

# CAL: Context Aware Lifestyle Final Report

Michael Bartling  
 michael.bartling15@gmail.com  
 UT Austin ECE CAEP

December 14, 2014

## Abstract

*This paper describes Cal: Context Aware Lifestyle. Cal is a platform for inferring contextual information from sensor aggregates on Android devices. Specifically we propose a novel method for anomaly detection in human motion patterns within a security context. Cal monitors a users motion patterns over a time frame. If the phone happens to change from User A to User B, then within 5 minutes Cal should accurately and reliably notice the difference in users and raise a security context flag.*

## 1. Introduction

Mobile platforms contain a large variety of sensors which can provide contextual information on user's activities. Although inferring user activities from sensor aggregates is not a new concept, previous research on the topic has generally been limited to one to two sensor experiments. Several studies show how accelerometers can be used to infer coarse-grained information such as user motion (walking, standing, sitting, running, etc) and others show how it is possible to achieve even finer granularities such as inferring key-strokes and other touch patterns. Changing perspective of what constitutes a "sensor", modern *virtual sensors* combine multiple hardware sensors to improve sensor readings or create entirely new sensor capabilities. For example, the Android Linear Acceleration Sensor builds upon the accelerometer and is *triggered* whenever a significant change in motion occurs.

Cal, or Context Aware Lifestyle, is a multi-modal sensor aggregation platform that is aware of its users context. The Cal system currently supports inertial indicator contexts, environmental indicator contexts, simple indoor localization contexts, first order user identification contexts, frequency contexts, and temporal contexts. The motion indicators classify cellphone movement as in a *low-motion*, *some-motion*, or *high-motion* state, and the environmental indicator contexts denote the phone *in-pocket* or *out-of-pocket*. Contexts can be combined to form new higher-level contexts much like classes in C++ and Java. Generally, when contexts are combined they begin taking on more abstract meanings such as typical walking behaviour or even anomalous behaviour. This generality may prove useful for a wide range of applications in healthcare, business logistics, crowd-tracking, intelligent no-user inter-

faces<sup>1</sup>, and security (as is the focus of this paper).

To show the effectiveness of the Cal contextually aware model we propose a novel system for identifying anomalies in human motion patterns as a second-order authentication scheme. If Cal determines that whoever holding the phone is a different user, then it will notify the Android system to enter a security context. For example, Android may disable one-click purchases in the app-store, require a password for sensitive emails, or –since the phone may have been stolen–automatically enable the *Find my Phone* feature. Cal itself serves as the first level of authentication, however traditional authentication methods will be used if Cals' prediction does not meet a specified confidence threshold.

## 2. Security Models

### 2.1. Primary Model: Alice and Eve

The primary model adopted throughout this paper is an Alice and Eve security model. Alice is a normal user with sensitive information or services on her phone. Evil Eve is a malicious attacker who wants access to Alice's sensitive data and has physical access to Alices' phone. Eve steals Alices' phone, and within 5 minutes the phone should detect that it has changed from an authorized user to an unauthorized user. Motion analytics are used for first order identification or authentication. If the service cannot classify the user within a known confidence threshold then set a flag for a second order authentication.

### 2.2. Secondary Model: Proof of Work

Another model for the proposed system is proof of work. Consider a logistics setting such as a factory floor or in retail. The branch managers want to make sure their employees are actually working while they are clocked into the system. Or the situation where employees may be dealing with sensitive packages and the managers want to keep track of who is actually handling these packages. Cal can provide simple motion analytics which, when combined, may prove insight into whether an employee is working or texting while on the job.

---

<sup>1</sup>The original goal of the Cal system was to create a smart project management system where users, once enrolled in a project context, do not need to manually save their relevant documents, bookmarks, notes, and emails because Cal would handle it automatically. Ideally, a user could ask Cal to load up all files related to "Final Report", and Cal would do so

### 3. Assumptions

- Android sensor interface maintains *constant* sample rates
- Sensor service communicated is trusted
  - Data from sensor service assumed valid and un-modified
- Attacker cannot directly extract passwords from phone
- Attacker may have access to motion pattern for a phone active windows

### 4. Glossary of Terms

**Note:** The values in parenthesis are what were used for this project.

- **Sample Rate Hz** ( 32 Hz)
- **Bin** Set of N samples (32 = 1 sec worth)
- **Window** Set of k bins (5 min worth)
- **Frame** Set of w Windows (1 Hour worth)
- **Active Window** A window containing at least some percentage of *some-motion* states (10%)

### 5. Method Overview

1. Collect accelerometer data → buffer into bins
2. Classify motion pattern of bin → buffer (data, motion) into window
3. Declare window as active/inactive window → buffer window into frame while keeping track of window activity labels
4. If active window then cluster window according to known user activity patterns → assign user ID
5. If current active window not *similar* to active window(s) in frame within confidence threshold → emit(different user)
  - Inactive windows give little useful information into user fine grain motion patters (i.e. A phone must move from one user to another)

## 6. Related Work

### 6.1. Indoor Localization and Direction of Motion Estimation

When indoors, accurately estimating where a person is and where they are heading or facing poses significant challenges not found outdoors. In this setting, GPS signals are noisy at best and other sensors, such as the magnetometer[12], are subject to interference. Many systems use WIFI direction of arrival estimates [16][2][4], however these systems often suffer from limited precision. Consequently, current research focuses on exploiting combinations of human-local sensors and location sensitive devices. Examples of human-local sensors include dedicated wearable sensors and modern smartphones. Some common location sensitive devices include security cameras[14] and WIFI access points[5]. Finally, some systems include map based information to help with accuracy measurements[7]. I included this last group for completeness, however as [12] points out, map based systems tend to fail in large open rooms and will fail if the floorplans are not in the database.

### 6.2. Geofencing

According to Google,

*Geofencing combines awareness of the user's current location with awareness of nearby features, defined as the user's proximity to locations that may be of interest.*

A geofence is a geographic boundary around an area of interest. Google Now, for example, uses these boundaries to determine nearby shops, events, and news. Furthermore, these boundaries double as privacy boundaries and contextual boundaries [13]. A user may be authorized access to a service or device while inside a given region. This access is revoked as soon as the user leaves the designated zone. The original geofencing paper [13] used WIFI access points to designate service boundaries, however this requirement has been relaxed over time. Most smartphones today primarily use GPS for geofencing. Android extends geofencing concepts to a local network using its Network Service Discovery (NSD) system. NSD searches for other instances of a particular service on a local subnet. This could be printers, network projectors, or even users playing a common video game.

