**FingClr: Surface EMG Time-Frequency Statistical Feature Extraction and Classification with both a Support Vector Machine (SVM) and a Recurrent Neural Network (RNN)**

By Jacob Kennedy and Connor Maynes

**1. ABSTRACT**

The purpose of these experiments was to collect surface electromyographic ( sEMG ) signals from a male subject, apply simple time-frequency domain feature extraction techniques ( mean, max, variance and root mean squared ( RMS ) ), and pass the resulting datasets through a recurrent neural network ( RNN ) and a support vector machine ( SVM ) and compare the results of both classification techniques. The RNN achieved a testing accuracy of 91.59%, while the SVM had a testing accuracy of 98.32%, on average, after building and training both classifiers ten times and collectively calculating the accuracies. Both classifiers were also trained and tested on a third-party dataset from another research paper that achieved a 94.80% accuracy rate using Empirical Mode Decomposition (EMD) for feature extraction, and then a simple linear classifier [32]. while the RNN achieved a testing classification accuracy of 95.80% and the SVM an accuracy of 98.53%.

**2. INTRODUCTION**

The purpose of this experiment was to first collect surface electromyography ( sEMG ) signals from skeletal muscles on the forearm of the subject using the CleveMed BioRadio 150 - hereinafter referred to as BioRadio - and second to use the collected data to classify the action as either pinky flex, ring finger flex, middle finger flex, index finger flex, thumb flex, wrist turn, elbow flex, or rest. Since sEMG signals are usually in the range of 0 to 400 Hz, a sampling rate of 960 Hz was used in order to avoid aliasing of the signal. Each sample consisted of 2640 timesteps lasting 2.75 seconds. Each of the seven action classes was performed a total of 20 times each and stored in separate files, while 117 samples were recorded for the rest class continuously and split into separate files after collection had completed.

A combination of features extracted from the time-domain and frequency-domain signals was used to train the RNN and SVM classifiers. The mean, max, variance and root mean squared ( RMS ) were extracted from each channel of both the time-domain and frequency-domain versions of the data for each sample. The features collected from the time-domain data were collected from raw, unfiltered, data. The features collected from the frequency-domain data were collected from data that was passed through a Butterworth filter to remove the 60 Hz - from the lights - and 0 Hz - the DC component - signals from the samples. The extracted features from the raw samples were oversampled before being fed into the classifiers, so that each class had a total of 117 samples for each class, giving each class an equal a priori probability.

Two different classification methods - a recurrent neural network ( RNN ) with 10 layers, which was called an LSTM ( long short-term memory ) in MatLab, and a decaying learning rate and a support vector machine ( SVM ) with a cubic polynomial kernel function were used. All other options for both of these classifiers were left as their defaults in MatLab. Both the SVM and RNN achieved their expected accuracies of between 90-95%.

All of the feature extraction and classification techniques were also applied to a third-party dataset from another research paper which achieved an accuracy of 94.8% using empirical mode decomposition (EMD) for feature extraction and a simple linear classifier [32]. The third-party dataset was a collection of sEGM signals collected at a 500 Hz sampling rate for six different hand gestures and with most of their noise coming from a 50 Hz signal from nearby powerlines. The validation set contained 30 samples for each of the 6 classes. The FFT was specifically configured for this third-party dataset to filter data between 1 and 250 Hz - instead of the 1 Hz to 479 Hz range used on the data collected for these experiments - and to filter out the 50 Hz noisy signal rather than 60 Hz noisy signal. The validation data was collected on multiple subjects, but for the purposes of validating the design of these classifiers, only the data named **male 2** was preprocessed and classified by both the RNN and SVM to test their functionality.

**3. LITERATURE SURVEY**

**3.1 GENERAL STATEMENTS**

Rapid classification of sEMG signals could allow more accurate and functional prosthetics that require a minimal amount of training. The design of machine learning systems is challenging and often progress is only made through trial and error of many different designs, but with each iteration the progress of the entire scientific community steps forward.

An NN or SVM for classifying 4 to 12 actions should have an accuracy of about 90-95% [10]. Recent work with NNs combining the techniques of using a 300 ms timeframe, pruning, and a hyperbolic tangent function for the action function in the neural network have proven highly effective at increasing classification accuracy [11-15]. SVMs have been shown to outperform multi-layer perceptrons ( MLPs ) [18]. SVMS using methods of multiple kernel learning ( MKL ), the one-against-one method, and linear kernels have been shown to produce accuracies ranging from 90-95% and to outperform ANNs and linear discriminant analysis ( LDA ) [21-23, 27]. For the aforementioned reasons, both NNs and SVMs were implemented as classifiers during these experiments in combination with some simplistic feature extraction techniques in the time and frequency domains.

**3.2 FEATURE EXTRACTION - FFT**

Fast Fourier Transform ( FFT ) is an analytical method that extrapolates frequency components from a time domain-based signal. FFT can often be used in filters, such as the Butterworth filter used for these experiments, to remove noisy signals present across all signals as background noise that does not contribute any discriminating information between the signals of different classes. FFT is an extremely useful tool especially when dealing with biopotentials since the biopotentials are often clustered and hard to differentiate in the time domain [29-31].

**3.3 CLASSIFICATION - RNN & SVM**

pattern recognition has shown promising results in recent years for classifying hand movements from accelerometer data; however, these methods do not provide robust natural control [1-5]. A recent study used a NN consisting of an input layer, four convolutional layers, four subsampling layers, and two fully-connected layers, both with a trained and untrained classifier. The results indicated better accuracy from the NN versus the SVM and that that network design was somewhat user-adaptive [6]. A user-adaptive design reduces training time for each user by generalizing part of the training results, so they can be used across different users.

