherent convenience. Notably, the market for voice-activated smart speakers has seen continual growth, propelled by the integration of built-in voice assistant systems. These virtual assistants adeptly understand and respond to user queries and commands, facilitating the seamless handling of daily tasks. This capability is made feasible through advancements in Natural Language Processing (NLP) and machine learning. A survey [1] reveals that Amazon Alexa commands the highest popularity among smart speaker users, with approximately 65% market share, followed by Google Home with a user base of 20%.

As these smart speakers extend their functionality to encompass diverse activities such as making phone calls, scheduling, and even placing grocery orders, they become increasingly integrated into our daily lives. However, the advantages of convenience and efficiency inherent in these systems come with a trade-off, as the voice commands transmitted by users over these speakers convey substantial amounts of private information, giving rise to significant privacy concerns.

Security researchers have voiced apprehensions about encrypted network traffic, which, even when secure, exposes metadata such as traffic volume and timing information. This leakage can potentially allow adversaries to infer users' online activities, including details about the websites visited and videos streamed. One such security threat is Website Fingerprinting (WF) [34, 35, 26, 27, 25, 29, 32, 12], where a global network adversary observes encrypted network traffic and employs machine learning models to discern communication destinations, thereby identifying visited websites.

This type of attack is also relevant to voice queries and responses over smart speakers. Despite the encryption of communication between the user and cloud servers, an adversary can still identify voice commands by analyzing encrypted network traffic patterns, akin to WF attacks. We call this Voice Command Fingerprinting (VCF) and the security concern posed by this attack has been explored in recent literature [10, 33, 17].

The VCF model proposed by Wang et al. [33] first introduced a deep learning-based approach that employed an ensemble of three distinct models: Convolutional Neural Network (CNN), Stacked Autoencoder (SAE), and Long Short-Term Memory (LSTM). This ensemble was designed to automate feature engineering, with a primary focus on incoming traffic from the server. Remarkably, their model achieved an impressive accuracy of 93%.

Subsequently, Hyland et al. [17] presented a refined version of the VCF model by incorporating the Deep Fingerprinting (DF) model architecture [32]. The DF model is recognized as one of the most robust WF models available. This enhancement aimed to further bolster the performance and capabilities of the VCF model in the context of voice command identification.

More recently, Ahmed et al. [10] contributed to the evolution of VCF models by showcasing an end-to-end approach to VCF attacks. Their approach involves two distinct stages: wake-word invocation detection and voice command identification. Noteworthy is the fact that their work represents a pioneering effort, as it is the first to consider a threat model that observes all traffic emanating from a smart home, rather than solely focusing on voice command traffic directed at the smart speaker. This broader perspective provides a more comprehensive understanding of potential vulnerabilities and attack vectors in smart home environments.

Still, the practical feasibility of VCF attacks remains uncertain due to the unrealistic scenarios assumed in existing literature.

Firstly, variations in voice command traffic over smart speakers can be linked to device locations, and users' characteristics, such as gender and demographics. However, these factors have not been thoroughly investigated in previous work.

Secondly, existing VCF models lack the capacity to fingerprint all voice commands globally, highlighting a scalability issue. Addressing this challenge requires the development of additional strategies.

Lastly, as the VCF adversary monitors communication between the smart speaker and the server during voice command transmission and corresponding responses, it's crucial to note that voice command traffic is influenced by both initiation and responses. Traces consistently show a higher volume of incoming packets than outgoing packets, emphasizing the pivotal role of responses in shaping overall traffic patterns. Further investigation into the correlation between responses and their traffic patterns is needed, necessitating exploration of diverse experimental scenarios impacting the nature of responses.

In this paper, we present the WhisperVoiceTrace (WhiVo) model as an extension of the previous VCF model [17] and the Traffic Aggregation Matrix (TAM) [30]—a robust feature representation for WF. Notably, our focus in this study involves an in-depth exploration of feature analysis to tailor the TAM for VCF. As a result of this analysis, we introduce the Volume Aggregation Vector (VAVs), a variant of TAM that consolidates traffic volume information from small trace segments constituting the overall trace. This enhancement aims to improve the efficacy of our model in capturing the nuanced patterns present in voice command traffic for more accurate and robust fingerprinting.

To the best of our knowledge, we have collected the largest voice command traffic dataset, encompassing 600 unique commands and a total of 453,350 traces. Our data collection process was designed to be comprehensive, incorporating diverse genders and demographics to ensure a more realistic simulation of users. Additionally, we considered various user locations, configured either manually or automatically on the Alexa device, in order to accurately reflect the global adoption of Alexa across different regions and environments. And, finally, we include samples from Google assistant for further comparison.

In the context of an infinite corpus of voice commands, we extend our proposal to incorporate voice command category fingerprinting, leveraging clustering algorithms to automate the discovery of clusters. The rationale behind this extension is that inferring the category alone could pose a significant privacy threat. Moreover, this approach exhibits improved scalability, as new commands can feasibly be grouped into existing categories, eliminating the need to continually introduce new categories. This adaptation allows for a more efficient and sustainable scalability model, acknowledging the likelihood of new commands sharing commonalities with established categories.

To this end, we summarize our contributions as follows:

- We conducted an extensive feature analysis, incorporating traditional WF feature sets and TAM representations, to formulate a VCF-specific feature set. This study encompassed the exploration of various TAM variants.
- The proposed WhiVo model demonstrated superior performance compared to the state-of-the-art [17, 33] across various VCF datasets [17, 33], achieving a 8% margin improvement at its best.
- We conducted a fingerprintability analysis, taking into account factors such as location, human characteristics, and command features, to identify advantageous settings for potential VCF attacks.
- We introduced a voice command category fingerprinting approach capable of inferring the content of voice commands with 86% accuracy. This method shows promise in detecting unknown commands by associating them with existing categories.

Through these contributions, we aim to provide a more comprehensive exploration of VCF, taking into account various practical factors that influence voice command traffic. This approach is crucial for gaining a better understanding of the fundamental causes of information leakage on voice command traffic, ultimately contributing to the development of effective defense mechanisms.

2 RELATED WORK

In recent years, a lot of research has been devoted to classifying encrypted network traffic, also known as traffic fingerprinting. These techniques can be used to extract potentially sensitive information

using network traffic patterns, representing a type of privacy attack. In the following section, we discuss key related works in the domain of privacy-violating traffic fingerprinting attacks.

2.1 Website Fingerprinting

The primary aim of website fingerprinting is to discern what websites a user has visited by analyzing the encrypted traffic generated by those visited sites. Previous studies have proposed website fingerprinting by using traditional machine learning models with hand-crafted features. Panchenko et al. [26] analyzed the traffic as the cumulative sum of packet length. This attack used a Support Vector Machine as a classifier and achieved high accuracy in identifying websites. Similarly, Shen et al. [31] use packet length information to fingerprint fine-grained, website subpage labels using a random forest classifier. Hayes et al. [15] analyze a large variety of traffic feature representations and showed that the most important feature was the number of incoming packets.

These feature-based privacy attacks generally showed relatively simple traffic features could produce effective results. However, more recent research using deep learning models has shown that greater performance and improved robustness to noise are achievable by using models with *learnable* feature extraction stages. Rimmer et al. [29] were the first to demonstrate that deep neural networks are superior to traditional machine learning methods in the field of website fingerprinting. CNN, LSTM, and SAE were used. A CNN trained by Sirinam et al. [32] on the sequence of packet directions was shown to defeat traffic padding defenses that were previously believed to be adequate to protect against website fingerprinting attacks. Rahman et al. [28] improved upon this study by adding raw traffic timestamp information into the model's input representation. At the time of writing, Shen et al. [30] proposed a novel traffic representation that aggregates packet direction counts within short time windows within the traffic. This attack achieves an average accuracy improvement of 8.9% over the prior state-of-the-art attack.

2.2 Voice Fingerprinting

Kennedy et al. [20] were the first to examine a voice command fingerprinting attack on smart home speakers, specifically Alexa. They applied traditional machine learning to analyze encrypted voice traffic on smart speakers. They collected a dataset of 100 real-world classes and achieved an accuracy of over 33.8% using hand-crafted features with an AdaBoost classifier. The authors examined several features, including the size of each burst, total transmitted bytes, the number of bursts, occurring packet sizes, percentage of incoming packets, and the number of packets. Wang et al. [33] developed an automated voice traffic collection tool and gathered a much larger dataset containing more than 150k samples from Amazon Echo and Google Home smart speakers. They can accurately infer voice commands over encrypted traffic with 92.9% accuracy on Amazon Echo by training a variety of deep-learning models on sequences of packet direction and sizes. Similarly, Hyland et al. [17] built their own data collection tool and collected a modest dataset containing traffic samples for 20 commands produced with an Alexa device. These researchers propose a new attack method that outperforms Wang et al., achieving 95.2% against Wang et al.'s Alexa dataset, by using an ensemble model of multiple feature-extractor deep-learning models trained on sequences of packet directions, timestamps, sizes, and inter-arrival times. Mao et al. [21] developed a novel deep-learning model using an attention-based bidirectional gated recurrent unit. Their achieved performance competitive to prior attacks, reaching 93.36% accuracy on the Wang et al. Alexa dataset. Recently, Ahmed et al. [10] are the first to present an end-to-end system that can both detect the invocation of voice assistants and determine the specific voice command issued. Using feature-based machine learning techniques, Ahmed et al. could detect assistant activation with 99% accuracy and classify commands at above 80% accuracy against a newly collected dataset containing traffic samples produced by Google, Alexa, and Siri assistants.

3 THREAT MODEL

We consider a network-level, passive adversary with the capability to monitor traffic between a smart speaker, such as Amazon Alexa or Google Home, and the cloud server responsible for processing users' voice queries and responses. This adversary is illustrated in Figure 1. Once users send voice queries over speakers, the adversary eavesdrops on the voice query response traffic between the smart speaker and the server, referred to as "voice command traces" in this paper. The adversary extracts feature vectors from these traces and feeds them to train the model.