### 6.3. Activity Recognition

Much research has gone into analysing and estimating human activities from sensor information[3][8]. In [8], the authors showed it is actually possible to identify a person by their gait given only accelerometer data. Of course, they assumed static positioning of the phone (it was placed on a belt clip) and admitted to variations in the shoes someone wears, the surface they walk on, and their mental state all have an impact on the final identification. Furthermore, the proposed system required complex training sets which makes it impractical for large scales. However, the idea that a sensor information can give insight into a users activities, and these activities can be used as a second degree of phone security has lasting importance both in this project and others.

Other research has taken the reverse roll, and attempted to violate security via sensor information. For example, TapLogger[15] is a keylogger based on smart phone accelerometer side channels. And there have been many, sometimes successful, attempts to infer user passwords from accelerometer data[1][11][10]. Logically, there must be some connection between user-smartphone interactions and a security context. Furthermore, a users activities contain important information depending on a relevant context. This is a very important concept which I will further discuss later.

### 6.4. Frequency and Temporal Analysis

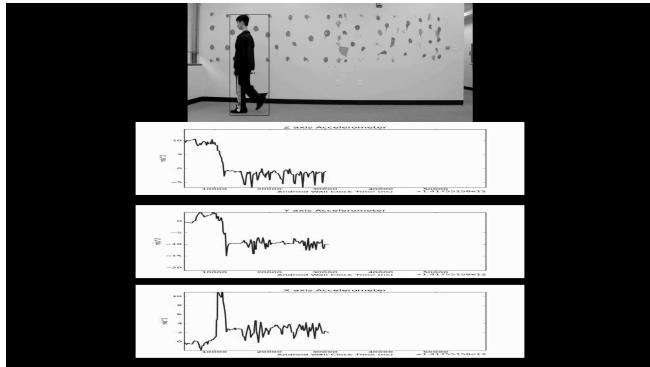
Frequency Analysis and Temporal analysis is more closely related to data mining techniques[6][9], however some concepts are relevant to this paper and will be discussed. Google Now, for example, uses a simple counting metric combined with temporal information and spatial information to infer a users home location and work location. This makes sense. Supposed

somebody travels to the same general location every day for somewhere between 6 to 9 hours, but spends the night at a different location. Given typical human behaviour, the prior location is likely to be marked *work* and the latter location is likely marked *home*. As with any scientific hypothesis, this model must be verified under repeated trials. Say for a given person this model works for 49 out of 70 trials. Empirical analysis shows that this person does not work on weekends. Factoring that into the model, on 49 out of 50 week days the person in question was in one of two general locations for the better part of the day. If we tilt our perspective a little, we get a whole new perspective. In the previous example, 49 out of 50 work days fit our model. In other words, 1 out of 50 did not fit our model. This give insight into anomaly analysis, or an understanding of things that differ from expectation.

## 7. The Human Walk Cycle

In [8], the authors decomposed the human walk cycle into *heel strikes*, *toe-strikes*, and *sway*. We verified these findings by registering a video sequence to accelerometer readings. The video was processed in OpenCV with the original goal of mapping motion states to video sequences. However, we found these videos gave us deep insight into nuances in human walk cycles which account for some of the results in later sections.

In the accelerometer context, high intensity signal tend to represent heel and toe strikes; these account for most of the middle to high frequency components in the waveforms. The orientation of the phone in pocket (or in hand) primarily contributes to the DC offset and very low frequency components, while lateral sway during motion, between heel/toe strikes, contributed to low to middle frequency tones. Figure 1 shows an example video trial registered to accelerometer data.



**Figure 1: Screenshot of example Video + tracking + accelerometer registration.**

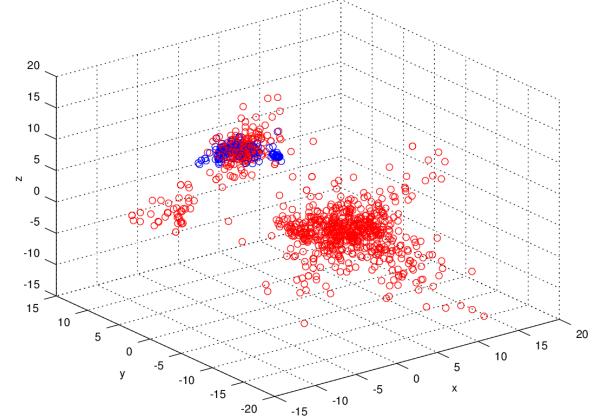
The human tracking method uses an extended Top-hat filter to account for illumination issues plus a background subtracter method for motion capture. In this video, the subjects were asked to press a button to start motion capture, put their phone in their pocket, and then walk across the room. Transferring

the phone to the pocket is evident in the first part of the X and Y acceleration waveforms where the signal drops to a new DC value (i.e. the phone changes orientation).

## 8. Identifying the Motion States in the Human Walk Cycle

we began by exploring different features using a time windowed binning method. Basically, by changing the window (bin) size we can control the resolution of our feature set. For example, a window size of 16 and 32 preserves most of the high frequency components of the signal. This is shown in Figure 13. These high frequency components directly translate to motion relative to the device. The bin sizes were limited to powers of two for efficient FFT computation. We chose a bin size of 32 samples, since it provides reasonable frequency resolution and is conveniently about 1 second of samples.

A simple scatter plot of XY, YZ, and XZ pairs, such as shown in Figure 2 and Figure 15, shows the accelerometer readings approximately follow a Gaussian Normal distribution. This result is verified in the other two graphs (not shown) and implies that I can use Gaussian Mixtures to classify the data. Gaussian Mixtures are an unsupervised learning algorithm based on Expectation Maximization methods that try to classify data into normally distributed clusters. Running the algorithm directly on unprocessed accelerometer data works until the phone is tilted. Also, the results were rather noisy so a pre-processing step is necessary.