The accuracy of any surface electromyography ( sEMG ) classification system is heavily dependent on the number of actions being classified with accuracy levels of 90-95% for 4 to 12 actions and 60-70% for 50 or more actions [10]. Before the data can be passed through the preprocessing and feature extraction sections of a classification system, an appropriate window size for the time-series data must be selected. Numerous time-windows have been attempted, often ranging from 100 to 125 ms or 280 to 300 ms [13-15]. The first time-window range of 100 to 125 ms results in lower classification latency, but often at the cost of accuracy, while the inverse is true of the second range. An additional method of affecting the efficiency of a NN is through pruning. A pruning approach, whereby the network is trained, unimportant connections are pruned, and then the network is retrained, has proven to dramatically reduce the number of parameters in the network by as much as a factor of ten [11]. The activation function is also important in the efficiency of a NN. The hyperbolic tangent function converges faster than the sigmoid or the the rectified linear unit ( ReLU ) and thus reduces training time [12].

Multi-layer perceptrons ( MLPs ) have been compared against the performance of SVMs and shown to be outperformed by SVMs [18]. Known problems relating to the use of SVMs are the memory requirements of the computer used to train them and the time-cost of training them [19-20]. A combination of multiple kernel learning ( MKL ) and SVM has been attempted and resulted in accuracy above 95% [21]. A linear kernel tends to produce higher accuracy over other kernels in some cases [22]. An SVM has been shown to provide better classification accuracy compared to ANNs and linear discriminant analysis ( LDA ) [23]. Another known problem with SVMs is that classification performance degrades over time by as much as 14.6%, when the SVM is not recurrently trained [24]. Time-frequency domain features, when combined with an SVM, provide superior accuracy over SVMs fed with only time domain features [25]. An approach called incremental online SVM, whereby only the samples closest to the current boundary are used to train the classifier, resulted in a dramatic improvement in accuracy over traditional SVM [26]. The one-against-one method has proven effective when working with high-dimensional vector spaces and SVMs [27]. In the one-against-one method, a binary classifier is created for each pair of classes using SVM. All of the binary classifiers are executed and then the class with the highest number of votes is selected.

NNs and SVMs have consistently shown higher classification accuracy over many alternatives independently. RNNs and SVMs should provide comparable results of 90-95% for the seven action classes and one rest class that were classified for these experiments [28].

**4. PREPROCESSING & LEARNING**

**4.1 DATA COLLECTION**

The BioRadio 150 was used in combination with the BioCapture software to collect the sEMG signals from the subject across six different channels. A male subject was used for data collection. The electrodes were placed on the subject as follows in Figures 1 and 2. Due to a technical issue with the BioRadio, channel 3 did not work, so channels 1,2,4,5,6, and 7 were used.



Figure 1: Placement of Electrodes for Channels 5,6 and Ground



Figure 2: Placement of Channels 1,2,4, and 7

The seven actions - as well as rest - performed by the subject were recorded and represented in Figures 3 through 8.