Furthermore, we assume that the adversary eavesdrops on the local network, wherein connections passing through the router have unique IP addresses. This allows the adversary to identify smart speaker traces by filtering out unnecessary traffic based on the victim's IP address. We also note that this is not unrealistic assumption as previous literature [10] demonstrated that voice assistant traffic can be effectively identified even when connections were observed as one global IP address due to Network Address Translation (NAT). According to their work, voice fingerprinting can be executed effectively as a two-staged attack in real-world scenarios. Initially, the adversary identifies voice query traffic by detecting the wake word invocation and this first stage completed successfully, as demonstrated by Ahmed et al. [10]. Subsequently, the adversary can distinguish between voice query traffic, utilizing his model to identify the specific command the user sends. *This paper focuses on improving the second stage of the attack.*

With this specific goal, our adversary aims to perform voice command fingerprinting in two experimental scenarios: the *closed*-world and *open*-world settings. In the closed-world scenario, only traffic for voice commands used in training the model can be observed. Thus, we assume that the victim uses a small, limited set of commands which we refer to as "monitored" commands. The open-world scenario is a more realistic setting, as the goal is extended to detect monitored commands against "unmonitored" commands. In this scenario, we assume that the victim can send any voice commands, resembling real-world users.

We assume that the adversary is resourceful enough to maintain a substantial traffic database and train a deep learning model capable of linking voice traffic to individual voice commands. Regularly collecting traffic data, the adversary retrains the model to ensure robustness against concept "drift," a phenomenon that could

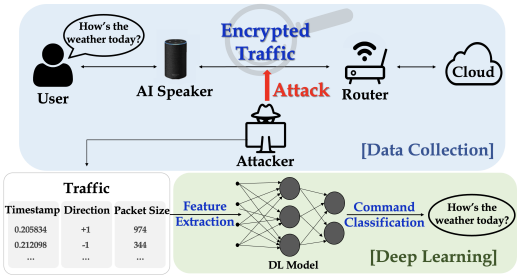


Figure 1: VCF Threat model.

degrade model performance over time due to potential changes in voice query responses [18, 11].

4 DATA

In this section, we discuss the methodology of voice command traffic collection and details of datasets used in the paper.

4.1 Voice Command Audio File Generation

To automate the process of sending voice queries, we recorded voice commands as audio files. Initially, we selected 600 widely used commands by aggregating recommendations from the official Alexa application [2], Google Nest Help [5], Google Search Help [6], Siri-Apple [8] and questions found in Alexa's official blog posts [3] published between November 2021 and December 2022. Additionally, the dataset in Hyland et al. [17] and the dataset in Wang et al. [33] were included.

Following this, we employed the Narakeet Text-To-Speech (TTS) API [9] to generate audio files based on the identified voice command texts. To enhance diversity, we configured voices with different accents and genders, encompassing a total of four demographics: American (Sarah, Matt), British (Beatrice, Spencer), Indian (Lakshimi, Dilip), and Australian (Gail, Tara), with both male and female variations for each accent. In total, eight versions of voice files were generated for each command. This approach allowed us to create a dataset that reflects the varied usage of voice commands in real-world scenarios and we demonstrate the impact of these variations on fingerprintability in Section 7.

4.2 Data Collection

We adopted the voice command traffic collection methodology implemented by Hyland et al. [17] to gather the voice command traffic dataset used in this paper. Our voice command traffic collection tool can be seen in Figure 8 in Appendix A.

The data collection process involved the use of five Raspberry Pi units, each equipped with the Raspberry Pi OS, along with five Alexa devices — one Amazon Echo, three Amazon Echo Dots, and one Amazon Echo Show. To facilitate data collection, we connected the Raspberry Pi units to the Ethernet via a router, configuring them as bridged access points. Additionally, we established a connection between Alexa and this access point to capture network traffic. Subsequently, we initiated the capture on a Raspberry Pi to monitor network traffic between a smart speaker and the wide-area network.

To produce voice output, we connected a speaker to the AUX port on the Raspberry Pi. For recording Alexa's responses, we connected a pin microphone to a USB port on the Raspberry Pi. Notably, instances where Alexa failed to capture commands or encountered unknown errors when connecting to the Amazon cloud were excluded from the recording, allowing for later filtering based on recordings.

The 600 commands were split into 300 monitored commands and 300 unmonitored commands, each configured with instances of 1000 and 300, respectively. As Alexa struggled to recognize the wake word when utilizing TTS-generated voice, human voice recording files were used for the wake word. Since most responses from Alexa devices were completed within 60 seconds, a time limit of 60 seconds was set for both traffic captures and response recordings.

To account for potential variations in network traffic based on geographic locations, we configured Alexa to operate in five different countries: United States, India, United Kingdom, Canada, and Germany — the top five countries with the highest Alexa usage. Within these regions, 20 commands capable of eliciting varied responses based on Alexa's location were identified.

The dataset was collected using the campus network from May to September 2023. To mitigate the impact of temporal changes in network traffic, we employed a round-robin fashion for voice command traffic collection. This involved initiating the collection for the next command only after completing the collection of all instances for a particular command. To simulate more realistic Alexa usage, data collection was conducted in an environment featuring ambient noise, including sounds such as door closures, faint human speech, and even the sound from another Alexa speaker.

We furthermore collected a smaller dataset using Google Assistant on an Android phone in parallel to our Alexa capture. This dataset was collected using the same crawling framework as our Alexa datasets, and used just one Raspberry Pi connected to a *Google Pixel 4a 5G* android phone for voice command traffic generation. Google Assistant on Android requires the use of Voice Match to enable "Hey Google" activation, so we limit our synthetic voice generation to male American accent voices and train the phone's voice match model on this voice profile. Additionally, the phone's search and app store region was set to the United States region. Due to unreliable sensing of the phone's activation response, we set a minimum traffic capture time of 20 seconds. We used this configuration to collect traffic samples for a subset of our Alexa command list. This dataset was collected between April and August 2023. After filtering erroneous samples and balancing the collected dataset, this capture produced 150 samples per each of our 427 commands collected.

4.3 Dataset

To this end, we used four datasets in this paper: WhiVo Alexa and Google datasets collected in this paper, and Shame Alexa Dataset, DeepVC Alexa Dataset provided by existing VCF literature [17], [33].

WhiVo Alexa Dataset. Using the list of voice commands compiled in Section 4.1, we constructed the Alexa dataset. After filtering out traces with a packet count fewer than 50, the dataset comprises 300 monitored commands, each consisting of 1,000 traces, and 300 unmonitored commands, each comprising 300 trace instances.

WhiVo Google Dataset. We similarly construct our Google dataset as our Alexa dataset. This dataset contains 427 commands with 150 instances each.

Shame Alexa Dataset. This dataset, sourced from prior research [17], was gathered using the Amazon Echo Dot (second generation). It consists of 20 voice commands, comprising 10 static commands and 10 dynamic commands. We discuss the definitions of static and dynamic commands and thoroughly explore the impact of these command characteristics on voice command traffic in Section 7.2.2. Each command includes 800 trace instances.

DeepVC Alexa Dataset. Wang et al. [33] collected a dataset using the Amazon Echo (second generation). Voice command traces were captured by submitting commands recorded by five users (three female and two male users). This dataset comprises 100 monitored commands, each containing 1,500 trace instances, and 100 unmonitored commands with 200 instances each.

5 FEATURE EXTRACTION

In this section, we investigated a range of feature sets and identified the most effective features for voice command fingerprinting. Specifically, we discuss traditional features that have been utilized in prior WF literature [35, 29, 15, 32, 28, 25, 12, 24, 26, 27], and introduce new features that extend more recent traffic feature representations, known as TAM [30].

5.1 Traditional Features

We extracted packet direction, timestamp, and size information from the Ethernet, TCP, and IP packet headers. We then created diverse feature vectors, which are detailed as follows.

Packet Directional Features. The packet directional feature vectors have been widely adopted by researchers [35, 32, 15, 25] due to their proven informativeness. We assigned values of "-1" or "+1" to indicate the direction of communication between the client and server. In this paper, for packets sent from the server to the smart speaker (denoted as "incoming" directional packets), they were encoded as "-1" in feature vectors. Similarly, for packets sent from the smart speaker to the server (denoted as "outgoing" directional packets), they were encoded as "+1" in feature vectors.

Packet Timing Features. In our exploration, we considered two types of packet timestamp sequences: one comprised of Packet Arrival Timestamps (PAT) and the other of Inter-Packet Delays (IPD). Both types of timing features have been extensively employed in WF literature [35, 28, 15, 24, 12] and are recognized as some of the most effective features for traffic analysis. PAT indicates the time at which each packet arrived, while IPD represents the time interval between consecutive packets, measured in milliseconds. According to Table 1, the IPD sequence performed more effectively than the PAT vectors to identify voice commands.

Packet Size Features. The packet size feature vector comprises the size of TCP packets and proves to be a robust traffic pattern for fingerprinting voice commands [17, 33] and websites [26, 27, 35]. This is attributed to the variation in traffic volume, reflecting the diverse sizes of websites and voice command responses. Table 1 supports this observation, demonstrating the effectiveness of the packet size feature in distinguishing between voice commands, surpassing other traditional features. Notably, the packet size representation achieves the highest accuracy, leading us to include it in the final features used in Section 7.

5.2 Variants of TAM

Shen et al. [30] introduced a novel traffic representation, TAM, which segments fixed-length time slots and calculates the count of outgoing and incoming packets per time slot, consolidating this information into a feature matrix. This representation has demonstrated robustness against WF defenses [13, 16, 19, 23], as it incorporates packet direction, timing, and size features, along with total packet statistics per time slot. Consequently, it appears to be more resilient in identifying website traffic patterns, even in the presence of packet delays and paddings. In this section, we provide a brief overview of TAM and elaborate on how we adapt the TAM representation for voice command fingerprinting.

