Home Assistant Sonar
Project Report

EE209AS: Security and Privacy for Embedded Systems, Cyber-Physical Systems, and Internet of Things
UCLA

June 14, 2019

Dennis Shim
Seraphine Goh

I. Introduction

The presence of IoT devices within the home is constantly growing as new technology is developed to assist users in various aspects of their lives. One such technology is the virtual assistant, an interactive software interface that can perform various tasks and provide a variety of services when prompted by the user. There exist several methods by which a user may interact with these virtual assistants, such as through text or voice. Furthermore, these virtual assistants may be incorporated into many types of platforms. One such case of this is the integration of virtual assistants and smart speakers, with Google Home and Amazon Echo as two prominent examples. Smart speakers are voice-activated IoT devices that provide a variety of services to the user when prompted by the appropriate “hot word” (e.g. "OK Google", "Alexa") and a following command. Since these smart speakers can be activated by the “hot word” at any point in time, the built-in microphones must be constantly active and waiting to perceive the designated command. As such, there are a variety of privacy concerns surrounding the use of smart speakers within the home due to their susceptibility to adversarial attacks.

Many adversarial attacks have been launched against smart speakers. One example of a more benign attack was when a Burger King TV advertisement had the audio: "OK Google, what is the Whopper Burger?" This triggered many Google Home assistants to read off a list of ingredients from Wikipedia, which were embellished by Burger King's marketing chief. This advertisement, however, was effectively a proof of concept of an attack against smart speakers that worked when nobody was around.

Since then, attacks on smart speakers have evolved, and sounds that are imperceptible to humans have been shown to be able to trigger home assistants. For example, very high frequency sounds whose harmonics are expressed on the microphones, or specially designed noise have both been shown to trigger these smart speakers. This can be used to nefariously open smart door locks, transfer money from bank accounts, or force smart speaker owners to listen to cat facts. These attacks can be performed by audio that does not even originate from inside of the room. User voice identification can help against some of these attacks, but record and replay attacks can circumvent this form of defense. Thus, there is incentive to create a variety of defenses against such attacks, so that even if one is circumvented, it is hard to break through all of them.

One method is to use the hardware already built into the speakers to detect whether a person is in the room or not. In this project, we used both speakers and the microphones to see if we could develop a "sonar" to detect the presence of a person.

II. Problem Statement

For our project, we consider the possibility of an adversarial attack upon an unsuspecting smart speaker. This attack would likely occur from a non-human device as the source, such as a voice command projected from a nearby speaker or other similar forms of attack. Thus, a possible defense against a malicious attempt to interact with the targeted smart speaker is to identify and locate a human presence within the room.

We first look at the binary case where we identify whether or not a human is present in the room. This identification would defend against adversarial attacks that attempt to take advantage of the user’s absence to perform a covert attack on the smart speaker. With this defense mechanism, the smart speaker would be able to determine whether the source of the received voice command was some non-human device.

In the multi-class case, we identify a human’s location in the room with respect to the smart speaker, i.e. in "front", "behind", to the "right", or to the "left" (see Figure 1 for clarification of these directions). This location identification would defend against adversarial attacks that may occur when the human is present in the room. One such instance of this could be a malicious command communicated by a television ad. If the smart speaker is able to locate the human in the room, it would therefore be able to determine the probability that the human was indeed the source of the received voice command.

III. Prior Work

EchoSafe [1] provides the inspiration for this project. In EchoSafe, the authors were able to detect user presence using 1 kHz sonar with 93.13% accuracy. In this project, we will attempt to reproduce the results of EchoSafe, which used a binary Random Forest classifier with data whose features are pruned by the Relief-F algorithm. In addition, we will collect additional data to expand on the results of EchoSafe.

One defense mechanism against voice command attacks is done using automatic speaker verification. However, there are still issues with fingerprinting the voice in this way, as there is a limited amount of data available to fingerprint an individual. Many solutions have been created to alleviate this problem, allowing for automatic speaker verification to reach a reasonable quality [2]. This makes it a good candidate for protection against attacks on smart speakers.

Furthermore, using a microphone array, it is possible to determine the angle of arrival for a given sound (e.g. a voice command) used to trigger a smart speaker. Combining this knowledge of sound direction with the computed sonar location of the human, we would therefore be able to detect a possible adversary attack if there exists a mismatch between the two directions.

IV. Technical Approach & Experimental Methods

We divided the development of this project into four key sections:

A) Data Collection
B) Feature Extraction
C) Feature Selection
D) Machine Learning Evaluation

A. Data Collection

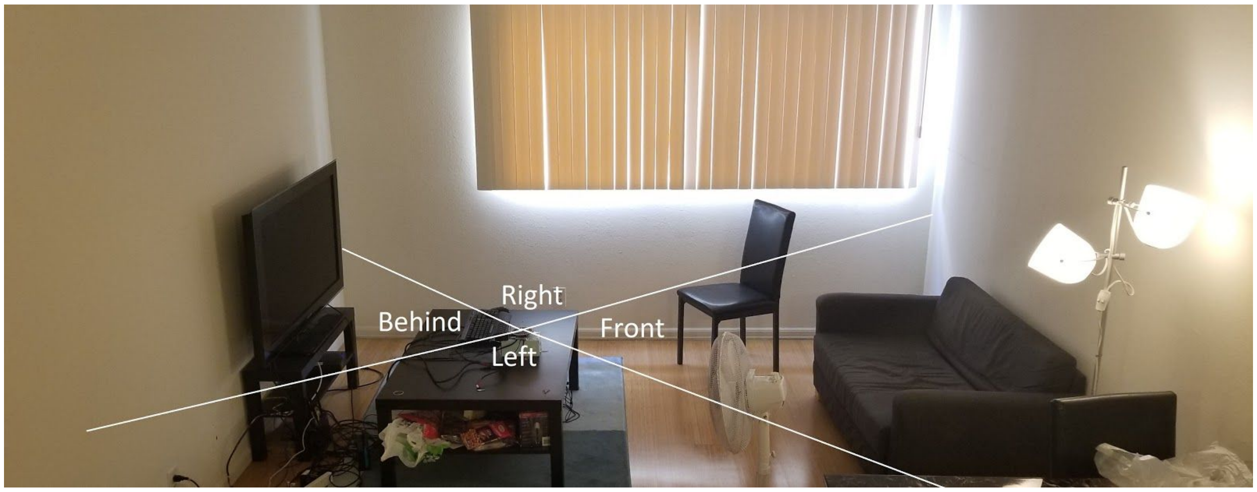

Figure 1: Experimental setup.