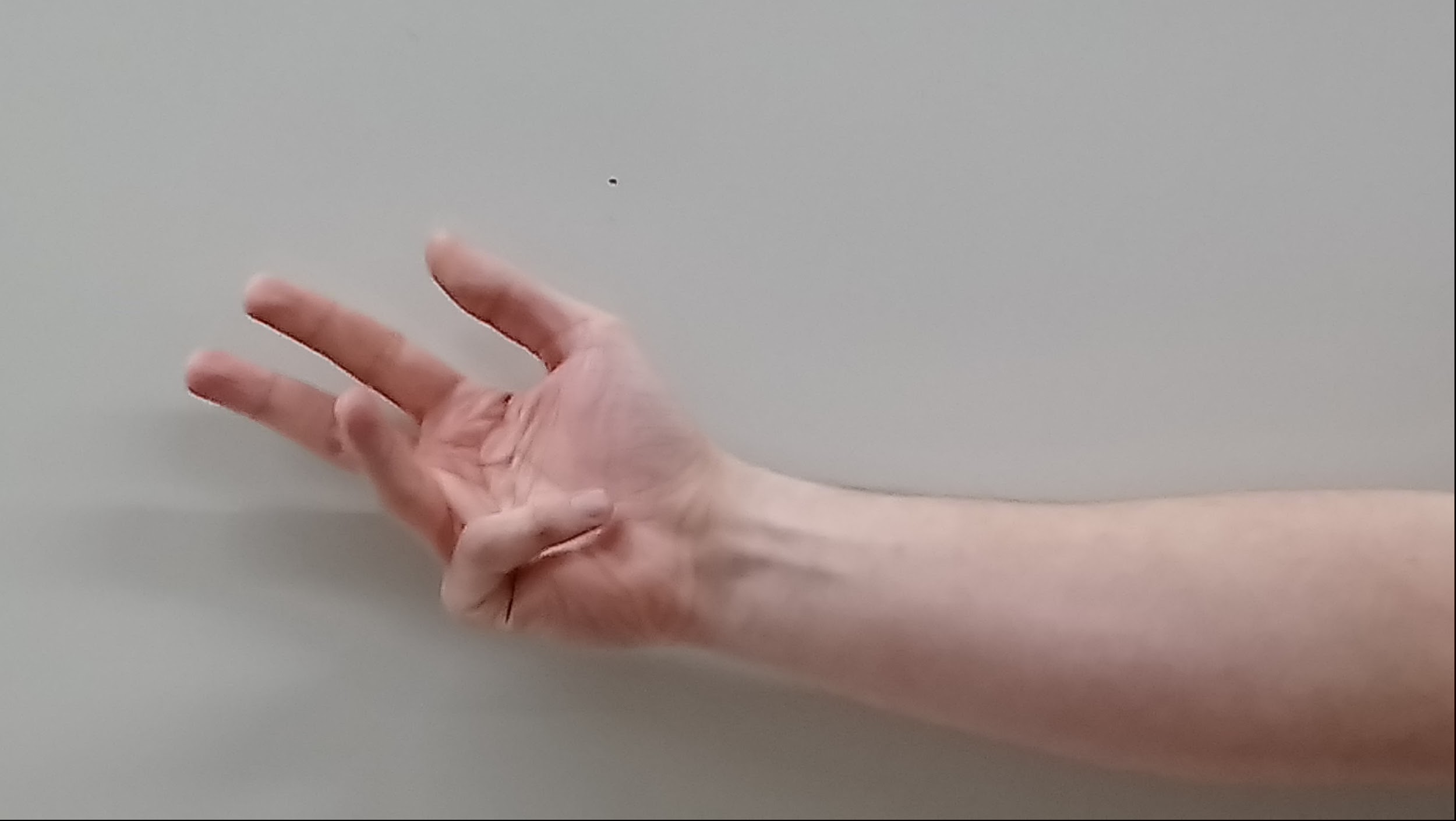


Figure 3: Pinky Flex Action

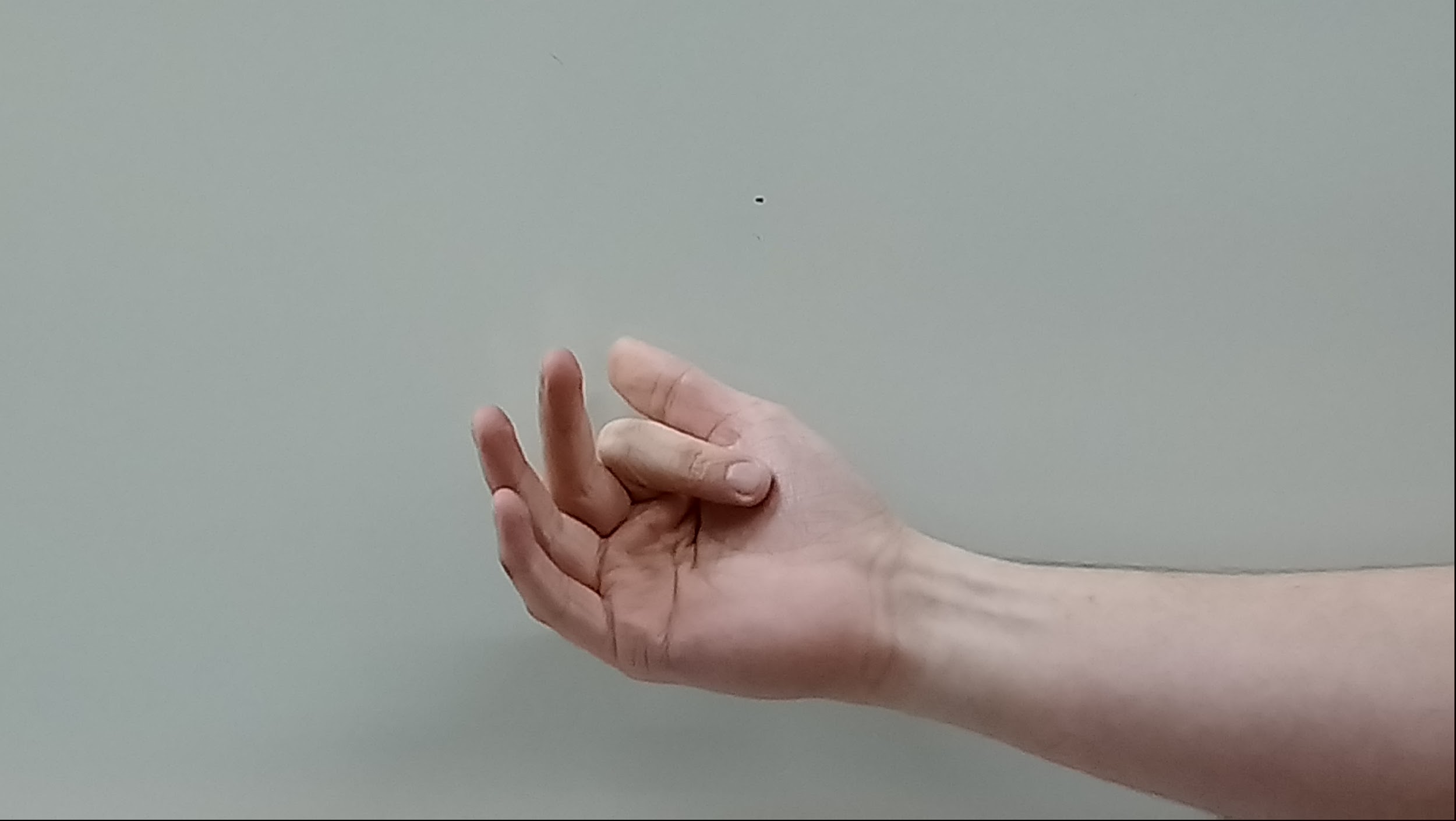


Figure 4: Index Finger Flex Action



Figure 5: Middle Finger Flex



Figure 6: Ring Finger Flex

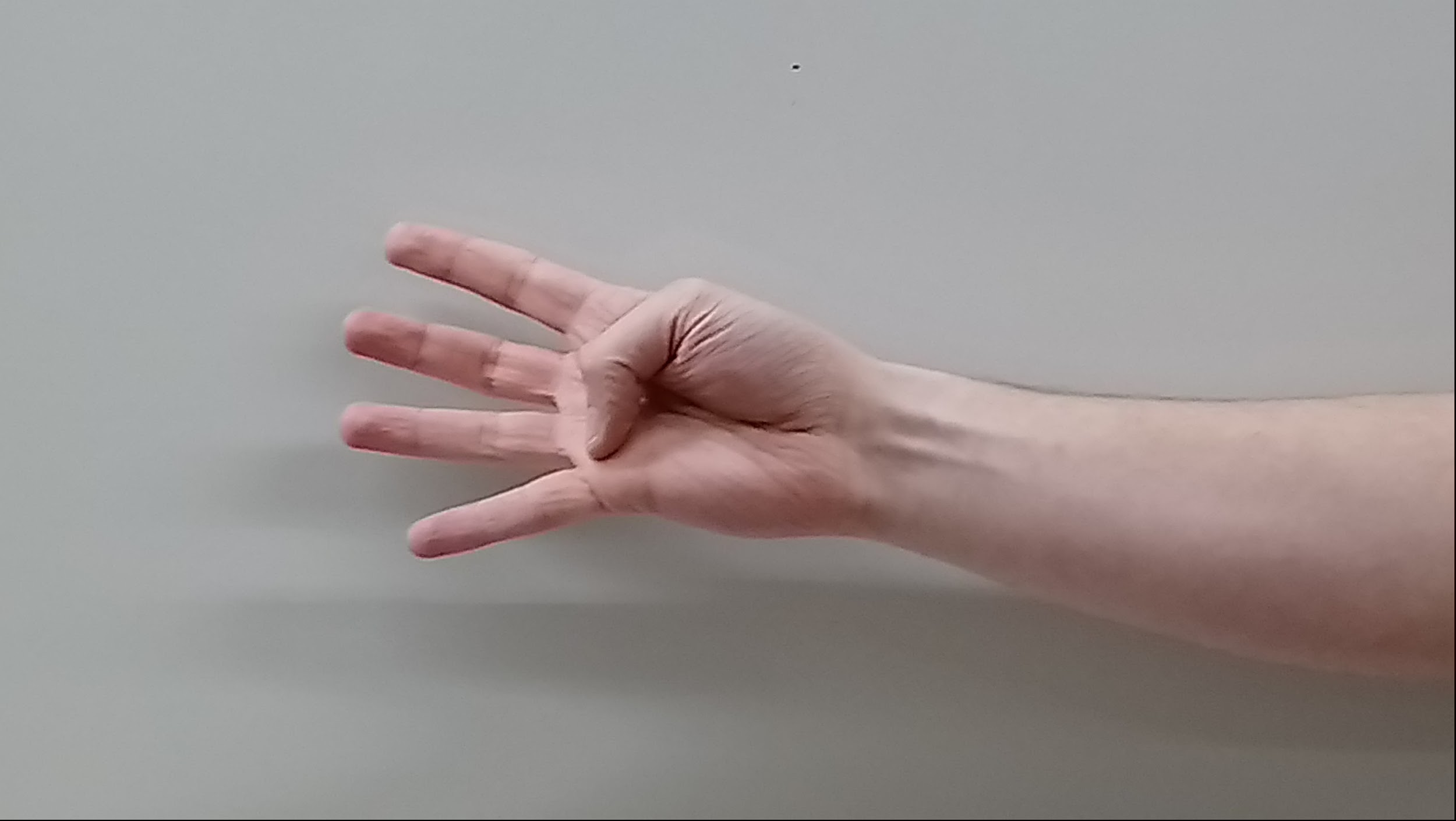


Figure 7: Thumb Flex



Figure 8: Wrist Flex



Figure 9: Elbow Flex

The subject performed each action twenty times before moving on to the next action over the course of a few hours on one day. The subject was asked to not move his arm or fingers too much during the recording of the data for the eighth class, the rest class.

After all the data had been collected, there were a total of 256 samples, with 117 for the rest class and about 20 for each of the other seven classes. Some samples for the rest class and other classes had to be discarded due to issues with the action not being properly recorded or there being far too much noise in the signal due to bad connections to the BioRadio, which was why there were not exactly 20 samples for each of the seven action classes and not a round number of samples for the Rest class. Each sample was recorded in a separate file, with the file name containing the project name, the name of the action, and the index of the sample. The format of the data for each sample was such that the columns were the channels and the rows were the timesteps. After all the samples had been collected, the files were iterated over to find the sample with the smallest number of timesteps and that value was used to clip all of the other samples to the same size to provide uniformity across all samples.