**Figure 2: 3D Scatter plot of X,Y,Z accelerometer tuples follow Gaussian Normal Distributions. Red denotes motion states, blue is low motion state.**

The preprocessing step removes the DC component, per Bin, of the signal<sup>2</sup> which is a clean way of normalizing all components of the signal. You can see the results of this normalization in Figure 16 where the top signal shows the original waveform and the second graph is the same waveform

<sup>2</sup>This simply means I set the Zeroth component to 0

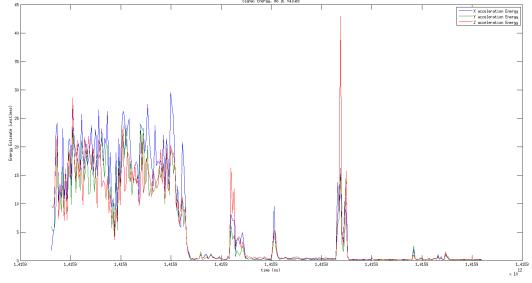
without the DC component. Removing the DC component effectively remove the orientation of the phone. This makes fitting the data far simpler. A simple low pass filter helps reduce ringing between bins.

Next, we ran some basic statistics on each bin before sending them to the Gaussian Mixture Fitting algorithm. Each statistic corresponds to a different feature set. For example, the feature set in the below figures is the *signal energy* without the DC component. This equates to  $SUM(X_{FFT}^H) = SUM(X_{FFT} \cdot X_{FFT}^{*T})$ . Since the input signal is solely real, its FFT is Hermitian Symmetric. This vastly simplifies the energy equation. We also supply the GMM model with the median of the Bin signal; this accounts for salt and pepper noise and adds a little bit of variance to the feature sets. There are some other feature sets such as window max, window min, histogram max, variance, and mean, which can also give valid results (not shown). For all of the feature sets, an Android phone should be able to compute the feature with no visible latency to the user. Some results of running a GMM clustering without the median feature set on 140k/32 data points is shown in Figure 16. The bottom 3 graphs correspond to the posterior probability that the cluster output equals  $N$  for each  $N$  in the cluster. I.e. the bottom graph corresponds to the cluster values of 3, which means *idle* in laymans terms. Figures 17 18 19 show example predictions and energy estimation with the inclusion of the median feature set. Unfortunately, the GMM method does not allow ordering of the clusters in any manner, so we must post process to actually assign proper labels. This is done by sorting the training window of (Energy, prediction) tuples and assigning the low-motion label to the lowest energy set, and high-motion to the highest energy set. This allows remapping from the original prediction to the proposed motion state labels.

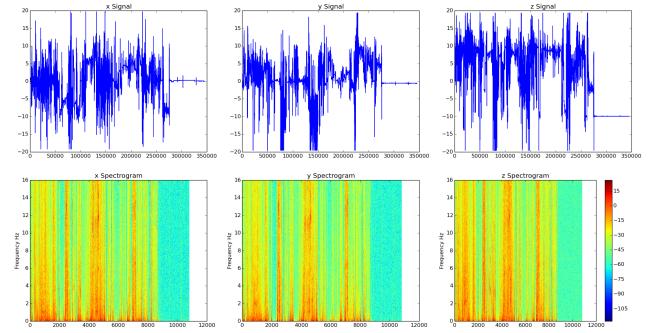
Since GMMs are unsupervised, characterizing accuracy of predictions is difficult. However, visual inspection of approximately 4 million data points for 12 users shows conservative average prediction accuracy  $> 95\%$ . Mis-predictions generally occur at the boundary between motion states because a Bin may be temporally skewed towards one motion state. In this case, the prediction miss impacts precision more than accuracy. The GMM pre-processing and post-processing methods described are implemented in OpenCV, SciPy (Python 2.7), and C++. All data sets, including predictions, are maintained on a MySQL database. Additionally, WiFi state information (RSSI, BSSID, SSID) is gathered for localization.

## 9. Distinguishing User Motion Patterns

Figures 4 5 6 7 show the Spectrograms of 4 different users over a 3 hour period as well as the corresponding Raw Accelerometer signals. Each vertical line represents a Bin of samples, the darker regions contain high motion states and the lighter regions denote low motion states. Each figure shows distinct horizontal and vertical banding patterns. User 2 has a characteristic hue apparent in the low motion regions. User



**Figure 3: Signal Energy plot of states walking, not walking and phone on desk. States roughly translate to High motion, some motion, little motion.**

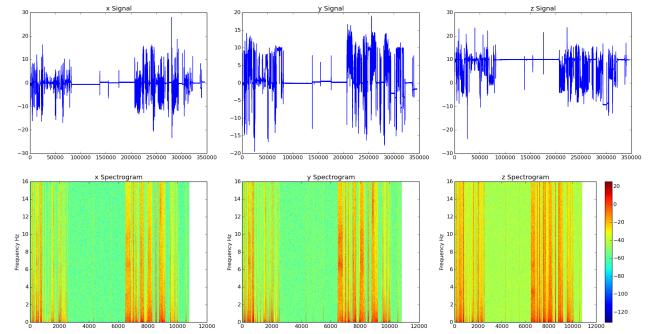


**Figure 4: 3 Hr Spectrogram User 1**

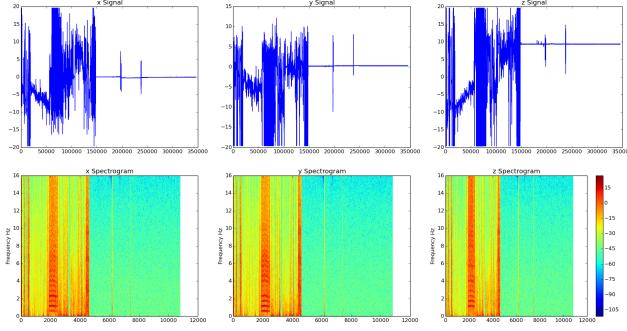
3 has visible banding between high-motion bins and a slight gradient in low-motion regions. Clearly, individual users have characteristic motion patterns which are distinguishable from other users patterns.