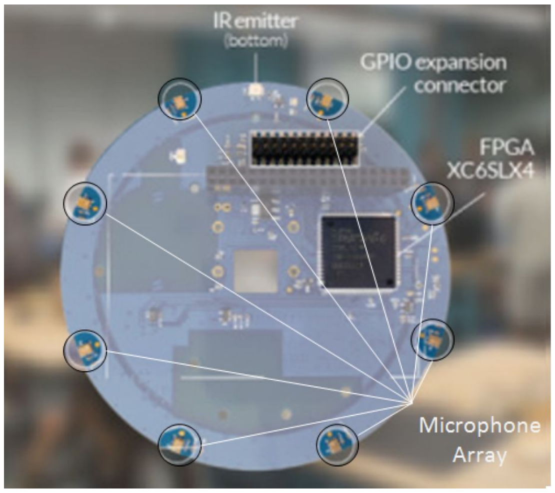

Figure 2: MATRIX Creator board.

The MATRIX Creator board (shown in Figure 2) and a Raspberry Pi 3 were used to record audio data. The MATRIX Creator board has an array of 8 microphones positioned uniformly in a circle. Sound was played using a speaker pointed up to simulate an omnidirectional speaker. The microphones are sampled at 40 kHz, so all frequencies in the audio spectrum (20 Hz to 20 kHz) can be captured. Audio signals from each of the 8 microphones are recorded, as well as a beamformed signal which captures the angle of the microphone. All 9 audio signals are used in the classification of the data.

Data was collected in the living room of an apartment. The experimental setup is shown in Figure 1. As shown in the figure, four different positions were used, and are indicated by "front" (towards the couch), "behind" (towards the TV), "right" (towards the window), and "left" (away from the window/towards the front door).

To collect data, a 1 kHz frequency tone was played for 1 second, and then audio from the 8 microphones of the Matrix Creator was recorded for 4 seconds. This results in the generation of eight 4 second .WAV files and one beamformed 4 second .WAV file. Data was collected and labelled when there was no subject in the room, and when one subject was in one of the four positions (front, behind, left, and right). The furniture (chairs, fan, and sofa) was moved around approximately every 15 trials. When the subject was there, they were either seated in different positions (approximately 30 trials per seated position with various furniture configurations), or standing in different positions (approximately 4 trials per standing position with various furniture configurations).

In addition, the frequency of the tone was also changed to determine if different frequencies could be used as sonar signals. A 20 kHz sound and 100 Hz sound were used, and data was taken when there was no subject, and when the subject was in the front. This is motivated by the fact that humans have difficulty hearing higher and lower frequencies, so a sonar system using sounds of these frequencies would be imperceptible or near-imperceptible to humans.

Data was also collected when the subject walked counterclockwise and clockwise around the table. The subject moved at variable speeds, with different paths, and different start/stop points.

A summary of the number of trials, labels, and positions can be found in Table 1.

Frequency of Tone	Subject Position	# of trials	Label (Binary Label)
1 kHz	No subject	455	0 (0)
	Front (couch-side)	400	1 (1)
	Behind (TV-side)	100	2 (1)
	Right (window-side)	200	3 (1)
	Left (front door-side)	200	4 (1)
20 kHz	No subject	210	0
	Front (couch-side)	200	1
100 Hz	No subject	210	0
	Front (couch-side)	210	1
1 kHz	Moving counterclockwise	100	1
	Moving clockwise	100	2

Table 1: Summary of number of trials, labels, and positions.

B. Feature Extraction

Each trial of data collection produces a total of 9 audio files, one for each microphone on the Matrix Creator and an additional file for the beamformed audio. A single audio file obtained from one trial contains 65,536 data points. We split these data points into 8 non-overlapping windows, resulting in 8,192 data points per window. For each window, we extract a total of 76 features: mean, median, standard deviation, max, skewness, kurtosis, and 70 MFCCs (Mel-Frequency Cepstral Coefficients). Mean, median, standard deviation, and max are standard statistical measurements with commonly known definitions that describe the distribution of data. Both skewness and kurtosis describe the shape of the data’s probability distribution. Skewness measures the lack of symmetry that exists amongst the given data points, while kurtosis measures the outliers present in the distribution. The MFCC coefficients capture characteristics of the frequency spectrum, and are popular features in audio classification. Combining the features from all 8 windows thus results in 608 features for a single trial for a given audio file and subsequently 5,472 total features for all 9 audio files. These features were then written to a corresponding data file for future use in feature selection.

It is worth noting that in our data collection, we found several trials that did not capture all 65,536 data points. We deemed these trials to be invalid and discarded them when forming the data files in order to maintain a consistent number of features per trial.

C. Feature Selection

In order to select the most important features to be used in the following machine learning classification, we compared the Relief-F algorithm and Principal Component Analysis (PCA) method separately. For the Relief-F algorithm, we used a K-nearest neighbor value equal to 10 (k = 10) and selected the top 125 most important features (N = 125) in order to maximize the resulting cross-validation classification accuracy. For PCA, we compared the resulting cross-validation accuracies for dimensionalities M = 50 and M = 125.

Relief-F is a feature selection method that ranks each feature based on its computed feature score. The feature scores are calculated using a K-nearest neighbor approach that increases the score when a difference in feature value is detected between neighboring points of differing classes and decreases the score when a difference is detected between points of the same class. The top N most important features are thus described by the first N indices in the rank matrix.

PCA, or Principal Component Analysis, is a method of dimensionality reduction that can be used to reduce the given dataset to dimension M while still retaining most of the vital information contained in the original dataset. In our case, this dimensionality reduction results in the M most important features.

D. Machine Learning Evaluation

We evaluated the performance for several different classifiers for both the binary case and the multi-class case. In particular, we considered the following.

Binary classifiers:

1. K-nearest neighbor (KNN)
2. Random Forest
3. Support vector machine (SVM) with Gaussian kernel
4. Support vector machine (SVM) with Linear kernel

Multi-Class classifiers:

1. AdaboostM2
2. LPBoost
3. Random Forest (RF)
4. RUSBoost
5. Subspace
6. SVM with Linear kernel
7. TotalBoost

For each classifier, we trained the corresponding cross-validated model with 10 folds in order to compute the classifier’s cross-validation loss and accuracy for the given dataset.

V. Analysis and Results

A. Evaluation of Feature Selection Methods

As mentioned previously, we considered the Relief-F and Principal Component Analysis (PCA) algorithms for feature selection and observed their respective impacts on classifier cross-validation accuracy.

Figure 3 demonstrates the results of running both the PCA algorithm and Relief-F algorithm with the number of selected features N = {50, 75, 100, 125} on a multi-class RF classifier.