**4.2 PREPROCESSING TECHNIQUES**

For preprocessing the data for each sample, which each contained 2,640 timesteps totaling approximately 2.75 seconds per action at the sampling rate of 960 Hz and six different channels, features were extracted both from the time-domain and frequency-domain data. The mean, max, root mean squared ( RMS ), and variance were extracted from each channel for each sample in the time-domain. The sample data was then converted to the frequency domain and a Butterworth filter was used to filter out the DC component at 0 Hz and the 60 Hz present from the lights in the room. The aforementioned features were extracted from the absolute value of the frequency-domain data. In a NN, data can become quite large through the learning process, so the extracted features were normalized across all channels before being fed into the NN.

Due to the imbalance of the 117 samples available for the Rest class data and the 20 samples available for each of the other classes, the other classes were oversampled using the existing samples until there were 117 samples for each of the eight classes. The oversampling technique was used both on the EMG data collected for these experiments as well as on the validation dataset. The validation dataset contained an equal number of samples for all classes, but to give the SVM and RNN more data to train on, all classes were oversampled until 100 samples were available for each of the 6 classes.

Since the available toolboxes in MatLab were used for building the two classification systems with an RNN in one and an SVM in the other, the data needed to be in a specific format for each. The SVM required that all the data for each sample be in one row and that one of the columns have an encoded class name. For the SVM, each of the features was extracted and concatenated together to form a row. The extracted feature name and the channel for which the feature was extracted were included in the name of the column in the table. The class names were encoded to numeric values that were stored in the last column of the table, with a mapping of the class names and their respective encodings held in memory. The RNN required a cell array with each cell containing a matrix with the rows as the channels and the columns as the features or time-steps. For the purposes of these experiments, the columns were the extracted features for the channels. There were a total of eight different extracted features for each channel.

After all of the preprocessing and formatting was completed for each of the classifiers and the data had been oversampled, the data was split between training and testing. Indices in the data were randomly selected using built-in MatLab functions and those were used to select 75% of the data for training and 25% for testing. A special function was built to ensure that 75% of each individual class’s data was allocated for training and 25% of each class’s data was allocated for testing. In the end, there was an equal number of samples for each class in the training and testing sets, respectively. Once the training and testing data had been randomly selected from the sample data, it was randomly shuffled, so that the classes were not presented to the RNN and SVM in chunks. The entire feature extraction process is represented in Figure 10.

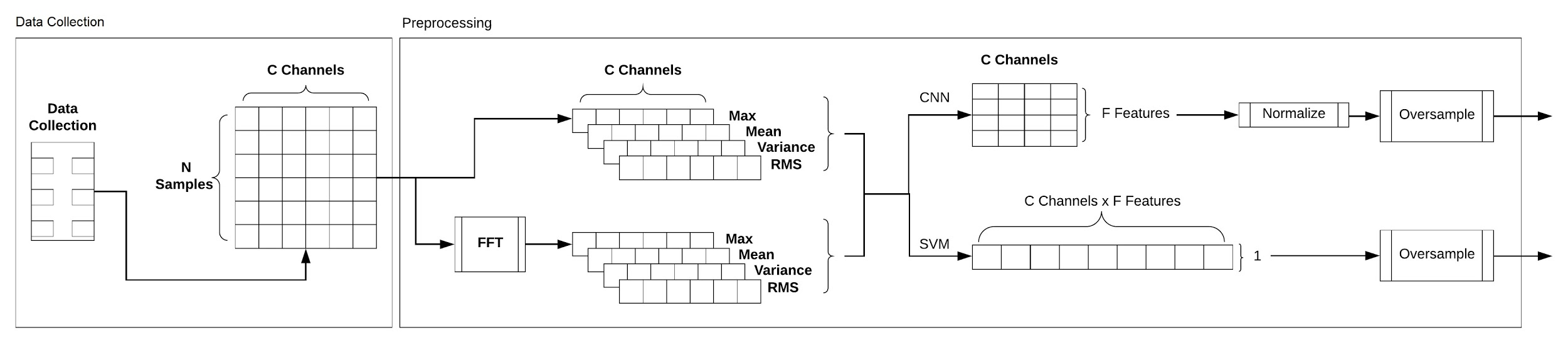


Figure 10: Preprocessing / Feature Extraction Block Diagram

**4.3 LEARNING METHODS**

An RNN and an SVM were constructed and used as classifiers for these experiments, given the success others had with these or similar methods. Both the RNN and SVM were constructed through trial-and-error in order to find a classification structure that worked well for the sEMG data collected and for the validation dataset.

After numerous variations in the design, a simplistic RNN structure with 10 hidden layers and a learning rate that decayed by 90% each epoch and started at a learning rate of 0.01 was built and trained. The decaying learning rate was used to reduce oscillations in the learning process as the RNN classifier converged to a solution. The RNN was used over other NN designs, because it provided the best performance compared to the other NN designs available in the MatLab toolboxes. The results of the other NN designs were not recorded, because they were not used in the final classifier designs for these experiments. The RNN differs from a regular NN in that it contains feedback loops, so that information from previous training sessions feed back into the network, giving the network a sort of memory [33].

Construction of the SVM was also through a trial-and-error process, whereby numerous configurations were attempted. Many different kernel functions were attempted. The linear and quadratic kernel functions had lower accuracies than the cubic polynomial kernel function, so the cubic polynomial kernel function was used. Other than the kernel function, all of the other options in the SVM were left unchanged from the defaults in the final implementation of the SVM. A block diagram of entire RNN and SVM classification processes was created and recorded in Figure 11.