Original TAM. We denote T as a voice command trace and $T = (p_1, p_2, ..., p_l)$ in which p_k is the k-th time slot in the trace and l is the total number of time slots in the trace. We also denote m_{+k} and m_{-k} as the outgoing packet count and incoming packet count in the k-th time slot, respectively. Let s be the time slot and D be the maximum duration of the trace. Then, the number of time slots, $l = D/s$. Note that we tuned l and D in Section 6.2. In the original TAM representation, the feature matrix is created by *stacking* two count vectors of v_{+c} and v_{-c} in which $v_{+c} = (m_{+1}, m_{+2}, ..., m_{+l})$ and $v_{-c} = (m_{-1}, m_{-2}, ..., m_{-l})$, and expressed as $F_{TAM} = [v_{+c}; v_{-c}]$. Sum of absolute values of total packet size

Volume Aggregation Vectors (VAV). We introduced a variant of TAM, termed Volume Aggregation Vectors (VAV), to investigate traffic volume per time slot, providing insights into the length of voice command responses. For each time slot indexed by k, s, s_{+k}, and s_{-k} denote the sum of absolute values in sizes of total packets, outgoing packets, and incoming packets, respectively. Subsequently, we constructed two types of VAVs using s_k or s_{+k} and s_{-k}. The first set of VAVs, denoted by $F_{VAV_1} = v_s$, is created with $v_s = (+s_1, +s_2, ..., -s_l)$, assuming that the first two packets are outgoing while the last packet is incoming. It is important to note that we annotate all entries of the vectors with "+" and "-" to directly encode the direction of packets. The second VAVs are constructed by *concatenating* two size vectors, v_{+s} and v_{-s}, where $v_{+s} = (+s_{+1}, +s_{+2}, ..., +s_{+l})$ and $v_{-s} = (-s_{-1}, -s_{-2}, ..., -s_{-l})$, expressed as $F_{VAV_2} = [v_{+s}||v_{-s}]$.

Timestamp Aggregation Vectors (TAV). Within this feature category, we explored two types of timestamps—Packet Arrival Timestamps (PATs) and Inter-Packet Delays (IPDs). For each time slot indexed by k, t, t_{+k}, and t_{-k} represent the sum of PATs of total packets, outgoing packets, and incoming packets, respectively. We created two types of TAVs using t_k or t_{+k} and t_{-k}. The first set of TAVs, denoted by $F_{TAV_1} = t_s$, is created with $t_s = (+t_1, +t_2, ..., -t_l)$, assuming that the first two packets are outgoing while the last packet is incoming. The second set of TAVs is constructed by concatenating two PAT vectors, t_{+s} and t_{-s}, where $t_{+s} = (+t_{+1}, +t_{+2}, ..., +t_{+l})$ and $t_{-s} = (-t_{-1}, -t_{-2}, ..., -t_{-l})$, expressed as $F_{TAV_2} = [t_{+s}||t_{-s}]$. We further investigated IPD sequences for TAVs by aggregating IATs in the time slot with the average IAT, this representation yielded lower accuracy compared to TAVs based on PAT sequences (Table 1).

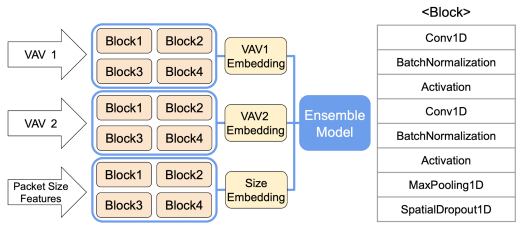


Figure 2: The architecture of the WhiVo model.

Consequently, we exclude the feature representation details using IPD vectors in this section.

Utilizing the VAV and TAV representations, we encode not only the total packet count but also the traffic volume and duration information per time slot. This expansion beyond the total packet count is essential for capturing such crucial statistics. In particular, aggregating packet size information per time slot provides a more robust traffic pattern against highly noisy and variable voice command traffic compared to fine-grained, per-packet features such as sequences of PATs, packet sizes, and packet directions. Notably, we choose to concatenate incoming and outgoing vectors instead of stacking them, a decision made to optimize input compatibility for 1D CNN — a model that has demonstrated effectiveness in traffic fingerprinting tasks [32] and has been endorsed by recent WF research [34, 35, 26, 27, 25, 29, 32, 12].

6 VOICE FINGERPRINTING ATTACK

In this section, we provide a detailed explanation of the WhiVo model, specifically focusing on an ensemble model built upon the DF architecture [32]. We also discuss the process of parameter selection for both the model and the VAV and TAV representations, as outlined in Section 5.2, various feature analysis results, and finally introduce a new variant of voice fingerprinting attack, namely voice command category fingerprinting.

6.1 Model Architecture and Adjustments

As demonstrated by Hyland et al. [17], the SHAME ensemble model proved effective in identifying the voice commands sent by the user. We adopted their model as the baseline. The SHAME model comprises three DF models trained separately to act as feature extractors for different traffic feature representations. Each model comprises four CNN blocks, each containing two 1D convolutional layers followed by a batch normalization layer. Additionally, each block includes one 1D maxpooling layer and one 1D spatial dropout layer, as illustrated in Figure 2. After generating three types of feature embeddings, namely VAV1 (F_{VAV_1}), VAV2 (F_{VAV_2}), and packet size (Size) embeddings (F_s), we constructed the input feature vector for the ensemble model by concatenating them, i.e., $F_{ens} = [F_{VAV_1}||F_{VAV_2}||F_s]$.

It's important to note that, although the ensemble model used by Hyland et al. involved further concatenation with a hand-crafted feature vector containing statistics such as mean, median, and sum, we excluded this feature array as it resulted in marginal performance differences.

Features		Accuracy
PAT		60.4%
Size		79.2%
IPD		69.7%
Direction		56.4%
TAM		88.0%
VAV	F_{VAV_1}	89.6%
	F_{VAV_2}	89.7%
TAV(PAT)	F_{TAV_1}	84.1%
	F_{TAV_2}	85.5%
TAV(IPD)	F_{TAV_1}	75.1%
	F_{TAV_2}	76.3%

Table 1: Feature comparisons using traditional WF features, TAM, and TAM variants.

	Acc	Ens 1	Ens 2	Ens 3	Ens 4	Ens 5
Size	79.2%				O	O
TAM	88.0%	O	O			
VAV1	89.6%			O		O
VAV2	89.7%	O	O	O	O	O
TAV2	85.5%	O				
WhiVo		91.1%	91.0%	90.7%	91.3%	92.1%

Table 2: Accuracy (Acc) of feature combinations for the ensemble input vector (Ens) (Note that O indicates that the corresponding feature is included.).

6.2 Hyperparameter Tuning

Model Parameters. We fine-tuned the DF model to better align with our dataset. The search space for each parameter in the model and the selected parameters are reported in Table 6 in Appendix B.
VAV Parameters. As discussed in Section 5.2, we chose VAVs, F_{VAV_1} and F_{VAV_2}, by extending a TAM proposed by Shen et al. [30], along with F_s. For a more effective VAV representation, choices on D and l are important. If the voice command trace is divided into many time slots, the length of VAV becomes too long and more sparse with more zero values. On the other hand, if it is divided into a small number of parts, it may lose some traffic pattern as each time slot aggregates a large volume of traffic from many packets. Therefore, we have to identify an effective number of time slots, l, balancing between sparsity and information loss, and in turn, resulting in good performance of the VCF model. According to Table 6, we selected 7,500 for l as it corresponded to the highest accuracy. Considering that we observed the average and longest duration of voice command traces to be 20 and 60 seconds, respectively, we explored this range as a search space and empirically chose D to be 60 seconds. Consequently, the duration for each time slot is 0.008 seconds. Additionally, to ensure uniformity in length, we padded VAVs shorter than 7,500. Lastly, we segmented the traces by allowing some overlap between slots with slot strides (δ) of 0, 0.002, 0.004, and 0.006 seconds. Interestingly, this variant did not make the WhiVo model more effective.

6.3 Feature Evaluation

Single Model Features. By adopting the DF model, we evaluated various features discussed in Section 5.1 and 5.2 to select the best features for the VCF model. We observed that most IPDs involved small floating-point values; thus, we applied the logarithm to enlarge the gap between these numbers. According to Table 1, for traditional features, PAT and IPD feature vectors performed less effectively; thus, we first excluded these from the candidate pool. For TAM extensions, VAVs clearly outperformed TAM, while both TAVs with IPDs and PATs failed to improve the performance of the VCF with TAM. Thus, we dropped TAVs from the candidate pool. As a result, we included Size, TAM, VAV1 (F_{VAV_1}), VAV2 (F_{VAV_2}), and TAV2(PAT) (F_{TAV_2}) in the final candidate pool and further created diverse combinations of them to propose more effective feature vector for the ensemble model.

Ensemble Model Features. We explored a variety of feature combinations, as shown in Table 2. When concatenating features in TAM extensions, we observed an increase in the ensemble model's performance by approximately 1% in accuracy compared to the best-performing single model. For instance, the feature combinations based on TAM, VAV1, VAV2, and TAV2 yielded marginal performance improvement with about a 1% increase in accuracy. However, when combining packet size features with TAM extension features (e.g., $[F_{VAV_1}||F_{VAV_2}||F_s]$), we noticed the ensemble model performance increased slightly by 1%. According to Table 2, the most effective combination of input vectors for the ensemble model is $[F_{VAV_1}||F_{VAV_2}||F_s]$. It seems that introducing more variability across the ensemble inputs to be concatenated could be helpful to enhance the ensemble model's performance, as $[F_{VAV_1}||F_{VAV_2}||F_s]$ outperformed $[F_{VAV_2}||F_{TAV_2}||F_{TAM}]$, although the performance gap was not considerable. We leave further investigation into improving learning capability of the ensemble model by more diversity in the combination of input feature vectors for future work.

6.4 Voice Command Category Fingerprinting

Given that the corpus of voice commands is infinite, collecting traces for all user commands to train the WhiVo model is impractical. To address this scalability challenge, we propose voice command category fingerprinting, detecting clusters of voice commands, and consequently, inferring their content rather than unveiling specific queries.

In this attack approach, our objective is to determine the voice category to which the target command traffic belongs, rather than linking each trace to a specific command. To achieve this, we first identified clusters by grouping similar commands and trained the model using voice command traces labeled with these clusters. For cluster discovery, we adopted two methodologies: manual cluster discovery and the Term Frequency-Inverse Document Frequency (TF-IDF) based clustering, described in detail as follows.

- **Manual Cluster Discovery.** Among the 600 commands in the Amazon Alexa dataset, we observed that 178 commands were labeled with 24 categories, as provided by the Amazon Alexa App [2] and Alexa Blog [3]. The remaining commands were manually classified into these categories, and all the categories are listed in Table 8 in Appendix E. Each of the 24 categories contains 5-10 voice commands, resulting in an imbalanced dataset. We discuss how we handle this imbalance when training the model in Section 7.
- **TF-IDF-based Clustering.** The TF-IDF Vectorizer is a tool that converts text documents into numerical vectors for machine learning. It represents each document as a vector, with each dimension corresponding to a unique term and values based on TF-IDF scores, capturing the importance of terms in specific documents relative to the entire collection. This enables algorithms to analyze and understand text quantitatively. We adopted this approach to generate voice command features, preserving the similarity between commands for traffic analysis in WhiVo models. To measure distances between commands accurately, we calculated pairwise Euclidean distances and assigned commands to the same cluster if the distance was below a specified threshold. This clustering methodology is outlined in Algorithm 1 of Appendix E. We sorted the clusters in descending order based on size and selected the top eight clusters, which are listed in Table 9 in Appendix E.