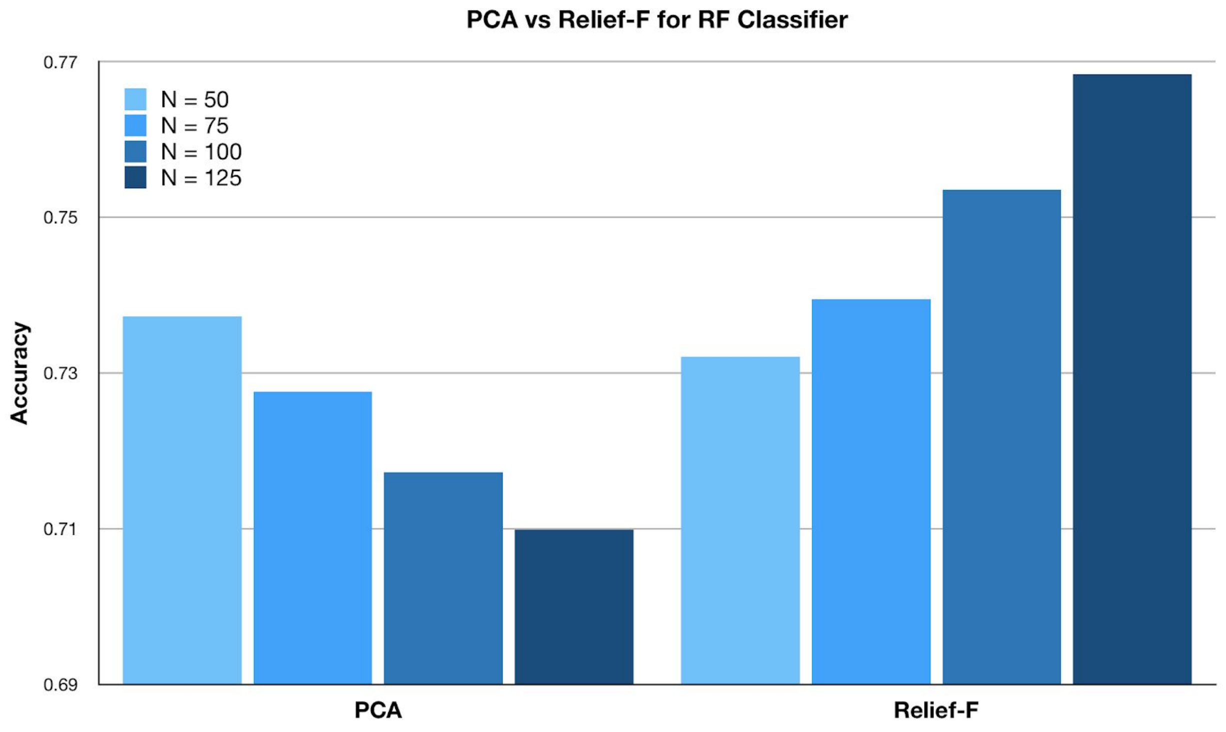

Figure 3: Cross-validation accuracy of multi-class RF classifier with varying N.

From the figure above, we can see that the Relief-F algorithm produced significantly higher accuracies than the PCA algorithm for values of N larger than 75. The results for Relief-F were computed using the default K-nearest neighbor value k = 10, as given by the documentation for the Relief-F algorithm. After sweeping through various possible values for k, we found that this default value produced the highest accuracies and so all following computations using Relief-F were done using k = 10.

Thus, we ultimately decided to use Relief-F for feature selection with N = 125 and k = 10 as we subsequently evaluate the performance of various classifiers in the binary and multi-class cases. In the following sections, we use Relief-F for pruning the data.

B. Evaluation of Machine Learning Classifier Performance

As mentioned previously, we evaluated the performance for several different classifiers for both the binary case and the multi-class case. After performing feature selection and pruning the data with N = 125, we obtained the following cross-validation accuracies.

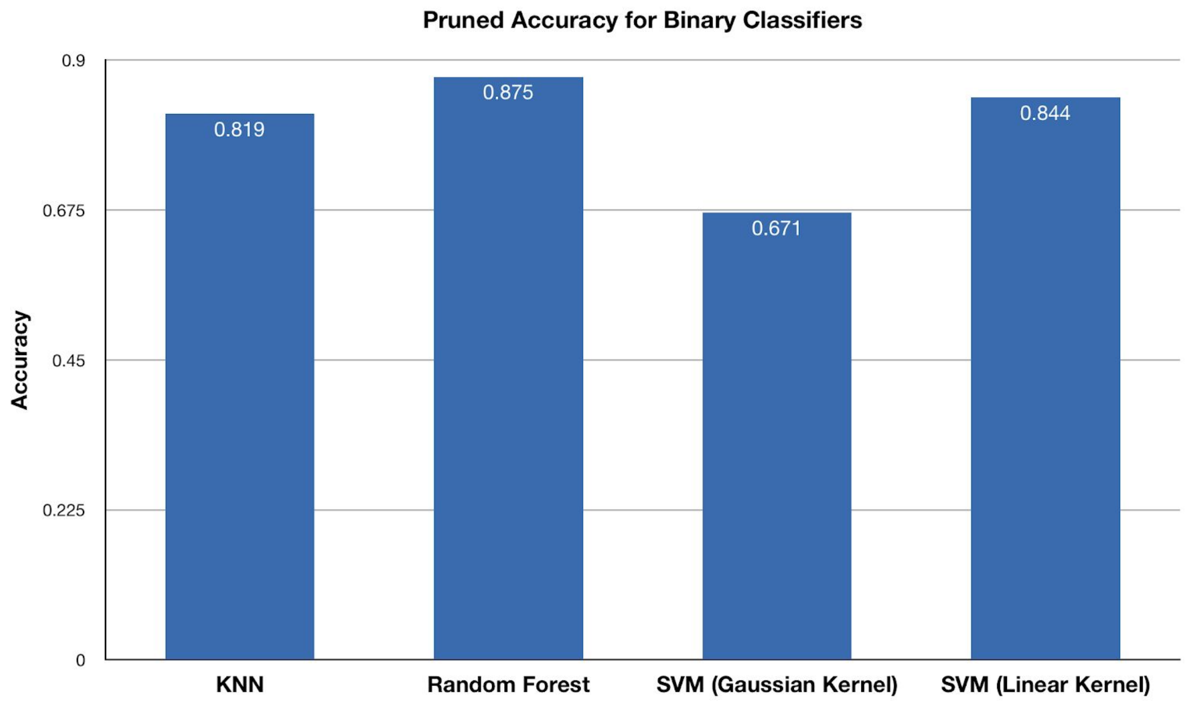

Figure 4: Cross-validation accuracy for data pruned with Relief-F (N = 125) with various binary classifiers.