### 9.1. Naive Approach: Direct SVM

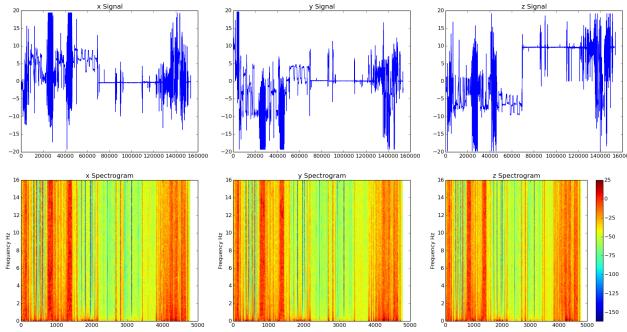
As a baseline, we supply an SVM with raw accelerometer signals for 9 different users classified according to a UUID. Inactive windows are defined for windows with more than 90% low-motion states. Inactive windows are ignored in the analysis because a phone must undergo motion to move from one user to another. Given the high dimensionality of the signals, highly correlated X,Y,Z signals, and low feature SNR,



**Figure 5: 3 Hr Spectrogram User 2**

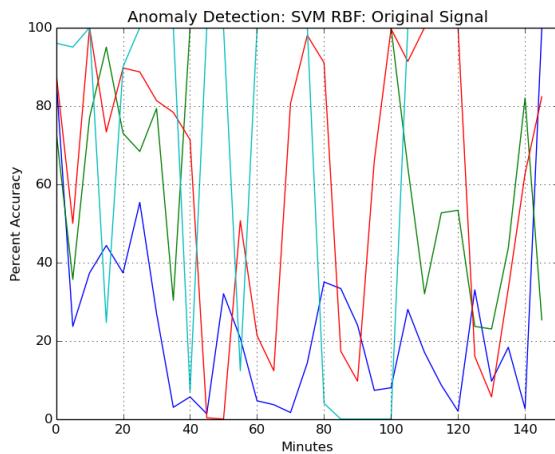


**Figure 6: 3 Hr Spectrogram User 3: Note the banding in the high energy regions**



**Figure 7: 3 Hr Spectrogram User 4**

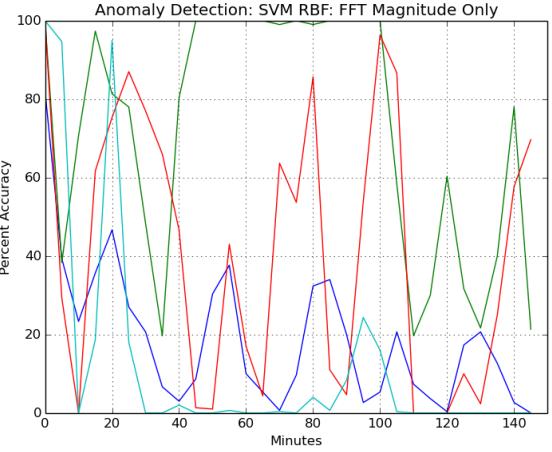
we expect the SVM to perform poorly. The Type-C SVM was configured for an RBF kernel and automatic optimum-coefficient calculations. Initial training took approximately 180 seconds to complete, and the accuracy for 4 of the users is shown in Figure 8. Clearly, the results met expectations.



**Figure 8: Accuracy of Naive SVM Anomaly Classification on Original Signals**

To account for phone orientation noise (low frequency) and

possible phone rattle (high frequency) a simple notch filter was applied to the signals. The Phase of the signal was discarded to account for position/temporal dependent signals and the results are shown in Figure 9. The results are highly similar to the original signal. Despite tweaking the SVM parameters, the classification will still behave similarly. This implies that we need a phase-sensitive feature set, and that the feature set must be compressed.



**Figure 9: Accuracy of Naive SVM Anomaly Classification on Position Invariant Signals**

## 9.2. Compressing the Feature Space

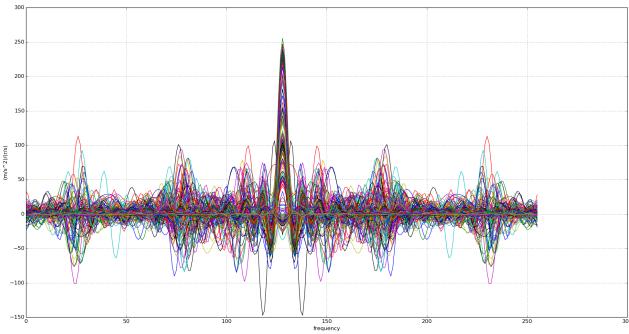
Compressing the signals into key components is crucial for the classification methods to work. So far, we have only considered a single Bin of samples for classification. Figure 10 shows a 2D representation of typical Accelerometer frequency responses for a given active window and Figure 11 shows the same response in 3D. Notice in Figure 10 that there are clear peaks and valleys in the intensity values. These locations are somewhat unique to a user, with minor variations in offsets due to changes in walking pace. Figure 11 indicates that the signal contains some sort of inertia in that it does not, generally, change instantaneously. However, there is instantaneous behaviour at the boundary of different motion states (low-motion, some-motion, high-motion). Furthermore, both figures show that the signal is band-limited.

Partial Energy is Defined as:

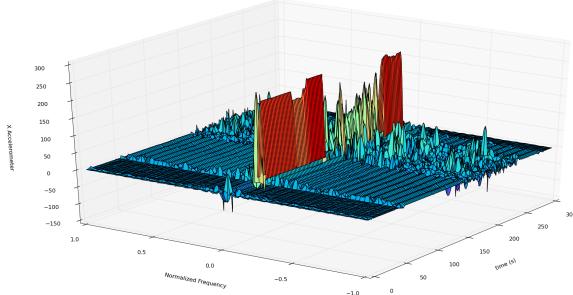
$$E[n] = \sum_{k=0}^n |h[k]|^2 = \sum_{k=0}^n H[k]H^*[k]$$

So for the  $i$ th bandpass filter  $h_i[k]$  whose corresponding representation in the frequency domain is  $H_i[k]$  in the filter bank

$$G_i[k] = H_i[k] \sum_{n=0}^{N-1} x[n]W_N^k n, k = 0, 1, \dots, N-1; G_i[k] = H_i[k]X[k]$$