Both the RNN and SVM were trained and tested on both the collected sEMG data and the validation data for a total of 10 different sessions. When the RNN was trained on the collected sEMG data, it trained over the course of 600 epochs. When the RNN was trained on the validation data, it trained over the course of 400 epochs. The number of epochs to use for training to achieve the desired level of accuracy for both datasets was determined through trial and error to find a good balance between overtraining the classifier and minimizing the training duration.

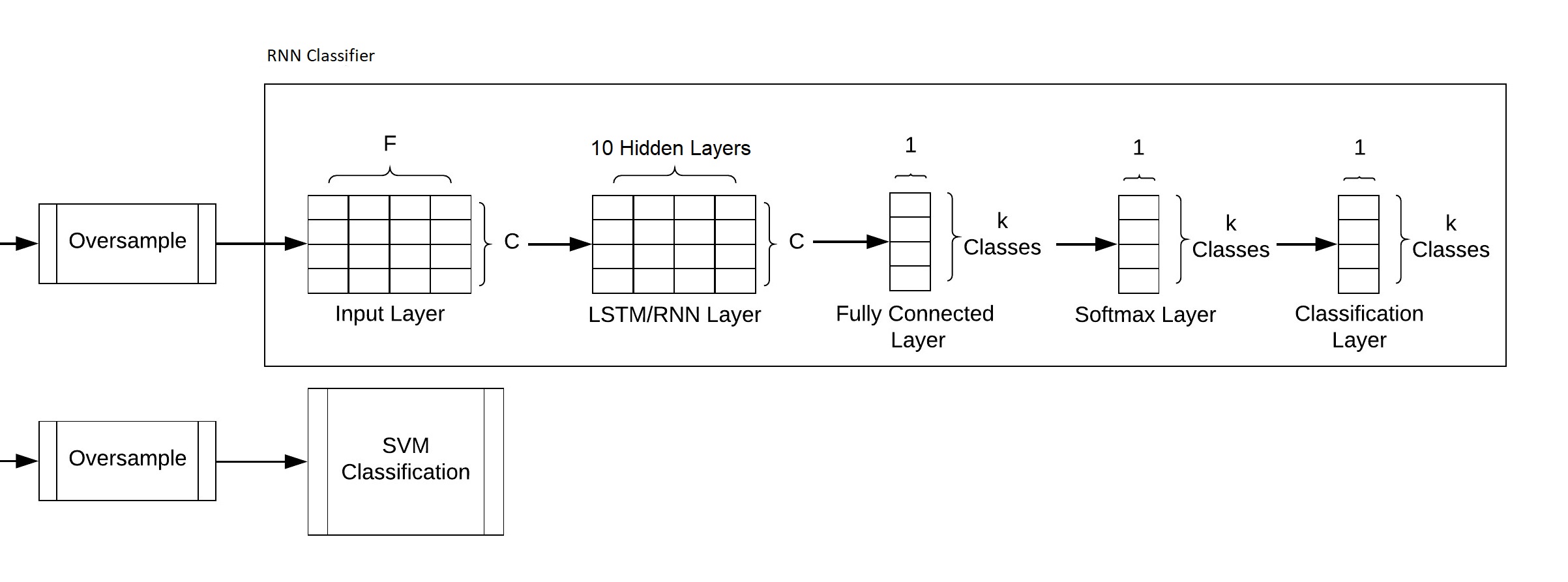


Figure 11: Classifier Block Diagram

**5. EXPECTED RESULTS**

The goal of this research was to classify 7 different hand and arm gestures, as well as a rest class as the 8th. Based on the literature survey, an accuracy of 90-95% was expected when using an RNN and an SVM. The designs used for these experiments required a much larger time frame ( 2.75 seconds ) than used in the research papers previously cited ( 0.1 to 0.3 seconds ), so the accuracy was expected to vary slightly from the expected results. Since the time frame used for these experiments was much larger than that used in most other research papers, a higher accuracy was expected given that the classifiers had more information to work with and to use to discriminate between classes.

**6. RESULTS**

**6.1 COLLECTED SURFACE EMG DATA PREPROCESSING VISUALIZATIONS**

Some graphical depictions of all or a sampling of the sEMG data collected for these experiments are represented in Figures 12 through 15. The figures serve to convey a basic understanding and feel for the shape and nature of the data collected and start to illustrate how the RNN and SVM classifiers may have used the extracted features to discriminate between the different classes. The plots in Figures 14 and 15 do the best job of illustrating how the features extracted from each of the channels could be plotted against one another and used to group the data for each class.