In Figure 4, we can see that KNN, RF, and SVM with Linear Kernel have approximately comparable cross-validation accuracies while SVM with Gaussian Kernel performed significantly worse. Thus, the top three binary classifiers with the highest performance, from highest to lowest accuracy, are RF, SVM with Linear Kernel, and KNN.

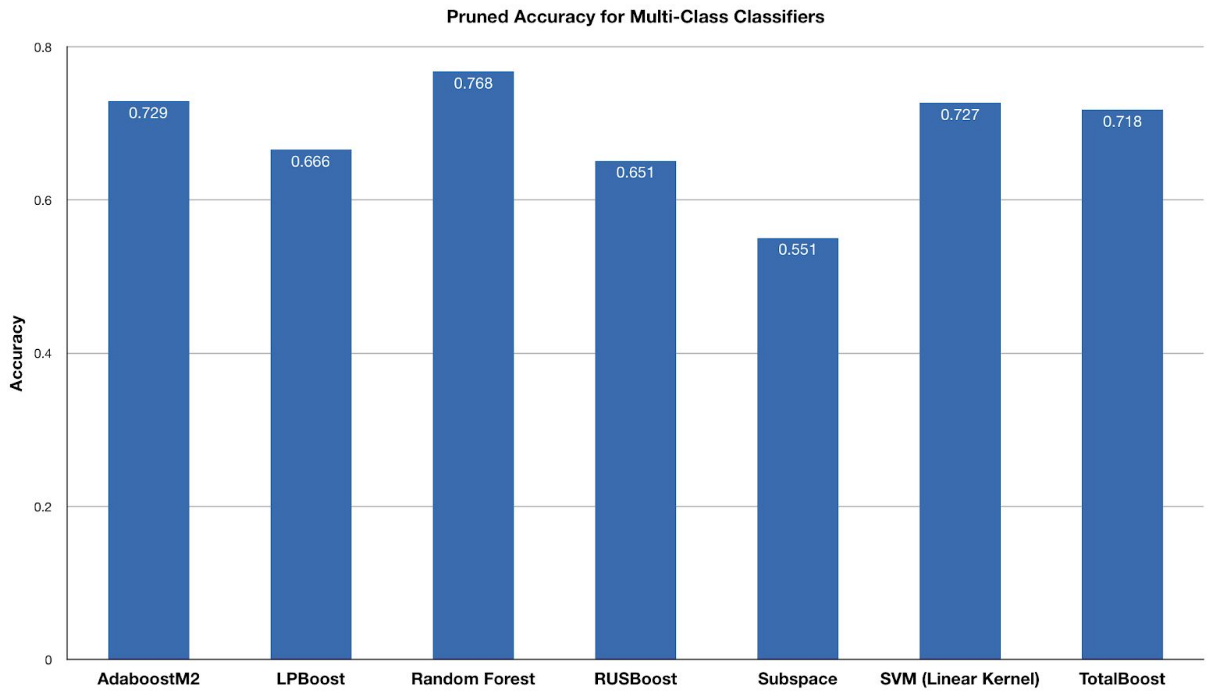

Figure 5: Cross-validation accuracy for data pruned with Relief-F (N = 125) with various multi-class classifiers.

In Figure 5, we can see that AdaboostM2, RF, SVM with Linear Kernel, and TotalBoost have approximately comparable cross-validation accuracies while LPBoost, RUSBoost, and Subspace performed significantly worse. Thus, the top three multi-class classifiers with the highest performance, from highest to lowest accuracy, are RF, AdaboostM2, and SVM with Linear Kernel.

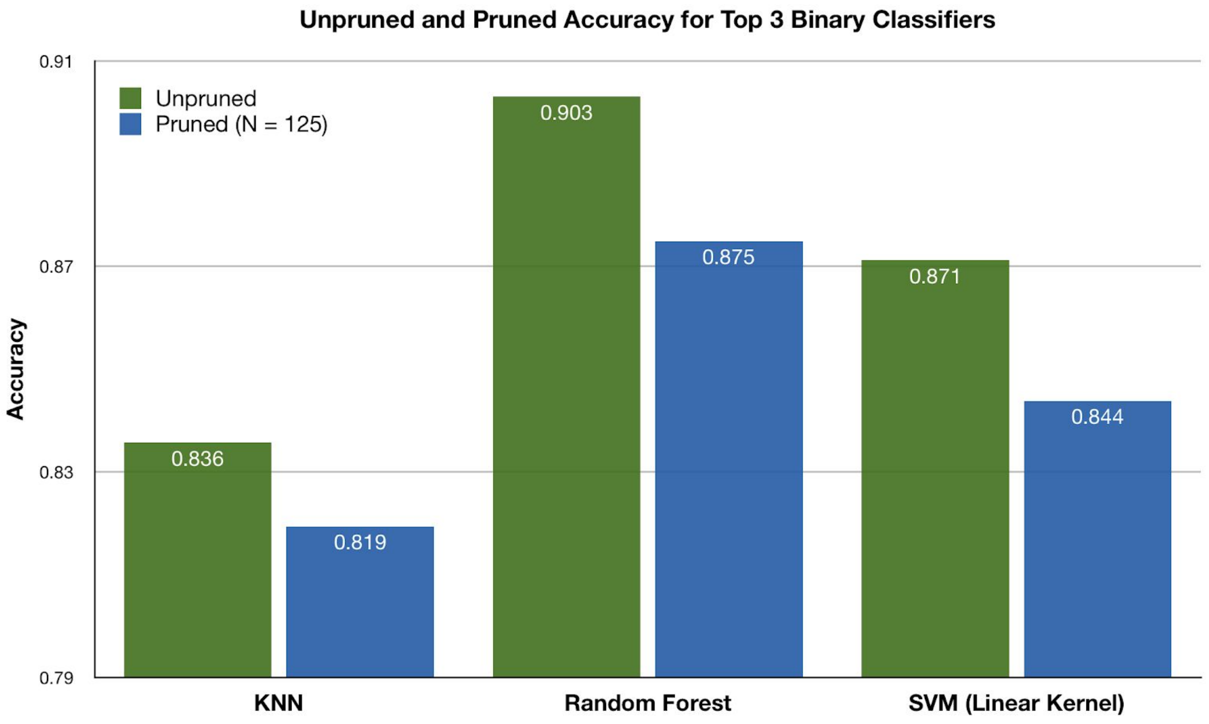

Figure 6: Comparison of unpruned and pruned accuracies for top 3 binary classifiers.