**Figure 10: Plot of X Accelerometer Data for single Time Window**



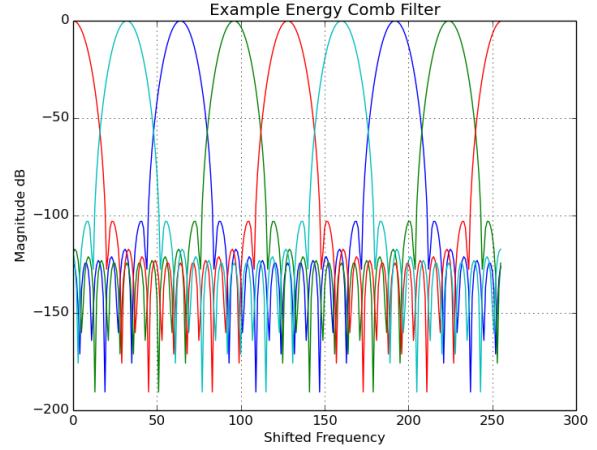
**Figure 11: 3D Plot of X Accelerometer Data for single Time Window**

$$E_i[n] = \sum_{k=0}^n |g_i[k]|^2 = \sum_{k=0}^n G_i[k]G_i^*[k]$$

1. Since the Energy signatures of the accelerometer signals generally do not change abruptly and 2. Since these signals exhibit band-limited behaviour, we can compress the  $k$ -dimensional signal into a  $i$ -dimensional feature set, where  $k$  is the width of the Fourier transform and  $i$  is the number of Bandpass filters in the filter bank. A regression model can then be applied to this  $k$ -dimensional feature set to predict the next motion states which acts as a lowpass filter between bins/frames. Furthermore, this prediction model can be validated via the GMM motion state estimators. In parallel, a Classifier algorithm will attempt to map this feature set plus the previous prediction to a typical pattern in the current frame. In other words, the system will keep track of the change in motion behaviour between bins, and between windows, relative to the current frame (2 levels of granularity). This should help minimize false positives in the system (where Cal falsely classifies a valid user as a different person). Ultimately, the two classifications available are {Same Person, Different Person}.

Another option for compressing the feature space involves Haar Wavelets and Boosting. Here the wavelets decompose the signal into compressed principle components while maintaining temporal/spatial information (i.e. heel strikes followed

quickly by a toe strike followed by a long sway). Individually, these wavelets can be thought of as weak classifiers for the system. Boosting takes a number of these weak classifiers and tries to form a strong classifier that is faster, and with higher accuracy, than traditional strong classifier models (SVM, GMM, K-means). This method is predominately used for template matching methods such as face tracking in computer vision due to its robustness and high performance. This should be explored as an option for the Cal system in future research.



**Figure 12: Example independent Comb Filter(s) for Decomposing Bins into Partial Energy Feature Sets.**

## 10. Attacks on Cal

Currently, Cal is highly susceptible to DoS attacks based on sample rate inconsistencies. Cal assumes approximately 32 Hz sample rate for timing reasons, and Cal would probably function as-is up to 60Hz. However, none of the algorithms presented nor the Cal system account for large changes Android sample rate (which normally just varies slightly). If someone were to interfere with the sensor sample rate such that it changes to some random value every few milliseconds then Cal would stop working. Fortunately, setting the sample rate too high will not break Cal since we can easily add a subsampling pre-processing step or some other rate matching method.

Cal trusts that Android is implemented correctly, in addition to the sensor service. If someone were to inject a false virtual-sensor in place of the android provided interface then an attacker can effectively bypass the second level authentication on any phone running Cal. However, this virtual sensor would need to supply a realistic signal to the Cal service because low-motion states do not trigger the auto-authentication methods. In fact, the attackers cannot simply supply Gaussian noise since this will be minimized in the GMM preprocessing steps, and caught in the post processing steps. Furthermore, we can use other algorithms to try to catch synthetic motion

models.

## 11. Conclusion

Although we discussed context aware sensing in a security context, our proposed system can be used to create contextual boundaries for a wide range of applications. Furthermore, there is some underlying algebra associated with contextual concepts. For example,

*Let  $a = \text{Accelerometer readings} \in \{\text{motion}_{\text{low}}\}^c$*

$\rightarrow \text{accelerometer readings} \in \{\text{motion}_{\text{some}}, \text{motion}_{\text{high}}\}$

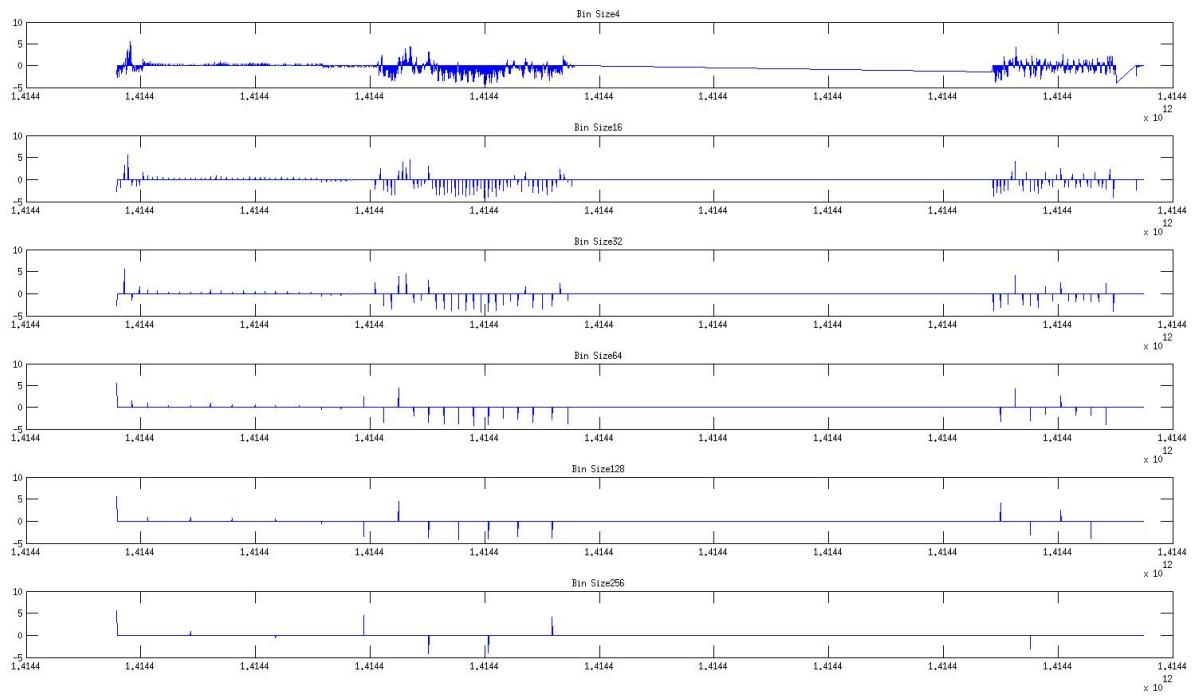
*Let  $b$  denote event GPS lock indicates user entered a building  $b$*