We believe this research direction is feasible because similar voice commands within the same category are likely to generate comparable responses in terms of both format and content. This consistency in responses would further contribute to the emergence of common traffic patterns in the responses. Further evaluation on both methodologies will be demonstrated in Section 7.3.

7 EXPERIMENTAL RESULTS

7.1 Overall Performance

In this section, we demonstrate the WhiVo model performance in the closed-world and open-world settings, and further compared them to state-of-the-art VCF models.

Experimental Settings. We implemented the WhiVo model following Hyland et al.'s work [17]. The experiments utilized an NVIDIA RTX A6000 GPU with 512GB RAM. We employed five-fold cross-validation to compute metrics, reporting mean accuracy and standard deviation. The dataset was split into training and testing sets in a 9:1 ratio, with 10% of the training set used for validation. The patience (maximum epochs with no improvement seen on the validation set) was set to 100, and the number of epochs was empirically chosen as 600.

Metrics. We used the following evaluation metrics to measure the performance of WhiVo model:

- Accuracy: The proportion of correctly classified voice command traces, applicable in closed-world scenarios.
- True Positive Rate (TPR): The proportion of correctly classified monitored voice command traces among monitored samples, relevant in open-world scenarios.
- False Positive Rate (FPR): The proportion of misclassified unmonitored voice command traces among unmonitored samples, used in open-world scenarios.
- Receiver Operating Characteristic (ROC): A graphical representation plotting the true positive rate against the false positive rate. This visualization aids in assessing the model's

Dataset	WhiVo	SHAME	DeepVC
WhiVo Alexa	**92.1%**	85.6%	82.4%
SHAME Alexa	**89.9%**	85.0%	65.9%
DeepVC Alexa	**91.4%**	82.6%	78.8%
WhiVo Google	**92.8%**	90.5%	87.7%

Table 3: Comparison to state-of-the-art VCF models across different datasets in the closed-world setting. Note, all standard deviations are less than 1%.

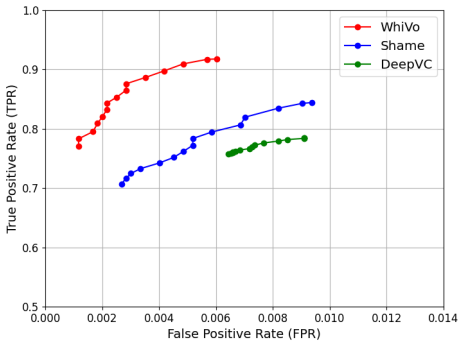


Figure 3: Comparison to state-of-the-art in the open-world setting using ROC curve.

ability to distinguish between the two classes across different thresholds.

Closed-World Evaluation. In this scenario, our objective is to accurately classify the monitored commands, assigning each one its unique label. We explored four datasets: WhiVo Alexa, SHAME Alexa [17], and DeepVC Alexa [33], which were collected using Alexa devices, and WhiVo Google, which was captured using a Google phone. We used 300 commands and 300 instances each for WhiVo Alexa, 20 commands and 300 instances each for SHAME Alexa, 100 commands and 300 instances each for DeepVC Alexa, and 300 commands and 150 instances each for WhiVo Google.

We assessed the performance of the WhiVo model across various datasets to compare it with state-of-the-art VCF models like DeepVC [33] and SHAME [17]. Note that we used the source codes [4, 7] provided by the authors for DeepVC and SHAME model evaluations. According to Table 3, the WhiVo model clearly outperformed all state-of-the-art models across four datasets by an improvement of about 7%, 5%, 9% and 2%, respectively. This result shows that our new features aggregated based on VAVs and packet size sequences improves the SHAME model performance as WhiVo adopted the architecture of SHAME.

Compared to state-of-the-art models, the performance gain for WhiVo model was marginal when differentiating voice command traffic collected using Google. All models performed more effectively against WhiVo Google, indicating that Google Assistant exposed more voice traffic patterns than Alexa.

Open-World Evaluation. In this context, we optimized the WhiVo model to maximize the TPR, representing the ratio of correctly classified monitored commands, while simultaneously minimizing the FPR, indicating the ratio of incorrectly classified unmonitored commands. The goal was to correctly identify each of the 200 monitored commands with its specific label against 300 unmonitored commands. We assigned 300 distinct labels for monitored commands and one label for unmonitored commands. In comparison with SHAME and DeepVC, both evaluated on the Amazon Alexa dataset, WhiVo outperformed these models, as depicted in Figure 3. This demonstrates that the WhiVo model achieves robustness even in more realistic scenarios.

Models		Accuracy
Global		83.2%
Local	US	94.7%
	UK	94.0%
	Germany	94.0%
	India	91.8%
	Canada	92.6%

Table 4: Results of location experiments(Note that we computed the average of accuracies after five experiments.).

7.2 Fingerprintability Analysis

To achieve optimal performance in fingerprinting attacks applicable to real-world scenarios, it is crucial to explore factors influencing the fingerprintability of commands. In this section, we consider various factors that can reflect potential variations in voice command traces, incurred by real-world users, including Alexa locations, static and dynamic commands, and human factors such as users' demographics and genders.

7.2.1 Location. For users seeking a more personalized experience with the Alexa device closely tied to their daily lives, location setting becomes imperative. This setting enables the device to provide responses tailored to the user's specific location. Amazon Alexa incorporates a function enabling users to specify their locations. In this section, we conducted experiments to examine the impact of location settings on voice command traces by configuring various locations, including the United States (US), the United Kingdom (UK), Germany, India, and Canada, on five Alexa devices. We manually selected 20 commands whose responses are likely to vary based on locations and queried them using Alexa devices. Examples of such commands include "How's the traffic?", "What restaurants are nearby?", and "What's the high today?". (Table 7 in Appendix C) We used 300 traces for each of the 20 commands. We investigated various experimental scenarios as follows.

Global vs. Local Models. In this scenario, we considered two types of attackers. The first attacker is interested in monitoring voice command traces across countries. In this case, we created one global WhiVo model capable of fingerprinting voice command traces collected from diverse users in various countries. The second attacker aims to build a single model for each country. We created five local WhiVo models, one for each country, enabling the fingerprinting of voice command traces gathered from a relatively smaller number of users within a specific country. While the first attack model may lead to performance degradation, it reduces time and space complexities for training and testing the WhiVo

model. The latter attack model could improve performance by using country-specific training sets and models, albeit at the cost of running multiple models.

For the first attack model, we constructed five datasets, each consisting of voice command traces collected from one device with its location setting configured to one country. For the second attack model, we built one dataset by mixing the five datasets described above. As shown in Table 4, we obtained an accuracy of 94.70% when considering the local model configured to use the US location. On the other hand, we achieved an accuracy of 83.2% when experimenting with the global model trained on traces from all five countries.

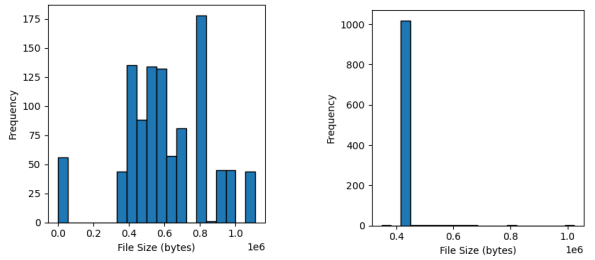
We further reported the performance of each of the four local models in Table 4. There was almost no difference between the performances of the local models, indicating that there were no locations more advantageous or disadvantageous for the WhiVo model. However, the model performed less effectively with increased intra-class variances when fingerprinting voice commands from users across multiple countries, and higher accuracy can be achieved by training multiple local models, albeit at the cost of increased time and space complexities, regardless of the specific location.

Fingerprinting Commands from Unknown Locations. As shown in Table 4, voice command traces vary depending on the location. We hypothesized that if an attacker tries to uncover commands from a user in a different location inaccessible to the attacker, the success rate of the attack would decrease. To test this assumption, we trained the model using a dataset collected with the device location set to the US, consisting of 270 instances per command. For testing, we used a dataset with 30 instances per command, collected from four devices configured to four countries other than the US.

The WhiVo model achieved an accuracy of 42.4%, indicating a significant degradation in model performance when fingerprinting command traces collected from unknown countries. Thus, we further examined misclassified voice commands and corresponding responses, revealing high variances in their voice command traffic and identifying semantic gaps between responses, even for the same voice queries.

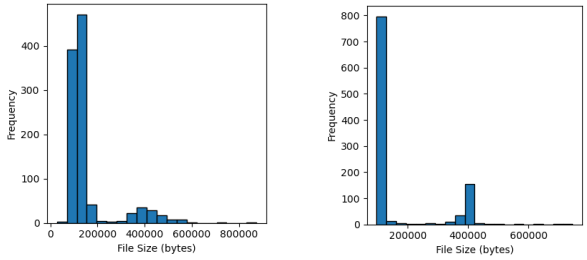
For instance, we observed that some of these confusions occur when the WhiVo model classifies commands involving different units in their responses. The US uses Fahrenheit as the unit for temperature-related measurements, while the other four countries use Celsius. For the command "What's the high today?", the response was "Today's expected high XX degrees Fahrenheit" in the US, whereas it was "Today's expected high XX Celsius" in other countries. Additionally, the USA uses miles as the unit of distance, while the other four countries use kilometers. For the command "How windy is it?", the response was "Currently, the wind is calm at XX miles per hour today" in the US. In contrast, it was "Currently, it's breezy at XX kilometers per hour" in the other four countries.

We observed even more significant differences in responses when they are highly relevant to the location. For the command "Is the post office open?", the response was "Hours of operation are not available. For post office XXX, you can ask me for their phone number" in India, while it was "There is a post office called XXX. It is currently closed but will be open today from XXX" in other countries. Additionally, for the command "How long will it take



(a) Dynamic (b) Static

Figure 4: Audio file size distributions for dynamic and static commands.