In Figure 6, we see that both the unpruned and pruned cross-validation accuracies of the RF classifier exceed those of the KNN and SVM with Linear Kernel. Thus, we determine that the best binary classifier to use for our purposes is the RF classifier pruned with N = 125.

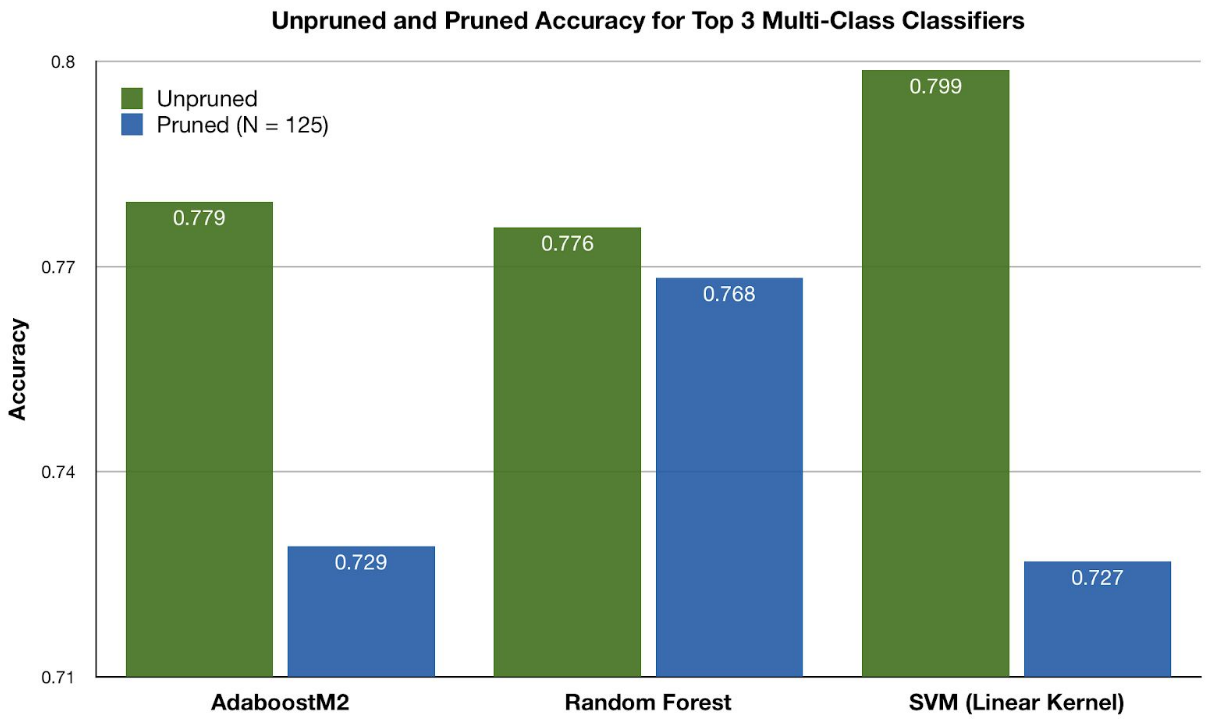

Figure 7: Comparison of unpruned and pruned accuracies for top 3 multi-class classifiers.

In Figure 7, we see that the pruned cross-validation accuracy of the RF classifier exceeds that of the AdaboostM2 and SVM with Linear Kernel. On the other hand, the unpruned accuracy of the SVM classifier exceeds that of the AdaboostM2 and RF classifiers. Due to the relatively large difference between the pruned and unpruned accuracies for both SVM and AdaboostM2, we predict that training a classifier on the unpruned data with all 5,472 features may lead to overfitting. Thus, a pruned version of the data should be used to obtain a more accurate portrayal of the true classifier performance. Under this assumption, we find that the RF classifier provides the most robust description of the given data since it demonstrates the smallest difference between the unpruned and pruned cross-validation accuracies. Therefore, we determine that the best multi-class classifier to use for our purposes is the RF classifier pruned with N = 125. This result is consistent with the previous analysis of binary classifier performance.

C. Impact of Frequency

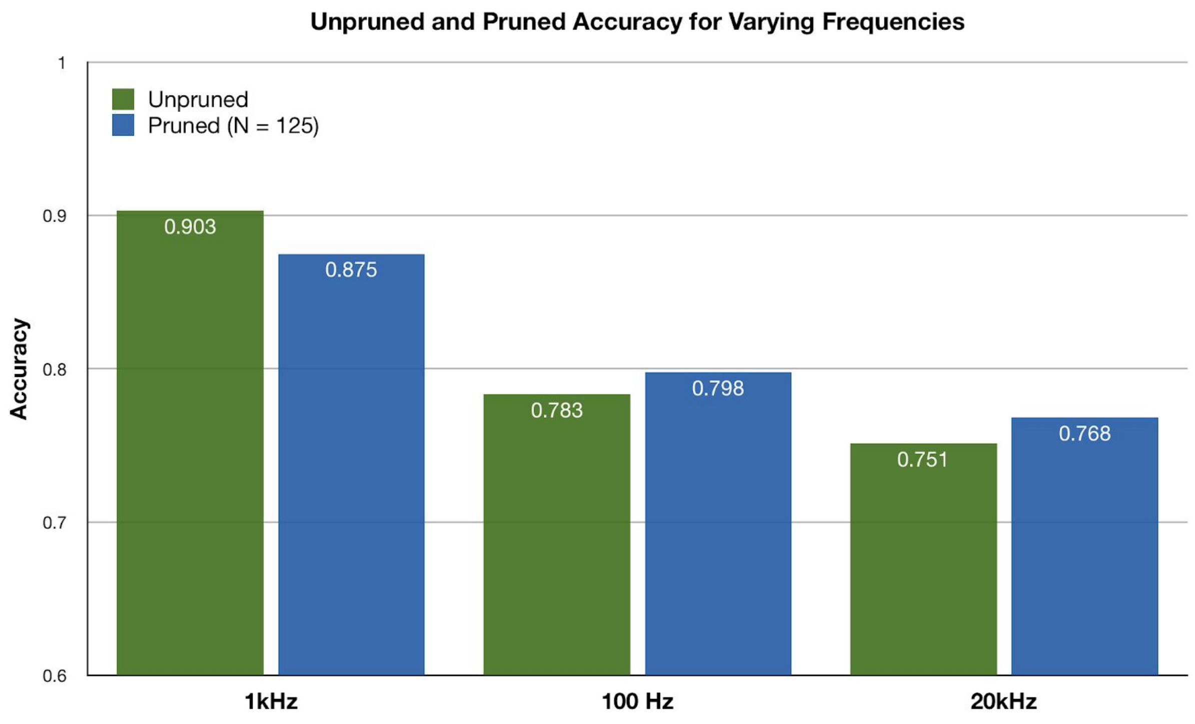

Figure 8: Comparison of unpruned and pruned accuracies for varying frequencies for binary RF classifiers.