*Let  $w = \{\text{Wi-Fi signal indicators given Room } r\} \text{ for } r \in b$ .*

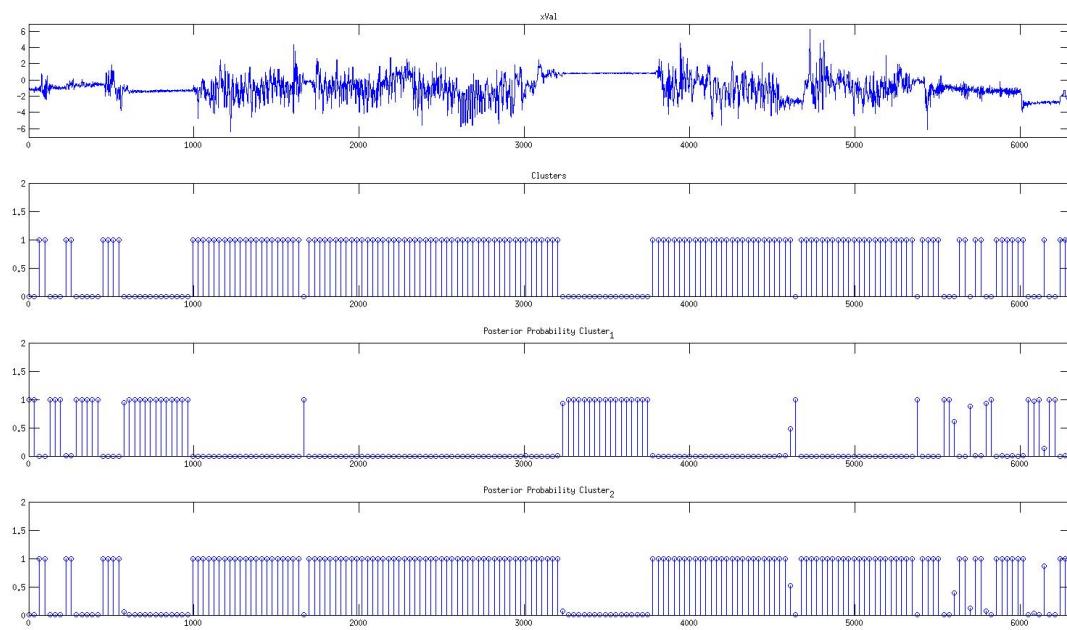
Then we can calculate the probability a person is in a particular room given they have **moved** within the building relative to known access points.