(a) Dynamic (b) Static

Figure 5: Pcap file size distributions for dynamic and static commands.

to get to Costco?", the response was "Based on current traffic, it will take about XXX to drive" in the US, UK, and Canada, while it was "Sorry, I am not possible to search for that" in India and Germany. This is because Costco does not operate stores in India and Germany. This variation makes it more difficult to definitely classify these commands.

Location Identification. In the final evaluation, we shifted the focus of fingerprinting to pinpoint the location by analyzing voice command traces labeled with the specific location, one of the five countries to which the collection device was configured. By using 4,800 training instances for each location label, the WhiVo model demonstrated effective performance with an accuracy of 86.5%. This outcome suggests that voice command traffic analysis can be extended to infer the locations of smart speaker users. This poses a significant privacy threat, as the location of users can be exposed to network-level adversaries.

Based on our study, the WhiVo model yielded high accuracies ranging from 92% to 95%, irrespective of countries, when employing local models. The global model also exhibits notable performance with an accuracy of 83%. Although the fingerprinting ability degraded in a much broader attacker scenario when detecting voice commands from unknown countries, there is potential for improvement, possibly by employing larger training sets or advanced features. Thus, our findings highlight a substantial amount of information leakage through encrypted network traffic from Alexa devices, across all five countries.

7.2.2 Dynamic vs. Static Commands. As explored in previous literature [17, 33], we also additionally organize voice commands into categories of "dynamic" and "static". The dynamic label indicates

that the voice command is expected to have variability in the responses and traffic produced between different calls of the same command. Conversely, the static label conveys that the voice command traffic is expected to be similar to each other. For example, "tell me a pun" is a dynamic command while "what's 99 divided by 12?" is a static command.

For labeling, previous VCF work assigned labels based on *manual, qualitative assessments* of the anticipated function and behavior of commands. In this work, we instead perform this labeling through *empirical analysis* of the data to directly reflect the degree of variations in voice command traces. To accomplish this, we observed the lengths of audio files recording responses. As longer responses would carry a greater number of packets, we regarded this as a reasonable criterion to assign dynamic or static labels to voice commands.

We visualized the distributions of audio file sizes for one dynamic command and one static command in Figure 4. The file size distribution is more skewed in the static commands than the dynamic commands since the static command responses are more similar to each other, while the dynamic commands contain voice command traces more variable in responses. With this observation, we assigned the static label to the command whose largest bin contains more than 90% of trace instances while determining the command as the static label when its largest bin includes less than 50% of instances. As a result, we constructed 24 dynamic commands and 24 static commands, each comprising 200 instances.

When we trained and tested the WhiVo model using dynamic command trace instances, the model performed less effectively than when training and testing using static command traces (80.5% vs. 92.8%). Presumably, because responses in dynamic cases consistently change, they might pose more difficulty in being grouped as a single fingerprint compared to the static ones, which always provide the same response.

To investigate the relationship between the responses of voice queries and voice command traces captured during the response, we further plotted the distributions of packet capture file (i.e., pcap) sizes in Figure 5. In comparison to audio file size distributions, the pcap file size distributions for both static and dynamic have characteristics of bimodal distributions. Despite the increased similarity between static and dynamic commands under this lens, we still see clear differences between the spread within each peak, showing further evidence for the material impact that dynamic command responses have on voice command network traffic.

Recognizing the correlation between response consistency and traffic pattern uniformity, implementing strategies to introduce variability in the responses generated by voice assistant systems could prove to be an effective defense against VCF attacks.

7.2.3 Human Factors. The objective of this section is to investigate the impact of diversity in voices on voice command traces. Wang et al[33] pointed out that this factor affects the voice traffic pattern by showing that incoming traffic could change by humans' different accents. In this paper, we revisited this influence by considering more various human factors including demographics and genders of smart speaker users.

The voice commands are recorded by both male and female speakers with diverse accents from the US, Australia, the UK, and India and sent over Alexa devices to collect the voice command traces.

Impact of User Gender. To investigate if gender influences fingerprinting attacks, we conducted experiments using datasets comprising both female voices (Sarah, Gail, Beatrice, Lakshmi) and male voices (Matt, Tara, Spencer, Dilip) generated using TTS [9]. We used 200 trace instances for each of the 295 commands, and each instance consists of 50 traces captured using voice commands recorded by each of the four users. The objective was to determine if the success rate of the WhiVo model differs based on the gender of users. Such knowledge could be crucial for attackers in selecting target users strategically.

As illustrated in Figure 10 in Appendix F , the WhiVo model achieved an accuracy of 90.6% with male voices and 89.4% with female voices. This suggests that the impact of user gender on the WhiVo model's performance is marginal.

Impact of User Demographics. Similar to user gender, user accents can vary based on demographics, thereby influencing voice command traffic. Considering four demographics—American (Sarah, Matt), British (Beatrice, Spencer), Indian (Lakshimi, Dilip), and Australian (Gail, Tara), We used 200 trace instances for each of the 295 commands, and each instance consists of 100 traces captured using voice commands recorded by each of the two users from each demographic. —the WhiVo model achieved an accuracy of 91.8% for the American accent, 93.6% for the Australian accent, 93.0% for the Indian accent, and 93.2% for the British accent. Figure 10 in Appendix F illustrates that there was no significant difference in the performance of the WhiVo model based on the user's accent.

While a targeted attack concentrating on users with specific gender and demographic attributes may not yield increased effectiveness, the robustness of VCF attacks is evident across various users. This suggests that network traffic analysis can be effectively conducted for a diverse range of users.

7.3 Voice Command Category Fingerprinting

As outlined in Section 6.4, we established category labels through two approaches: manual cluster discovery and TF-IDF-based clustering. In the context of voice category fingerprinting, we opted for the DF model [32] over the WhiVo model, given its comparable performance with significantly lower time and space complexities. The subsequent discussion delves into the DF model's performance relative to the employed methodologies.

Manual Cluster Discovery. When manually assigning category labels to voice command traces, multiple commands are grouped into each voice category, resulting in increased intra-class variances. However, the model demonstrated robust performance with an accuracy of 84.0% in correctly identifying its category. Confusion matrices presented in Figure 9 in Appendix D reveal that 18 out of 24 categories exhibited particularly strong fingerprintability, each achieving an accuracy exceeding 80%.

Upon scrutinizing several categories, category 1 encompasses voice commands related to stock prices, such as "What's the Google stock price?". Category 4 comprises weather-related commands, like "What is the weather?". Category 13 involves language translation-related queries, such as "How do you say 'table' in Portuguese?". Similarly, category 14 includes commands related to searching for

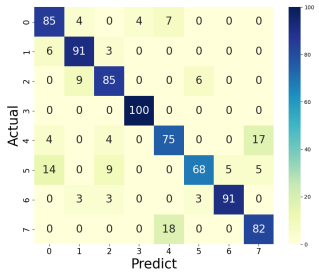


Figure 6: Confusion matrix for TF-IDF-Based Clustering approach (Note that all numbers are in %).

locations like restaurants and grocery shops, while category 23 consists of commands asking for details about past presidents.

These categories appear to exhibit more consistency in responses regarding the length and format, resulting in distinct patterns in voice command traces and, consequently, better fingerprintability with fewer confusions to other categories.

In contrast, category 19 and 21 were less fingerprintable, mainly due to confusions between category 19, 20, and 21. Category 19 comprises commands asking about world records, such as "What's the world record for push-ups?". Category 20 involves queries about animals, like "How big is the smallest cat?". Category 21 consists of commands asking about music award records, such as "Why are the Grammys called the Grammys?". We observed that voice commands in category 19 and 21 exhibited more variations in their responses, leading to increased intra-class variances in their voice command traces and, consequently, more confusions with other categories.

TF-IDF-based Clustering. Manually clustering a large number of commands from our experiment would be impractical, and human-defined criteria for grouping categories may lack clarity. To address this, we assessed attack accuracy using the TF-IDF-based clustering algorithm outlined in Algorithm 1.

In the experiment, the model achieved an accuracy of 86.2%. The confusion matrix in Figure 6 shows that each category's accuracy exceeded 68% without significant confusions.

However, we observed that certain commands, such as "What time is it?", "Good morning.", and "What is the data tomorrow?", were grouped into the same category, despite their expected dissimilar responses. This highlights a limitation in the automation of cluster generation, as TF-IDF vectors tend to be similar when combinations of words in voice commands are similar, which may not consistently reflect the actual similarity in voice command traces. Nevertheless, this approach demonstrates the DF model's robust performance with an 86% accuracy, higher than the classification performance with manual clustering (e.g., 86% vs. 83%). We also acknowledge a slight degradation in the attack performance compared to voice command fingerprinting, as indicated in Table 3. However, we could have enhanced scalability of the attack by efficiently grouping new commands into existing categories, which will be elaborated on in the next section.

Identifying Unknown Voice Command Category. In this section, we investigated the capability of voice category fingerprinting against unknown voice commands not exposed during training. By

	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Avg
Cate 0	76.6	90.0	40.0	16.7	86.7	62.0
Cate 1	0.0	0.0	20.0	80.0	86.7	37.3
Cate 2	80.0	83.3	93.3	66.7	73.3	79.3
Cate 3	26.7	73.3	96.7	63.3	76.7	67.3
Cate 4	0.0	56.7	0.0	3.3	50.0	22.0
Cate 5	0.0	0.0	13.3	0.0	0.0	2.7
Cate 6	86.7	93.3	100.0	36.7	93.3	82.0
Cate 7	100.0	63.3	80.0	63.3	100.0	81.3

Table 5: The accuracies of 8 categories (Cate) using five experiments (Exp) (Note that all numbers are in %.).

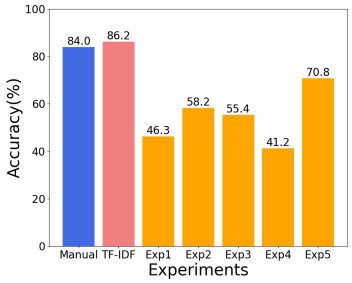


Figure 7: The performance of voice category fingerprinting model when identifying unknown command categories with five experiments (Exp1-5) and when identifying known command categories using manual category discovery (Manual) and TF-IDF-based clustering (TF-IDF).

gathering one command randomly chosen from each of eight categories generated by the TF-IDF-based clustering, we constructed the testing set using voice command traces of the chosen commands. And, we built the training set using the remaining voice command traffic. Since we randomly selected a testing command per category, the model performance could vary depending on the chosen testing commands. By taking this into account, we conducted five experiments and reported the resulting accuracies in Figure 7. The model performance degraded somewhat compared to as increased variations within each category.