Different frequency sounds were used as sonar. Traditional radar and sonar both use frequencies that are much higher 1 kHz, and ultrasonic sensors use frequencies above 20 kHz. We tested one lower frequency (100 Hz) and one higher frequency (20 kHz) to determine if accuracy could be improved by increasing or decreasing the frequency.

We compared the binary RF classifiers (checking if subject is present or not present) for each of the frequencies. The results are shown in Figure 8. The accuracy of the 1 kHz binary RF classifier is significantly better than both the 100 Hz and 20 kHz classifiers, so we used it for the multi-class classifiers. However, it is notable that the 100 Hz and 20 kHz achieve 79.8% and 76.8% accuracy, respectively, which are significantly better than random guessing (50%). These frequencies are at the edges of the hearing frequency spectrum, so hardware or sampling limitations of the speakers and microphones may reduce the accuracy of the audio recordings. Thus, using better hardware, it may be possible to achieve similar classifier accuracies using one higher or lower frequencies. This would make it possible to perform detection imperceptibly to the human in the room.

One additional test that we performed was to pass the 20 kHz and 100 Hz unpruned data through the 1 kHz RF model to see if it would be able to classify it correctly. However, we found that the resulting accuracy was 52.44% for the 20 kHz data, and 51.90% for the 100 Hz data.

This is essentially random classification, as the labels were binary. This could mean that the classification done using the 1 kHz RF model is done using features extracted from the reflected signal, rather than other confounding sounds (e.g. human breathing, background computer noise indicating human presence, etc.).

D. Detection of Movement

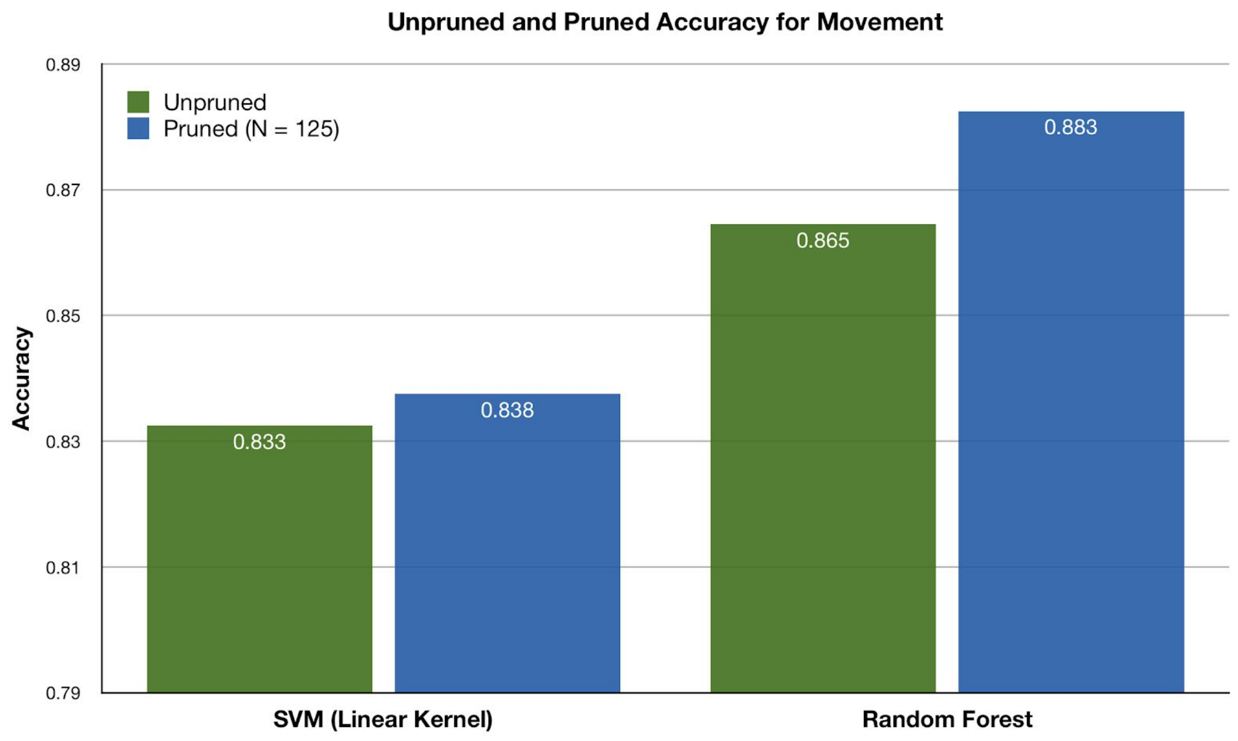

Figure 9: Comparison of unpruned and pruned accuracies for RF and SVM classifiers with the assumption of a moving subject.

We also attempted to classify data in which the subject was moving either counterclockwise (label 1) or clockwise (label 2) during the audio recording, with 100 trials for each class of movement. Classification was done including data where the human was present in the front and stationary (label 0). We used RF and SVM as multiclass classifiers for both the pruned and unpruned data. The accuracy results can be seen in Figure 9. The data pruned using Relief-F (N = 125) using the RF classifier had the best accuracy at 88.3%, which is quite high. However, looking at the confusion matrix in Table 2, we see that all of the stationary (label 0) data points are classified correctly, and the only errors arise from misclassification of data in which the subject is moving (labels 1 and 2). This indicates that the classifier is perfect at classifying stationary data when compared to CCW and CW data, but not very good at classifying the CCW and CW data. When performing a binary classification (i.e. CCW vs CW only), the accuracy and confusion matrix remain the same, as expected, and the accuracy is 74%. This is better than random guessing, and could potentially be improved on with more data.

However, one issue with the data is that moving is inherently more noisy than the stationary case, and although it was attempted to minimize noise due to movement, it is highly likely that this noise was used as an important feature. To confirm this, the most impactful features were extracted from the tree ensemble, and of the top 30 features, 7 are either mean or median features of the microphones, rather than MFCC coefficients. This could mean that louder noise in the audio is used to distinguish when the subject is stationary versus when they were moving.

		Predicted class		
		Stationary (Label 0)	CCW (label 1)	CW (label 2)
True class	Stationary (label 0)	200	0	0
	CCW (label 1)	0	78	22
	CW (label 2)	0	30	70

Table 2: Confusion matrix of RF classifier on data pruned using Relief-F (N = 125).