## References

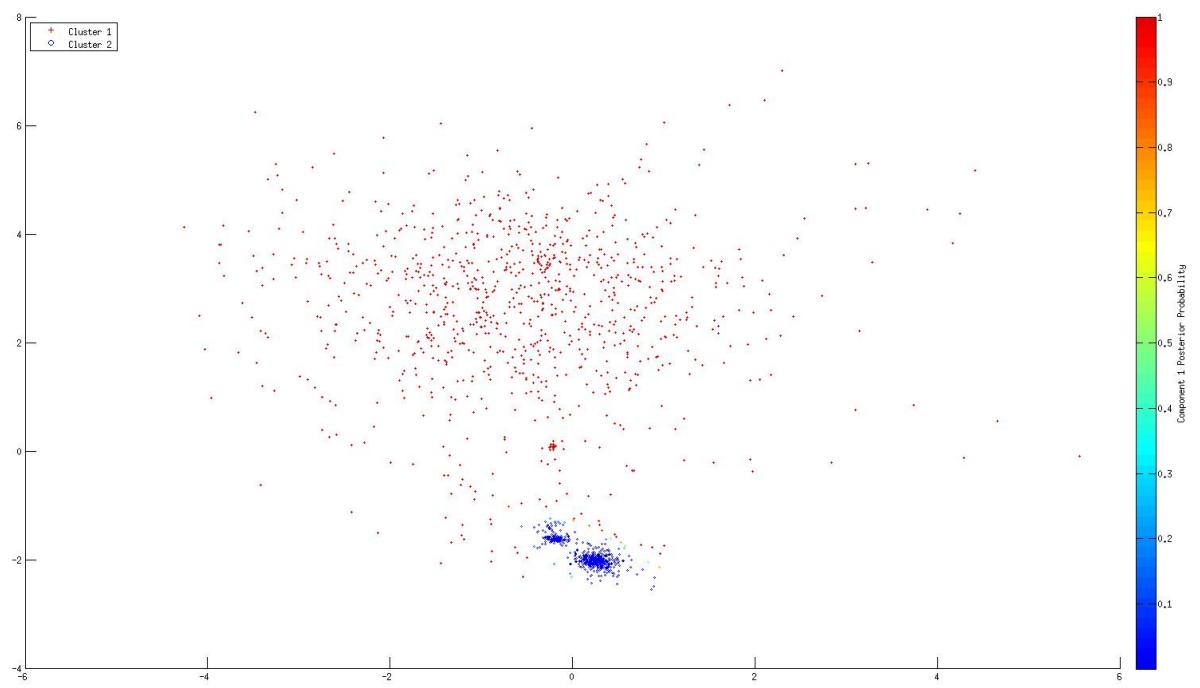
- [1] A. J. Aviv, B. Sapp, M. Blaze, and J. M. Smith, "Practicality of accelerometer side channels on smartphones," in *Proceedings of the 28th Annual Computer Security Applications Conference*. ACM, 2012, pp. 41–50.
- [2] P. Bahl and V. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, 2000, pp. 775–784 vol.2.
- [3] Y.-H. Chen, "Ada: Context-sensitive context-sensing on mobile devices," Ph.D. dissertation, MIT, 2013.
- [4] R. K. Harle and A. Hopper, "Deploying and evaluating a location-aware system," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '05. New York, NY, USA: ACM, 2005, pp. 219–232. Available: <http://doi.acm.org/10.1145/1067170.1067194>
- [5] E. H. J. Krum, "Locadio: "inferring motion and location from wi-fi signal strengths"," in *IEEE Computer Society*, 2004.
- [6] S. Laxman and P. S. Sastry, "A survey of temporal data mining," *Sadhana*, vol. 31, no. 2, pp. 173–198, 2006.
- [7] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, ser. UbiComp '12. New York, NY, USA: ACM, 2012, pp. 421–430. Available: <http://doi.acm.org/10.1145/2370216.2370280>
- [8] J. Mantyjarvi, M. Lindholm, E. Vildjounaite, S. marja Makela, and H. Ailisto, "Ailisto, identifying users of portable devices from gait pattern with accelerometers," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005.
- [9] H. J. Miller and J. Han, *Geographic data mining and knowledge discovery*. CRC Press, 2009.
- [10] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury, "Tappprints: your finger taps have fingerprints," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 323–336.
- [11] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "Accessory: password inference using accelerometers on smartphones," in *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*. ACM, 2012, p. 9.
- [12] N. Roy, H. Wang, and R. Roy Choudhury, "I am a smartphone and i can tell my user's walking direction," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '14. New York, NY, USA: ACM, 2014, pp. 329–342. Available: <http://doi.acm.org/10.1145/2594368.2594392>
- [13] A. Sheth, S. Seshan, and D. Wetherall, "Geo-fencing: Confining wi-fi coverage to physical boundaries," in *Pervasive Computing*. Springer, 2009, pp. 274–290.
- [14] T. Teixeira, D. Jung, G. Dublon, and A. Savvides, "Pem-id: Identifying people by gait-matching using cameras and wearable accelerometers," in *Distributed Smart Cameras, 2009. ICDS 2009. Third ACM/IEEE International Conference on*. IEEE, 2009, pp. 1–8.
- [15] Z. Xu, K. Bai, and S. Zhu, "Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors," in *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*. ACM, 2012, pp. 113–124.
- [16] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '05. New York, NY, USA: ACM, 2005, pp. 205–218. Available: <http://doi.acm.org/10.1145/1067170.1067193>



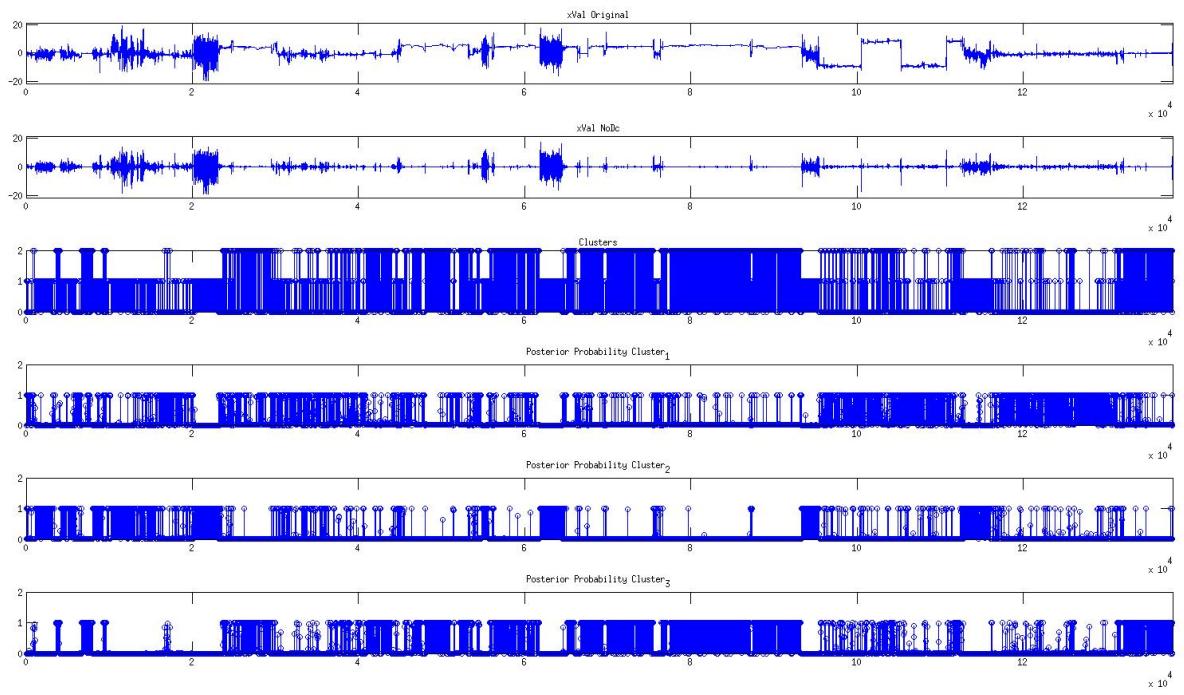
**Figure 13: Experimenting with different bin sizes to determine optimal width.**