Across eight categories, the five experiments showed accuracies ranging from about 41% to 71%. We also note that the fingerprintability of each category against unknown voice command traces is different across categories and experiments. As a different testing command was randomly chosen for each experiment, the model performance could become fluctuating even for the same category detection. Thus, we reported the accuracy of each experimental result along with the averaged accuracy for each of the eight categories as shown in Table 5.

Table 5 demonstrates that categories 0, 2, 3, 6, and 7 showed more successful experimental results, with the model achieving more than 70% accuracy in three out of five experiments. Upon close investigation of their voice commands and responses, we found commonalities in the short length or the response sentence structure in these categories. For example, in category 2 commands such as "How many euros are in a dollar?" and "How many ounces are in a cup?", their responses are "XX dollars." or "XX cups.". Thus,

the model effectively detected the category even against voice commands not presented during training.

On the other hand, as per Table 5, categories 1, 4, and 5 achieved very poor performance, with few of the experiments gaining over 70%. We found that most voice commands in these categories carry very different responses within the category, leading to high intra-class variances. For example, in category 5 commands such as "What is the capital of Spain?" and "What's the animal of the day?", the former command response is "The capital of Spain is Madrid." which was a short sentence. However, for the latter command, the response was "The animal of the day is Southern right whale!" and it continued with details about the whale for 20 seconds. As such, the responses were considerably different, leading to voice command traces within the same category being highly variable.

The lower fingerprintability of certain categories could primarily be attributed to the limitations of the TF-IDF-based clustering algorithm. This algorithm considers only semantically conveyed information about the voice commands, not their responses, to generate clusters. Nevertheless, we observed an intriguing correlation between semantic factors, such as words, length, and sentence structure, which resulted in both similarity in responses and consistency in voice command traffic patterns.

Furthermore, since the current state of the attack demonstrated effective detection in several categories with over 80% accuracies, as illustrated in Table 5, this approach demonstrates significant potential for improved scalability and could become more potent with additional enhancements in cluster generation. Future work will entail a more in-depth exploration of enhancing voice command category fingerprinting through the incorporation of advanced clustering algorithms.

In summary, we emphasize our key findings as follows.

- The VCF showed effective performance against both Google and Alexa voice assistants.
- Substantial information leakage was identified through encrypted network traffic from Alexa devices across five countries, US, UK, India, Canada, and Germany.
- VCF attacks exhibit robustness across various users, indicating effective network traffic analysis for a diverse users.
- A noteworthy connection was identified between semantic elements like words, length, and sentence structure, resulting in both response similarity and consistency in the patterns of voice command traffic.
- Command category fingerprinting demonstrates promising potential for scalability in the real-world through automatic clustering of semantically related voice commands.

8 DISCUSSION

In this section, we address potential limitations of our study and, furthermore, propose avenues for future research to enhance the realism of experimental settings and explore potential defenses specifically tailored to VCF attacks.

More Private Commands. We relied on the recommended commands provided by the voice assistant service. It's important to note that Amazon Alexa integrates additional sources, allowing users to connect to various apps such as games, phone locators, quizzes, and more. Notably, these apps are closely aligned with users' daily lives, providing functionalities related to 'Smart Home,' 'Connected Car,' 'Shopping,' and beyond. Inferring commands related to these activities could potentially yield more impactful insights into users' private information, thus introducing a more threatening scenario.

Fingerprintability Based on Devices. We employed a total of five Amazon Alexa devices with different versions for data collection, including the Amazon Echo (Second, Third, and Fifth generation), and Amazon Echo Show. Investigating the fingerprintability depending on the version of Alexa devices presents an interesting avenue for exploration. It could provide insights into whether there are discernible variations in traffic attributable to the device version.

Advanced Category Fingerprinting. As discussed in Section 7.3, the TF-IDF-based clustering algorithm demonstrated limitations as the TF-IDF vectors could not adequately represent the similarity in voice command traces. This is primarily because TF-IDF relies solely on the frequency of common words appearing in voice commands to gauge similarity. Future work could explore the integration of more advanced, NLP-based clustering algorithms or augmentations to appropriately encode the similarity in voice command traces. Enhancing the learning capability of the category fingerprinting model through these advancements could lead to more accurate and robust results.

Defenses. Past studies in WF have explored various defense strategies, including regular or random padding of traces [19, 14], mimicking traces from representative sites [36], and splitting sending paths [13]. As a next step, it would be intriguing to revisit the WhiVo model against these existing WF defenses. This exploration could pave the way for designing defense mechanisms specifically tailored to VCF, taking into account the unique characteristics and challenges posed by voice command traffic. Moreover, recent practicality and feasibility studies on WF defenses [22] could serve as valuable insights to further enhance the security and privacy of voice assistant systems.

9 CONCLUSION

In this paper, we introduced the WhiVo model with a variant of WF features that empowers adversaries to surpass current state-of-the-art models. Our work provides a nuanced understanding of the potential risks associated with VCF. This involves an in-depth feature analysis, leveraging both traditional WF features and TAM variants tailored specifically to VCF.

Our study delves into the variations in voice command traffic, considering influential factors such as device locations and user characteristics. This exploration contributes to a more realistic assessment of the challenges and implications posed by VCF, enriching the understanding of the security landscape surrounding voice assistant systems.

Furthermore, we introduced a voice command category fingerprinting. This approach enables the inference of voice command categories, even for unknown commands, presenting a more practical and scalable model for defending against potential privacy threats.

Finally, we have shared the dataset and the code used in the paper in a GitHub repository [1].

[1] https://github.com/AISec-VoiceFingerprinting/WhisperVoiceTrace-A-Comprehensive-Analysis-of-Voice-Command-Fingerprinting

ACKNOWLEDGMENTS

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2022-00155966, Artificial Intelligence Convergence Innovation Human Resources Development (Ewha Womans University)), and by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(No. RS-2022-00166669, RS-2023-00222385). We would like to express our gratitude to Goun Kim, a master's student in the Ewha AISec Lab, for her help with collecting the dataset.

REFERENCES

[1] 2,697 u.s. smart speaker owners (18-84 y/o) surveyed oct. 2022 - sep. 2023. Statista Consumer Insights.
[2] Alexa application. https://www.amazon.com/b?node=21576558011&ref=MARS_NAV_desktop_skills_feat.
[3] Alexa blog. https://alexaanswers.amazon.com/blog.
[4] deepvc-github. https://github.com/SmartHomePrivacyProject/DeepVCFingerprinting.
[5] Google nest help. https://support.google.com/googlenest/answer/7207759?hl=en&ref_topic=7196346.
[6] Google search help. https://support.google.com/websearch.
[7] shame-github. https://github.com/jock0rama/automated-voice-command-traffic-collection.
[8] Siri-apple. https://www.apple.com/siri/.
[9] Tts-narakeet. https://www.narakeet.com.
[10] Dilawer Ahmed, Aafaq Sabir, and Anupam Das. Spying through your voice assistants: realistic voice command fingerprinting. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2419–2436, 2023.
[11] Alireza Bahramali, Ardavan Bozorgi, and Amir Houmansadr. Realistic website fingerprinting by augmenting network traces. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1035–1049, 2023.
[12] Sanjit Bhat, David Lu, Albert Kwon, and Srinivas Devadas. Var-CNN: A data-efficient website fingerprinting attack based on deep learning. *Proceedings on Privacy Enhancing Technologies (PETS)*, 2019(4):292–310, 2019.
[13] Wladimir De la Cadena, Asya Mitseva, Jens Hiller, Jan Pennekamp, Sebastian Reuter, Julian Filter, Thomas Engel, Klaus Wehrle, and Andriy Panchenko. Trafficsliver: Fighting website fingerprinting attacks with traffic splitting. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1971–1985, 2020.
[14] Jiajun Gong and Tao Wang. Zero-delay lightweight defenses against website fingerprinting. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 717–734, 2020.
[15] Jamie Hayes and George Danezis. k-fingerprinting: A robust scalable website fingerprinting technique. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 1187–1203, 2016.
[16] James K Holland and Nicholas Hopper. Regulator: A straightforward website fingerprinting defense. *Proceedings on Privacy Enhancing Technologies*, 2:344–362, 2022.
[17] Jack Hyland, Conrad Schneggenburger, Nick Lim, Jake Ruud, Nate Mathews, and Matthew Wright. What a shame: Smart assistant voice command fingerprinting utilizing deep learning. In *Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society*, pages 237–243, 2021.
[18] Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. A critical evaluation of website fingerprinting attacks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 263–274, 2014.
[19] Marc Juarez, Mohsen Imani, Mike Perry, Claudia Diaz, and Matthew Wright. Toward an efficient website fingerprinting defense. In *European Symposium on Research in Computer Security (ESORICS)*, pages 27–46. Springer, 2016.
[20] Sean Kennedy, Haipeng Li, Chenggang Wang, Hao Liu, Boyang Wang, and Wenhai Sun. I can hear your alexa: Voice command fingerprinting on smart home speakers. In *2019 IEEE Conference on Communications and Network Security (CNS)*, pages 232–240. IEEE, 2019.
[21] Jianghan Mao, Chenyu Wang, Yanhui Guo, Guoai Xu, Shoufeng Cao, Xuanwen Zhang, and Zixiang Bi. A novel model for voice command fingerprinting using deep learning. *Journal of Information Security and Applications*, 65:103085, 2022.
[22] Nate Mathews, James K Holland, Se Eun Oh, Mohammad Saidur Rahman, Nicholas Hopper, and Matthew Wright. Sok: A critical evaluation of efficient website fingerprinting defenses. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 969–986. IEEE, 2023.
[23] Milad Nasr, Alireza Bahramali, and Amir Houmansadr. Defeating {DNN-Based} traffic analysis systems in {Real-Time} with blind adversarial perturbations. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2705–2722, 2021.
[24] Se Eun Oh, Nate Mathews, Mohammad Saidur Rahman, Matthew Wright, and Nicholas Hopper. GANDaLF: GAN for data-limited fingerprinting. *Proceedings on Privacy Enhancing Technologies (PETS)*, 2021(2):305–322, 2021.
[25] Se Eun Oh, Saikrishna Sunkam, and Nicholas Hopper. p1-FP: Extraction, classification, and prediction of website fingerprints with deep learning. *Proceedings on Privacy Enhancing Technologies (PETS)*, 2019(3):191–209, 2019.
[26] Andriy Panchenko, Fabian Lanze, Jan Pennekamp, Thomas Engel, Andreas Zinnen, Martin Henze, and Klaus Wehrle. Website fingerprinting at internet scale. In *NDSS*, 2016.
[27] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *Workshop on Privacy in the Electronic Society (WPES)*. ACM, 2011.
[28] Mohammad Saidur Rahman, Payap Sirinam, Nate Mathews, Kantha Girish Gangadhara, and Matthew Wright. Tik-tok: The utility of packet timing in website fingerprinting attacks. *arXiv preprint arXiv:1902.06421*, 2019.
[29] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. Automated website fingerprinting through deep learning. *arXiv preprint arXiv:1708.06376*, 2017.
[30] Meng Shen, Kexin Ji, Zhenbo Gao, Qi Li, Liehuang Zhu, and Ke Xu. Subverting website fingerprinting defenses with robust traffic representation. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 607–624, 2023.
[31] Meng Shen, Yiting Liu, Liehuang Zhu, Xiaojiang Du, and Jiankun Hu. Fine-grained webpage fingerprinting using only packet length information of encrypted traffic. *IEEE Transactions on Information Forensics and Security*, 16:2046–2059, 2020.
[32] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. Deep fingerprinting: Undermining website fingerprinting defenses with deep learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1928–1943, 2018.
[33] Chenggang Wang, Sean Kennedy, Haipeng Li, King Hudson, Gowtham Atluri, Xuetao Wei, Wenhai Sun, and Boyang Wang. Fingerprinting encrypted voice traffic on smart speakers with deep learning. In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 254–265, 2020.
[34] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. Effective attacks and provable defenses for website fingerprinting. In *USENIX Security Symposium*, pages 143–157, 2014.
[35] Tao Wang and Ian Goldberg. Improved website fingerprinting on tor. In *ACM workshop on Workshop on privacy in the electronic society (WPES)*, pages 201–212, 2013.
[36] Tao Wang and Ian Goldberg. {Walkie-Talkie}: An efficient defense against passive website fingerprinting attacks. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1375–1390, 2017.