E. Other Considerations

When we revisit the purpose of our classifier, we note that some instances of misclassification may have greater negative implications than others. For instance, if our classifier were to determine that a human was present in the room when the room was actually empty, this would create an opportunity for adversarial attack since any commands issued to the smart speaker during this time would not trigger any warnings. Thus, we seek to weigh our classifier such that the misclassification error in the case when the true label is 0 (no human present) and the predicted label is 1 (human present) has a greater penalty than the other case of misclassification. To perform and analyze the effect of this weighting, we specify a cost matrix when fitting the binary RF classifier. The cost matrix in the binary case is of the form:

$$Cost = \begin{bmatrix} 0 & a \\ b & 0 \end{bmatrix}$$

Where the default values for a and b are 1.

In this definition of the cost matrix, the rows correspond to the true class while the columns correspond to the predicted class. Keeping $b = 1$, we increase a in order to more heavily penalize the misclassification between true label = 0 and predicted label = 1. The resulting confusion matrices for $a = \{1, 2, 3, 4, 5\}$ are shown below.

		Predicted class	
	a = 1	Label 0	Label 1
True class	Label 0	348	103
	Label 1	69	831

		Predicted class	
	a = 2	Label 0	Label 1
True class	Label 0	382	69
	Label 1	106	794

		Predicted Class	
	a = 3	Label 0	Label 1
True Class	Label 0	392	59
	Label 1	138	762

		Predicted class	
	a = 4	Label 0	Label 1
True class	Label 0	403	43
	Label 1	154	746

		Predicted Class	
	a = 5	Label 0	Label 1
True Class	Label 0	407	44
	Label 1	187	713

Table 3: Confusion matrices of binary RF classifier for $a = \{1, 2, 3, 4, 5\}$ in defined cost matrix.

From Table 3 above, we can clearly see the impact of the implemented cost function as the number of misclassifications where true label = 0 and predicted label = 1 is reduced significantly between $a = 1$ and $a = 2$. The misclassification for this particular case is further reduced for $a = \{3, 4, 5\}$.

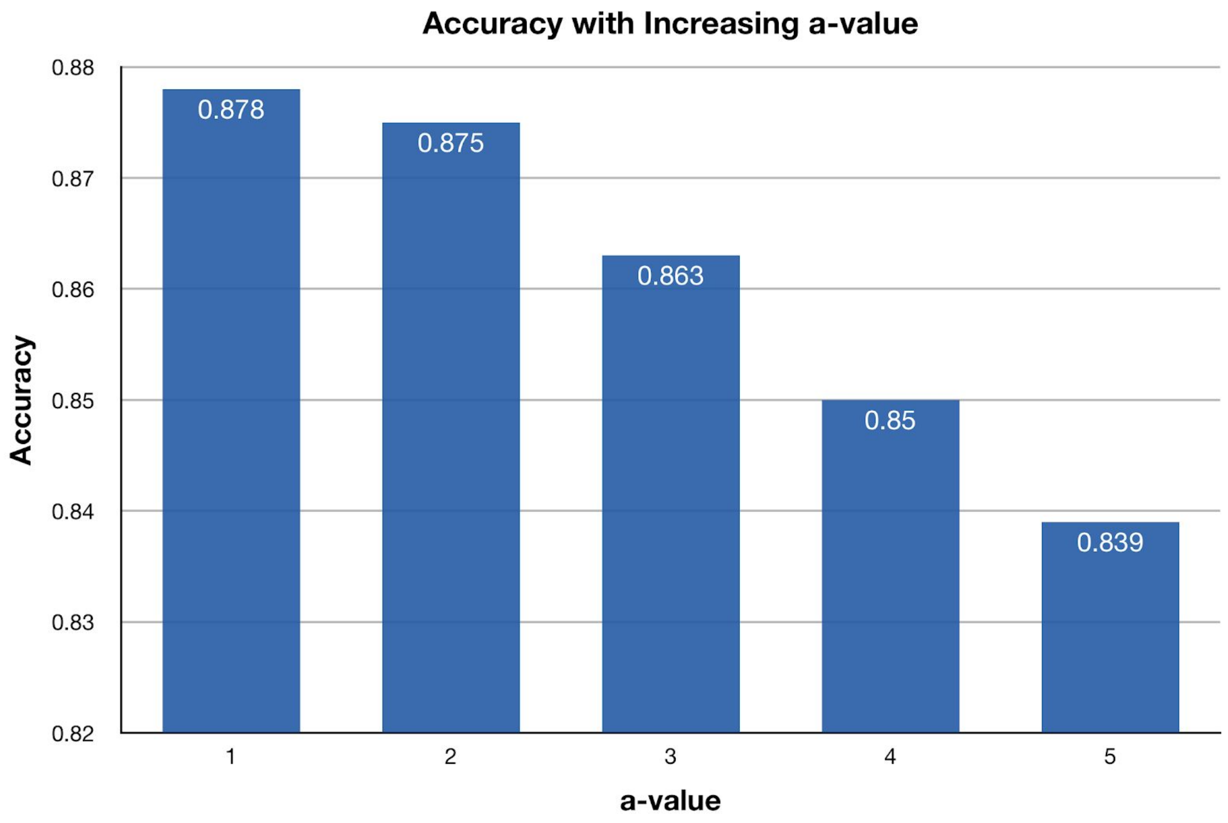

Figure 10: Cross-Validation accuracies for binary RF classifier with varying a-value.

While the classifier does indeed exhibit the desired behavior when an appropriate cost matrix is applied, we did note a drop in accuracy with increasing a as shown in Figure 10. In particular, the cross-validation accuracy of the binary RF classifier dropped from approximately 87.8% with $a = 1$ to approximately 83.9% with $a = 5$.

Though the implementation of a cost matrix does negatively impact the overall accuracy of the classifier, it may still be beneficial to retain such a weighting in order to ensure that the occurrence of the worst-case misclassification scenario we defined previously is reduced. We would likely use a cost matrix with $a = 3$ since the rate of undesired misclassification is nearly halved, seen in Table 3, at the expense of about 1.5% of accuracy. Therefore, this choice of cost matrix minimizes both the occurrence of worst-case misclassification and the corresponding loss in accuracy.

VI. Future Directions

The most viable application of these classifiers is to use them to classify whether a human is present or not using the RF binary classifier with 90% accuracy. If the human is present, then there is less probability of an adversarial attack overall, as it makes it harder for an attack to occur when a human is not present. In addition, if a user is present and an attack occurs, the smart speaker will likely give some audio feedback that would alert the user to a possible attack, further reducing the potential for attack.