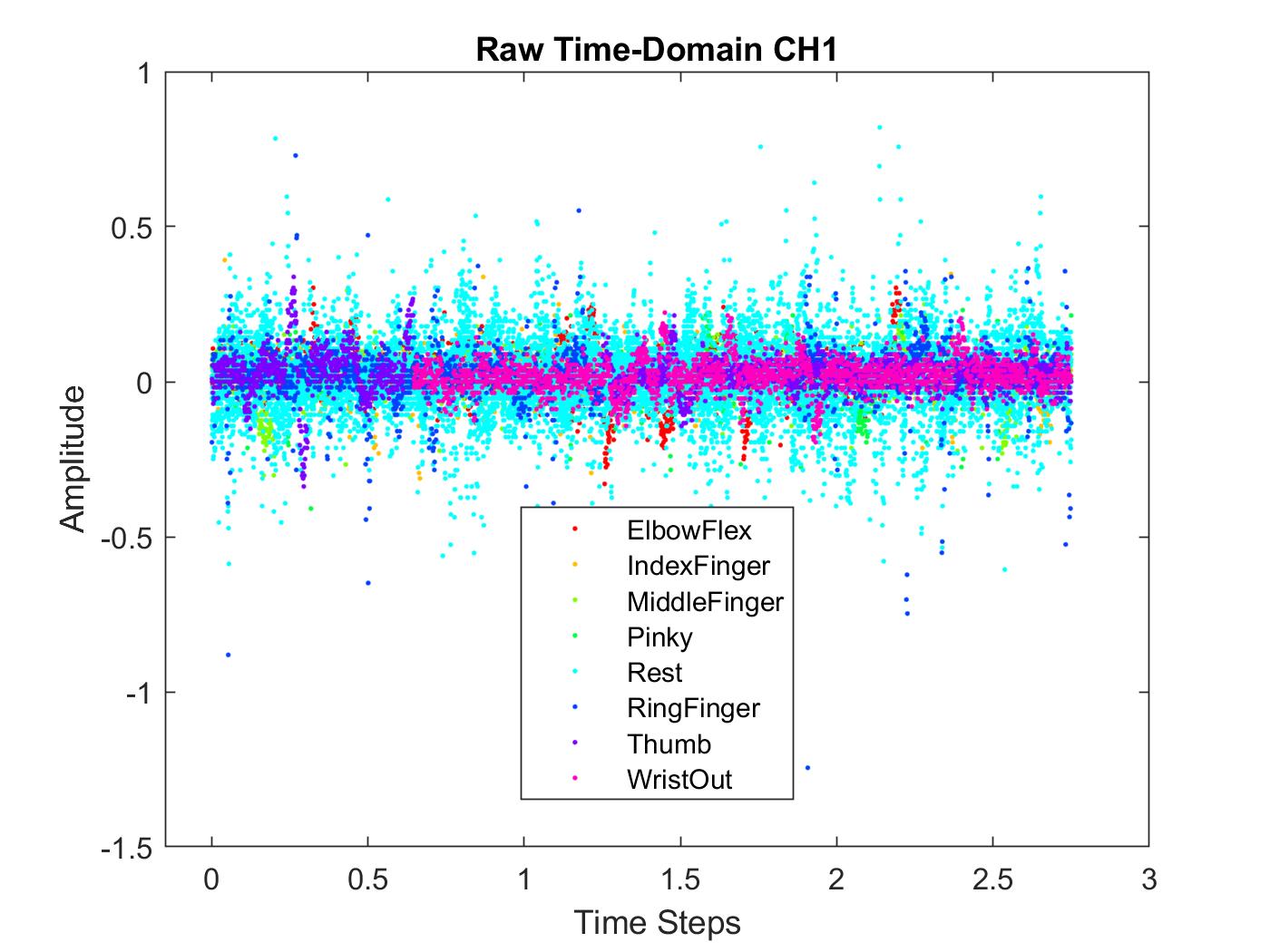


Figure 12: Channel 1 Raw Time-Domain EMG Data

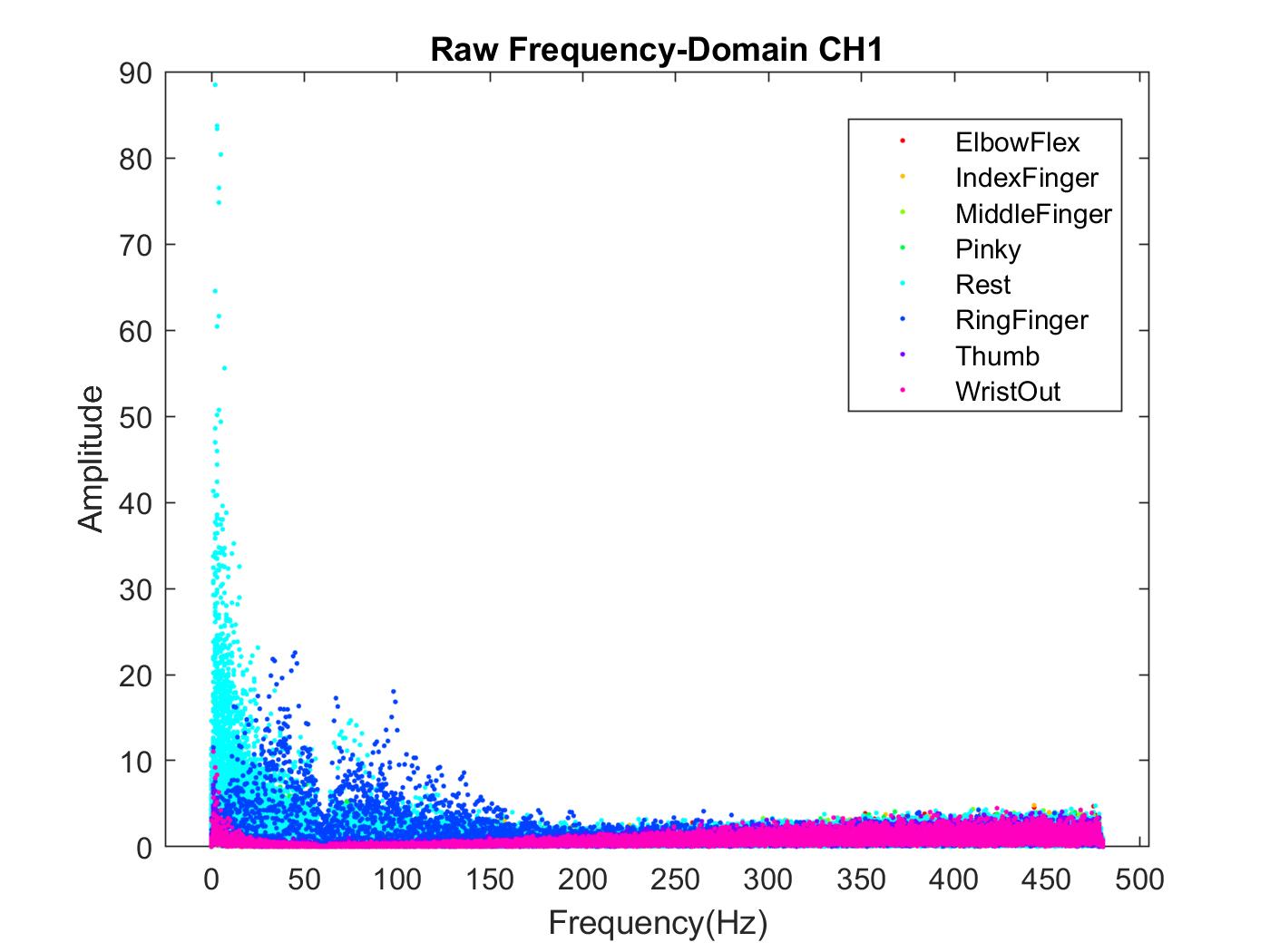