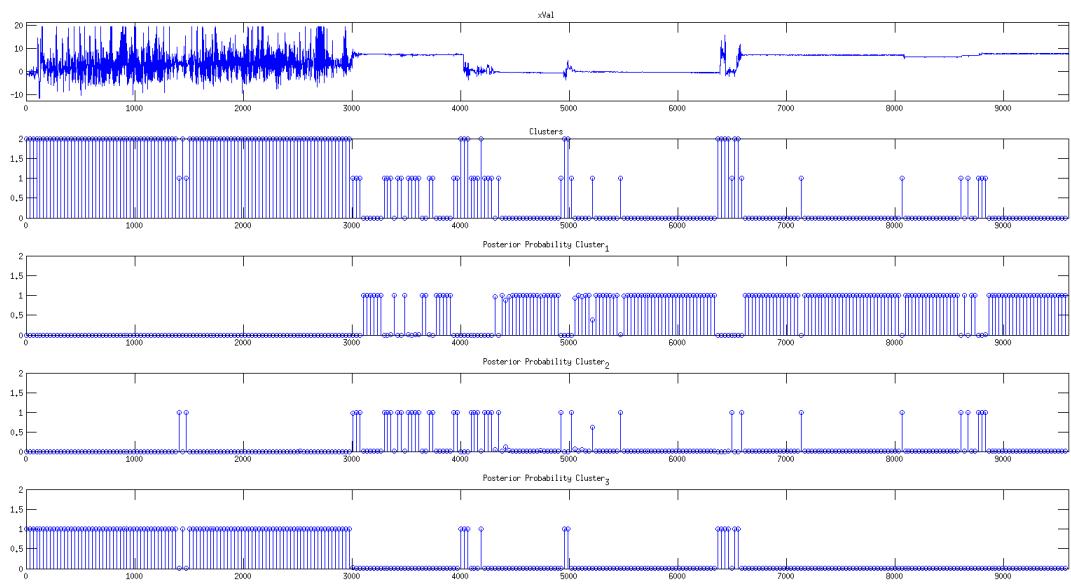
**Figure 14: Clustering output with 32 sample bin and 2 features**



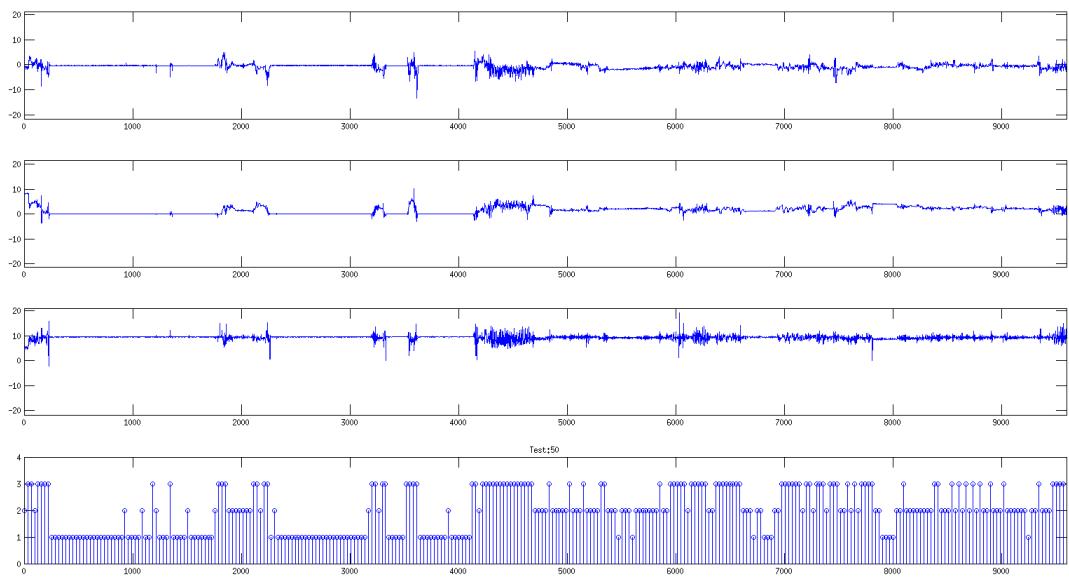
**Figure 15: Same Clustering with XY scattered and Probabilities Assigned**



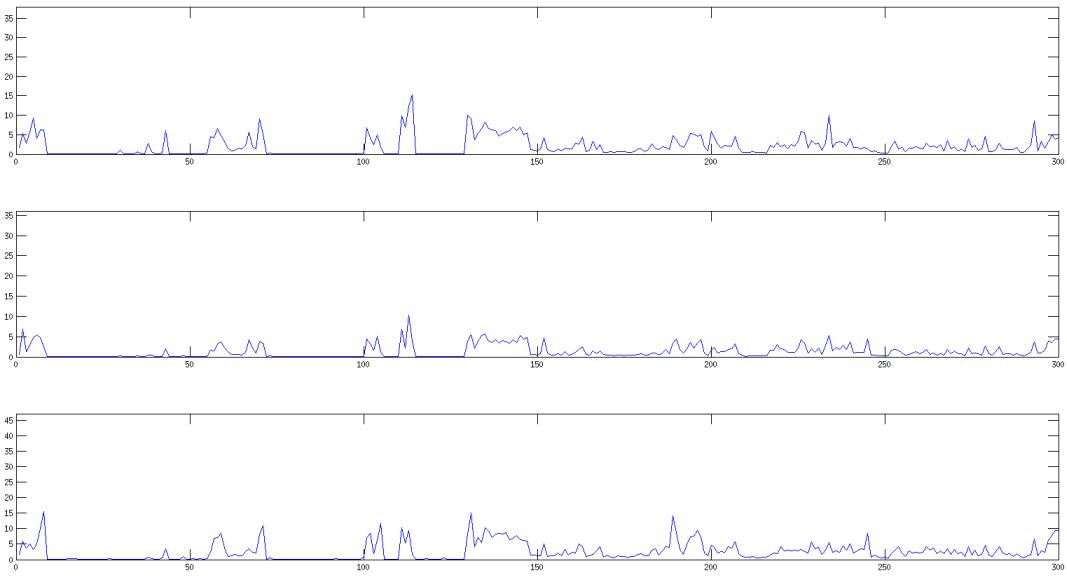
**Figure 16: Extending the number of clusters and data set.**  
**Note, the second graph is the same as the first graph with DC component removed. Bin Size is 32. This signal was trained with roughly 10k/32 data points from a previous signal (not shown). These predictions do not contain the median feature set**



**Figure 17: GMM Prediction with inclusion of median feature set.**



**Figure 18: GMM Prediction with inclusion of median feature set. Waveforms shown with DC Component removed**



**Figure 19: Energy Signatures for same waveforms used in GMM predictions with median feature set. Note, DC components removed.**