Another direction would be to improve and/or extend the multiclass classifier. Improving the accuracy of the classifier could be done by taking more data and using a more effective classifier (e.g. neural network), adding more features, using better hardware, and using different frequencies (e.g. ultrasonic). The multiclass classifier could also have more classes. Currently, it is classifying quadrants of the room (i.e. 90 degree slices). Given enough data, it could potentially be extended to octets (45 degree slices) and further, or it could be turned into a regression problem to get an exact angle.

If the accuracy of the multiclass classifier is increased enough, it could be used in conjunction with the angle given by the beamformed audio signal to determine if a smart speaker user is in the expected quadrant (or other range of angles). This would provide even further protection against attacks on smart speakers from audio emanating from the wrong angle. When pairing this method with user voice identification, record and replay attacks could also be nullified.

VII. References

[1] Amr Alanwar, Bharathan Balaji, Yuan Tian, Shuo Yang, and Mani Srivastava. 2017. EchoSafe: Sonar-based Verifiable Interaction with Intelligent Digital Agents. In Proceedings of SafeThings'17, Delft, Netherlands, November 5, 2017, 6 pages.

[2] Arnab Poddar, Md Sahidullah, and Goutam Saha. 2018. Speaker verification with short utterances: a review of challenges, trends and opportunities. IET Biometrics 7, 2 (2018), 91–101. DOI:http://dx.doi.org/10.1049/iet-bmt.2017.0065

VIII. Appendix

Classifier	Dimensions (M)	Binary/Multiclass	Accuracy
Random Forest	50	Multiclass	0.737231
	75	Multiclass	0.727609
	100	Multiclass	0.717246
	125	Multiclass	0.709845
	150	Multiclass	0.709104

Table A.1: Accuracy when using RF multiclass classifier on data that has been transformed using PCA with various dimensions M.

Classifier	Number of Features	Binary/Multiclass	Accuracy
Random Forest	50	Multiclass	0.732050
	75	Multiclass	0.739452
	100	Multiclass	0.753516
	125	Multiclass	0.768320

Table A.2: Accuracy when using RF multiclass classifier on data that has its features pruned using the Relief-F algorithm with K = 10.

Classifier	Select Features	K-value (Relief-F)	Binary/Multi	Un/Pruned	Accuracy
AdaboostM2	5472 (all)	-	Multi	Unpruned	0.779423
	50	10	Multi	Pruned	0.686158
	125	10	Multi	Pruned	0.729090
KNN	5472 (all)	-	Binary	Unpruned	0.835677
	50	10	Binary	Pruned	0.794226
	125	10	Binary	Pruned	0.819393
LPBoost	50	10	Multi	Pruned	0.669643
	125	10	Multi	Pruned	0.666173
Random Forest	5472 (all)	-	Binary	Unpruned	0.903055
	50	10	Binary	Pruned	0.853442
	125	10	Binary	Pruned	0.874907
	5472 (all)	-	Multi	Unpruned	0.775722
	50	10	Multi	Pruned	0.732050
	125	10	Multi	Pruned	0.768320
RUSBoost	50	10	Multi	Pruned	0.614360
	125	10	Multi	Pruned	0.651369
Subspace	50	10	Multi	Pruned	0.530718
	125	10	Multi	Pruned	0.550703
SVM (Kernel: Gaussian	50	10	Binary	Pruned	0.799408
	125	10	Binary	Pruned	0.671355

Classifier	Select Features	K-value (Relief-F)	Binary/Multi	Un/Pruned	Accuracy
SVM (Kernel: Linear) [Default]	5472 (all)	-	Binary	Unpruned	0.871207
	50	10	Binary	Pruned	0.820873
	125	10	Binary	Pruned	0.843819
	5472 (all)	-	Multi	Unpruned	0.798668
	50	10	Multi	Pruned	0.697261
	125	10	Multi	Pruned	0.726869
SVM (Kernel: Polynomial)	50	10	Binary	Pruned	0.570724
	125	10	Binary	Pruned	NaN
TotalBoost	50	10	Multi	Pruned	0.696429
	125	10	Multi	Pruned	0.717987

Table A.3: Accuracies for all binary and multi-class classifiers, both unpruned and pruned using Relief-F with N = 50 and N = 125.

Classifier	Select Features	K-value (Relief-F)	Binary/Multi	Un/Pruned	Accuracy
SVM (Linear)	All	-	Binary	Unpruned	0.783333
	125	10	Binary	Pruned	0.792857
Random Forest	All	-	Binary	Unpruned	0.783333
	125	10	Binary	Pruned	0.797619

Table A.4: Accuracies for RF and SVM binary classifiers when a 100 Hz signal is used.

Classifier	Select Features	K-value (Relief-F)	Binary/Multi	Un/Pruned	Accuracy
SVM (Linear)	All	-	Binary	Unpruned	0.871207
	125	10	Binary	Pruned	0.843819
Random Forest	All	-	Binary	Unpruned	0.903055
	125	10	Binary	Pruned	0.874907

Table A.5: Accuracies for RF and SVM binary classifiers when a 1 kHz signal is used.

Classifier	Select Features	K-value (Relief-F)	Binary/Multi	Un/Pruned	Accuracy
SVM (Linear)	All	-	Binary	Unpruned	0.707317
	125	10	Binary	Pruned	0.756097
Random Forest	All	-	Binary	Unpruned	0.751219
	125	10	Binary	Pruned	0.768292

Table A.6: Accuracies for RF and SVM binary classifiers when a 20 kHz signal is used.

Classifier	Select Features	K-value (Relief-F)	Binary/Multi	Un/Pruned	Accuracy
SVM (Linear)	All	-	Multi	Unpruned	0.8325
	125	10	Multi	Pruned	0.8375
Random Forest	All	-	Multi	Unpruned	0.8645
	125	10	Multi	Pruned	0.8825

Table A.7: Accuracies for RF and SVM multi-class classifiers in the case that movement is performed during data collection.

Frequency of Tone	Subject Position	# of trials	Label (Binary Label)
1 kHz	No subject	455	0 (0)
	Front (couch-side)	400	1 (1)
	Behind (TV-side)	100	2 (1)
	Right (window-side)	200	3 (1)
	Left (front door-side)	200	4 (1)
20 kHz	No subject	210	0
	Front (couch-side)	200	1
100 Hz	No subject	210	0
	Front (couch-side)	210	1
1 kHz	Moving counterclockwise	100	1
	Moving clockwise	100	2

Table A.8: Summary of number of trials, labels, and positions.