APPENDIX

A VOICE COMMAND TRACE COLLECTION METHODOLOGY.

The voice command traces were collected by capturing network traffic while asking and receiving responses from the Alexa device. To automate these processes, we implemented the voice command trace collection tool (Figure 8) by extending previous collection methodology [17], primarily comprised of three components: the Raspberry Pi, the AIexa device, and a general speaker. The microcomputer (Raspberry Pi) is connected to the Ethernet and utilizes the hostapd daemon to establish a Wi-Fi access point. The Alexa device communicates with the Amazon cloud server through this Wi-Fi connection. Then, we executed the tcpdump command on the Raspberry Pi to capture the network traffic between the Alexa device and the cloud server.

The voice command traces for 600 commands were collected in parallel using five sets of the Alexa device, the speaker, and the Raspberry Pi. The process for collecting voice command traffic is outlined as follows. When the voice command audio file is played, and the voice signal is received by the Alexa device, communication with the cloud server started and at the same time, the packet capture initiated. Then, the response from the Alexa device is recorded

Model	Hyper Param	Search Space	Choice
DF	Block #	3-5	4
	Conv filter #	16-512	[32,64,128,256]
	Conv kernel size	4-16	8
	Conv dropout	0.1-0.2	0.1
	Conv Activation	ReLU, ELU, LeakeyReLU	ReLU
	FCL layer #	2-3	2
	FCL node #	512-4,096	4,096
	FCL dropout	0.5-0.9	[0.9,0.8]
	Batch size	64-512	256
	Epoch	300-1,000	600
Ensemble	Conv layer #	0-2	0
	FCL layer #	2-4	3
	FCL node #	2,048-10,000	4,096
	FCL dropout	0.5-0.9	0.7
	Optimizer	Adam, SGD	SGD
	Loss	Cate-Cross, Mean-Squared	Catel-Cross
	Learning rate	0.00005-0.05	0.003
	Batch size	64-512	256
	Epoch	300-1,000	600
VAV	l	100-10,000	7,500
	D	40-60	60
	δ	0-0.006	0

Table 6: Hyperparameter Tuning (Note that Conv: Convolution layer, FCL: Fully Connected Layer, Block: Convolution layer block, Cate-Cross: Categorial-Crossentropy).

in the audio file using the microphone. This is one batch. Then, the next batch begin with playing the next voice command audio file.

B HYPERPARAMETER TUNING DETAILS.

In the WhiVo model, since the feature is different from that of the SHAME model, we decided to identify good parameters to gain more effectiveness. The search spaces for hyperparameters and chosen values are reported in Table 6. Since we used three DF models to generate the ensemble model feature vectors, we tuned the parameters of both the DF and ensemble models.

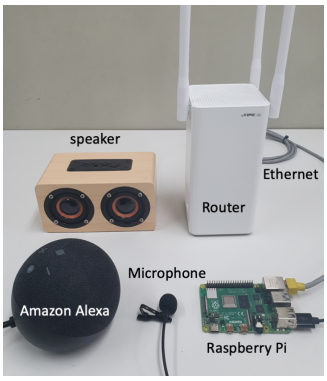


Figure 8: Our voice command traffic collection tool.

Index	Commands
0	What's the high today?
1	Wow windy is it?
2	Wow much is the humidity?
3	What's the date?
4	What restaurants are nearby?
5	Is there a post office nearby?
6	Is the post office open?
7	where's the nearest gym?
8	How's traffic?
9	What's the weather tomorrow?
10	What's the weather for the next 4 days?
11	Where are pharmacies nearby?
12	How long will it take to get to Costco?
13	What's the temperature outside?
14	What time does the Apple Store close?
15	What's the fastest way to get to the airport from here?
16	How many days until christmas?
17	How many days until Thanksgiving?
18	What is the date tomorrow?
19	What time is it?

Table 7: Commands used in location experiments

For the DF model, we attempted to increase and decrease kernel size and the number of blocks, convolution filter and fully connected layers. However, due to underfitting or overfitting issues resulting in low accuracy, we reverted to the original values. Regarding activation functions, we found that using ReLU for all blocks, as tuned in SHAME, yielded better results compared to using ELU in the first block and ReLU in the rest, as suggested by DF. When using PAT, IPD, and Packet size features, LeakyReLU had a higher accuracy, but it did not affect VAV. In addition, dropout, epoch, and batch size were adjusted

For the Ensemble model, we experimented with increasing and decreasing the number of convolution layers and fully connected layers, confirming that underfitting and overfitting occurred. When adjusting the optimizer and corresponding learning rates, we found that the combination of SGD and a learning rate of 0.03 was the most suitable. In addition, we tried to find the optimal dropout rate, loss, batch size, and epoch.

However, as the size of the input increased, it was confirmed that the increase in the number of fully connected nodes affected the improvement of accuracy, and 4096 became the most optimized number of nodes.

C COMMANDS OF LOCATION EXPERIMENTS.

Table 7 contains commands that we hypothesize Alexa utilizes the user's location information for. Consequently, responses and traffic patterns for these commands may vary depending on the user's location. This set encompasses 20 commands collected through Alexa devices set to locations in five countries: the US, UK, Germany, India, and Canada.

Algorithm 1 TF-IDF-Based Clustering Algorithm

Require: The set of commands, S
Ensure: The set of clusters, G

$thr \leftarrow 1.8$ ▷ Threshold
$G \leftarrow$ empty list ▷ Set of clusters
for each $x \in S$ **do**
  $v \leftarrow TF_IDF(x)$ ▷ TF-IDF Vectorizer
  $find_group \leftarrow False$ ▷ True if group is assigned
  for each $c \in G$ **do**
    if $find_group$ **then** break
    end if
    $g \leftarrow$ empty list
    for each $m \in c$ **do**
      if $E(v, m) \leq$ thr **then** ▷ E: Euclidean distances
        $c \leftarrow c \cup [v]$
        $find_group \leftarrow True$
        break
      end if
    end for each
    if not $find_group$ **then**
      $g \leftarrow g \cup [v]$
      $G \leftarrow G \cup g$
    end if
  end for each
end for each

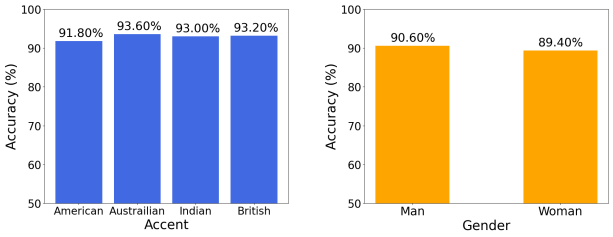


Figure 10: The impact of users' genders and demographics on the WhiVo performance.

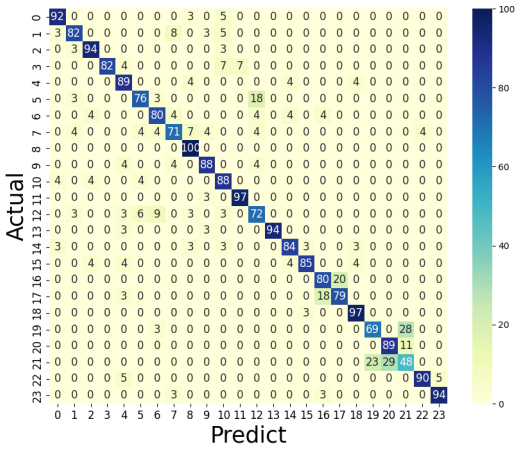


Figure 9: Confusion matrix of Manually Cluster Discovery(Unit: %).

D MANUAL CLUSTER DISCOVERY RESULTS.

The voice category fingerprinting with manual cluster discovery achieved an accuracy of 84.0% to correctly identify categories of voice command traces. Based on the confusion matrix shown in Figure 9, we observed that, excluding category 19 and category 21, most categories demonstrated a prediction accuracy of over 70%. This implies that the majority of commands were correctly inferred into their respective categories.

E IDENTIFIED CATEGORIES USING MANUAL CLUSTER DISCOVERY AND TF-IDF-BASED CLUSTERING

When manually assigning categories to 178 commands, we established 24 categories, with each category consisting of 5 to 10 commands. The listed categories, such as business, finance, local search, weather, etc., along with their corresponding commands, are presented in Table 8.

We additionally automated the process of generating voice command categories using TF-IDF-Based Clustering detailed in Algorithm 1. Table 9 displays categories and their commands after applying TF-IDF-Based Clustering to group 600 commands into clusters.

Remarkably, we observed common clusters appearing across both Table 8 and 9. For instance, Category 0 in both tables represents commands relevant to stock price queries.

F HUMAN FACTOR RESULTS

Figure 10 shows that user diversity, including gender and demographics, has minimal impact on voice command trace analysis. Both gender and demographic factors show only slight variations in the accuracy of the WhiVo model, indicating its robustness across different user profiles.

Cluster	Commands	Cluster	Commands
Category 0 (Business)	What's the stock price of general electric?		How old is Serena Williams?
	What's the Google stock price?		How old is Queen Elizabeth?
	How's the stock market doing today?		How tall is Barack Obama?
	What is the stock price of General Motors?		How tall is Steph Curry?
	What is the price of Monero?	**Category 13 (Translations)**	How do you say 'table' in Portuguese?
	What is the stock price of Tesla?		How do you say 'I love you' in French?
	Where's the London Stock Exchange today?		How do you say 'good morning' in Japanese?
Category 1 (Finance)	How many euros are in a dollar?		How do you say 'thank you' in German?
	What is the price of bitcoin?		How do you say hello in Japanese?
	What is the dollar to euro exchange rate?		How do you say cucumber in Spanish?
	What is the price of silver?		How do you say 'nice to meet you' in Mandarin?
	What is the price of gold?		How do you say 'thank you' in Italian?
Category 2 (Sports)	What are the basketball scores?		How do you say 'Good Morning' in German?
	What's the score of the Celtics game?		How do you say happy birthday in Chinese?
	When is the NBA all star game?	**Category 14 (Local Search)**	What restaurants are nearby?
	Who is second in the NBA Western Conference?		What's the closest florist?
	How are the Warriors doing?		Is there a post office nearby?
	What is the Warriors record?		Where's the nearest gym?
	Who are Warriors playing today?		Where are pharmacies nearby?
	Who are Warriors playing this week?		Any ATMs nearby?
	Did Warriors win or lose?		Find pizza restaurants nearby.
	NBA standings.		Where's the closest coffee shop?
Category 3 (Weather)	What's the weather in New York?		Where's the closest gas station?
	What's the weather tomorrow?		What are some flower shops nearby?
	What's the weather for the next 4 days?	**Category 15 (Traffic)**	How's traffic?
	What's the weather this weekend in Mountain View?		How is traffic to 100 Main Street?
	What's the weather in London?		How is traffic to the stadium?
	What's the weather in Celsius?		How is traffic to Los Angeles?
	What wil the weather be like between 1st and 5th?		How's the traffic around downtown San Francisco?
	What's the weather like tomorrow morning?		What is my traffic report?
	What is the weather for Sunday?		How long will it take to walk, train, bus, or bike to the library?
	What is the weather?		How long will it take to get from work to San Francisco International Airport?
Category 4 (Day 1)	When is daylight saving time?		How long will it take to get to Costco?
	When does daylight saving time end?	**Category 16 (Story)**	Tell me a riddle.
	When is April Fools Day?		Tell me a limerick.
	When is Shark Week?		Tell me a haiku.
	When is the autumn equinox?		Tell me a story.
Category 5 (Day 2)	How many days until Valentines day?		Give me a shark limerick.
	How many days are in September?		Tell me a pun.
	How many days in a year?		Recite a haiku.
	How many days until christmas?		Give me a random fact.
	How many days until Thanksgiving?	**Category 17 (Joke)**	Tell me a Star Trek joke.
Category 6 (Time)	What time is it in Seoul?		Tell me a golf joke.
	What time is it in London?		Tell me a ninja joke.
	What time is it in Berlin?		Tell me a joke about you.
	What is the time in Singapore?		Tell me a shark joke.
	What time are the Emmys?		Tell me a basketball joke.
	What time is it?	**Category 18 (Conversation)**	How are you?
Category 7 (Measurements 1)	What is 300 degrees Fahrenheit in Celsius?		Who inspires you?
	How many miles are in 10 kilometers?		Can you do an impression?
	How many pounds are 2 kilos?		Are you left-handed?
	2 gallons are how many liters?		What's on your mind?
	How many kilometers in a mile?		You talkin' to me?
	What's 16 ounces in pounds?	**Category 19 (World Records)**	What's the world record for pushups?
	How many grams are in a pound?		What's the world record for squat?
	How many ounces in a pound?		What's the world record for handstand?
	What is 90 degrees Fahrenheit in Celsius?		What's the world record for Lego?
Category 8 (Measurements 2)	How many calories are in an egg?		What's the world record hottest temperature?
	How much fiber in kale?		What's the world record snowfall?
	How much sugar is there in a Coke?		What is the world record pumpkin?
	How many calories are in an ounce of cheddar cheese?		What's the world record for showering?
	How much caffeine is in coffee?		What's the world record for staring?
	How much fiber is in broccoli?		What's the world record for tickling?
	How many calories are in a donut?	**Category 20 (Animal)**	How big is the smallest cat?
Category 9 (Math)	What's 99 divided by 12?		How high can a guinea pig jump?
	What's the area of a circle?		How long can a dog remember?
	Calculate Pi.		How long is one day for dogs?
	What is 15% of 92?		What colors can fish see?
	What's the 5th root of 97?		How long is one dog minute?
	What's 20% of 80?		What colors can guinea pig see?
	What's the square root of 2209?	**Category 21 (Music Award)**	Why are the grammys called the grammys?
	What is zero divided by zero?		Who won the most grammys at the grammys?
Category 10 (Population)	What's my 2020 US Census recap?		Who decides the grammys?
	What is the population of Seattle?		Where is coachella held?
	How many people live in the US?		How many people attend coachella?
	How many people live in Wyoming?		How long has coachella been going on?
	How many people live in China		What is the most expensive vinyl record?
	How many people live in Cincinnati?		When was the first vinyl record released?
Category11 (Word)	What's the definition of adept?	**Category 22 (Yellowstone)**	Where is Yellowstone National Park?
	What does circumlocution mean?		Why is Yellowstone Park called Yellowstone?
	What does "gregarious" mean?		Which president declared Yellowstone National Park?
	What does gluttony mean?		When is the best time to visit Yellowstone?
	How do you spell…misspell?		Can you see all of Yellowstone in one day?
	How do I spell vigorously?		Can you kayak in Yellowstone Lake?
	How do you spell appreciate?	**Category 23 (President)**	Who was Vice President when Johnson was President?
	What is the longest word?		What years was President Nixon President?
Category 12 (People)	How old is Tom Brady?		How old was President Reagan when he became President?
	How old is Beyonce?		Which President was only President for thirty two days?
	How old is Henry Winkler?		Name the first five Presidents.

Table 8: Categories generated by Manual Cluster Discovery.

Cluster	Commands	Cluster	Commands
Category 0 (Price)	What's the stock price of general electric?		What is the capital of Spain?
	What's the Google stock price?		What is the history of Labor Day?
	What is the price of bitcoin?		What is the purpose of Earth Day?
	What is the stock price of General Motors?		What do people usually do on Earth Day?
	What is the price of silver?		What was the first capital of the United States?
	What is the price of gold?		What's the history of oktoberfest?
	What is the price of Monero?	**Category 5 (Time)**	What time is it in London?
	What is the stock price of Tesla?		What time is it in Berlin?
Category 1 (Sports)	When do the Cowboys play next?		What is the date tomorrow?
	When are the Warriors playing next?		What is the time in Singapore?
	Who are Warriors playing today?		Good Morning.
	Who are Warriors playing this week?		What time is it?
	Who's playing in the NBA today?		What time zone is the north pole in?
	NBA standings.	**Category 6 (Weather)**	What's the uv index tomorrow?
	What is the AFC North Standings?		What's the weather tomorrow?
	When is the NBA all star game?		What's the weather for the next 4 days?
	Who is second in the NBA Western Conference?		Will it rain tomorrow?
	When is the Patriots' first game?		What's the weekend forecast?
	Where is the MLB All-Star game being played?		What's the 7 day forecast?
	What are the basketball scores?		What's the weather this weekend in Mountain View?
	What were yesterdays scores?		What's the weather in London?
	What are the standings in the English Premier League?		What's the weather in Celsius?
	What is the score of the Eagles game?		What's the weather like tomorrow morning?
	What is the score of the Red Sox game?		What is the weather in Canberra, Australia?
Category 2 (Quantity)	How many euros are in a dollar?		What is the weather for Sunday?
	How many ounces are in a cup?		What is the weather?
	How many miles are in 10 kilometers?	**Category 7 (World Records)**	What's the world record for the longest Monopoly game?
	How many kilometers in a mile?		What's the Jenga world record?
	What's 16 ounces in pounds?		What's the world record hottest temperature?
	How many milliliters are in a cup?		What's the world record snowfall?
	How many grams are in a pound?		What is the world record pumpkin?
	How many ounces in a pound?		What's the facetime world record?
	How many teaspoons are in a tablespoon?		What's the world record for showering?
	How many calories are in a donut?		What's the world record for staring?
	How many days are in September?		What's the world record for tickling?
	How many days in a year?		What is a crazy world record?
	How many days until christmas?		How to break a world record?
	How many days until Thanksgiving?		What's the biggest Guinness world record?
Category 3 (Story)	Tell me a basketball joke.		What's the world record for pushups?
	Tell me a pun.		What is the planking world record?
	Tell me a shark joke.		What's the world record for squat?
	Tell me a tongue twister.		What's the world record for handstand?
	give me a shark limerick.		What is the Tetris world record?
	Tell me a joke about you.		What is the Minecraft world record?
	Give me a random fact.		What's the Pac-Man world record?
Category 4 (Day)	What's the animal of the day?		What's the world record for Lego?

Table 9: Categories generated by TF-IDF-Based Clustering.