Figure 13: Channel 1 Raw Spectrum EMG Data

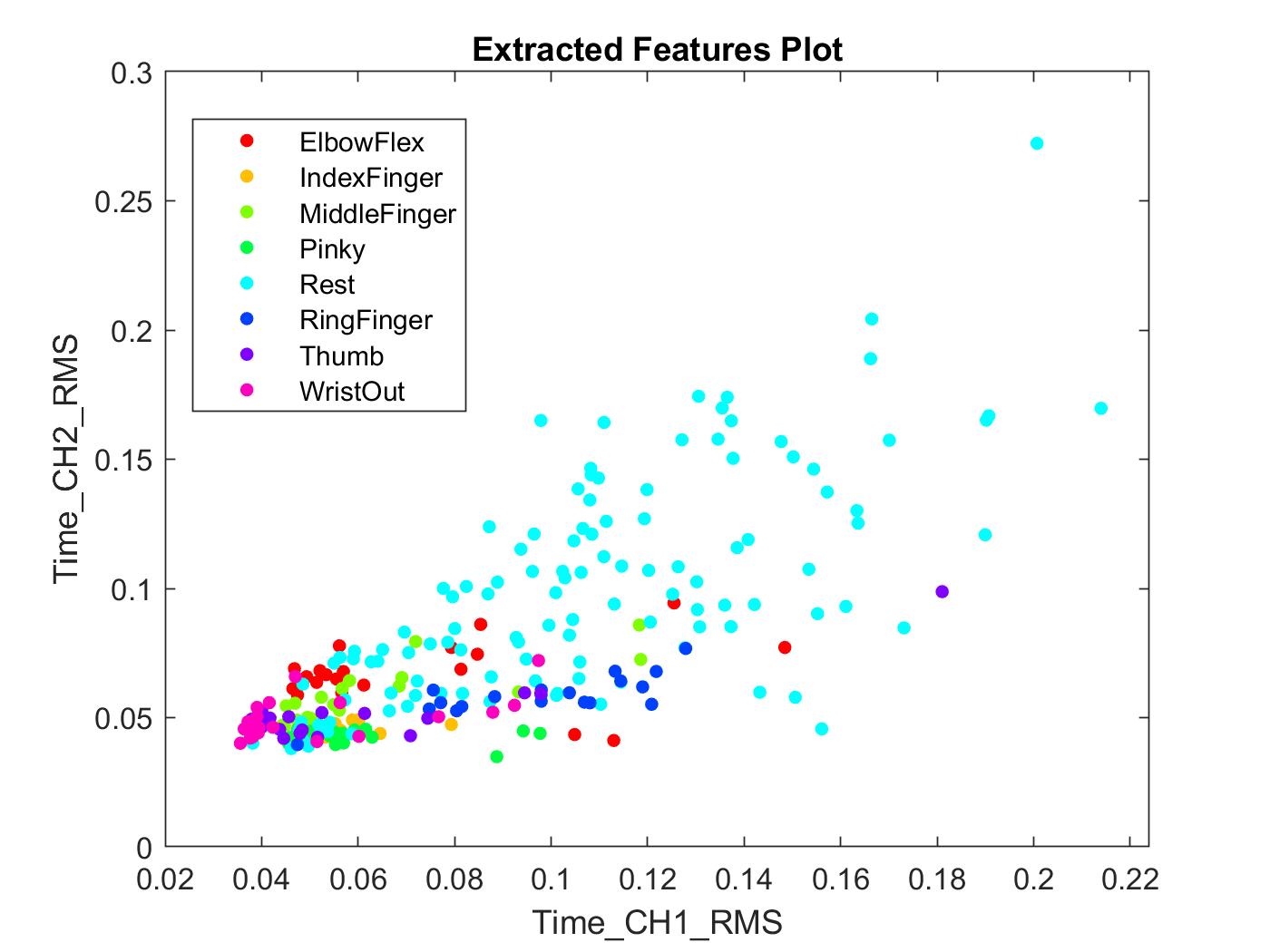


Figure 14: Feature Extraction Data

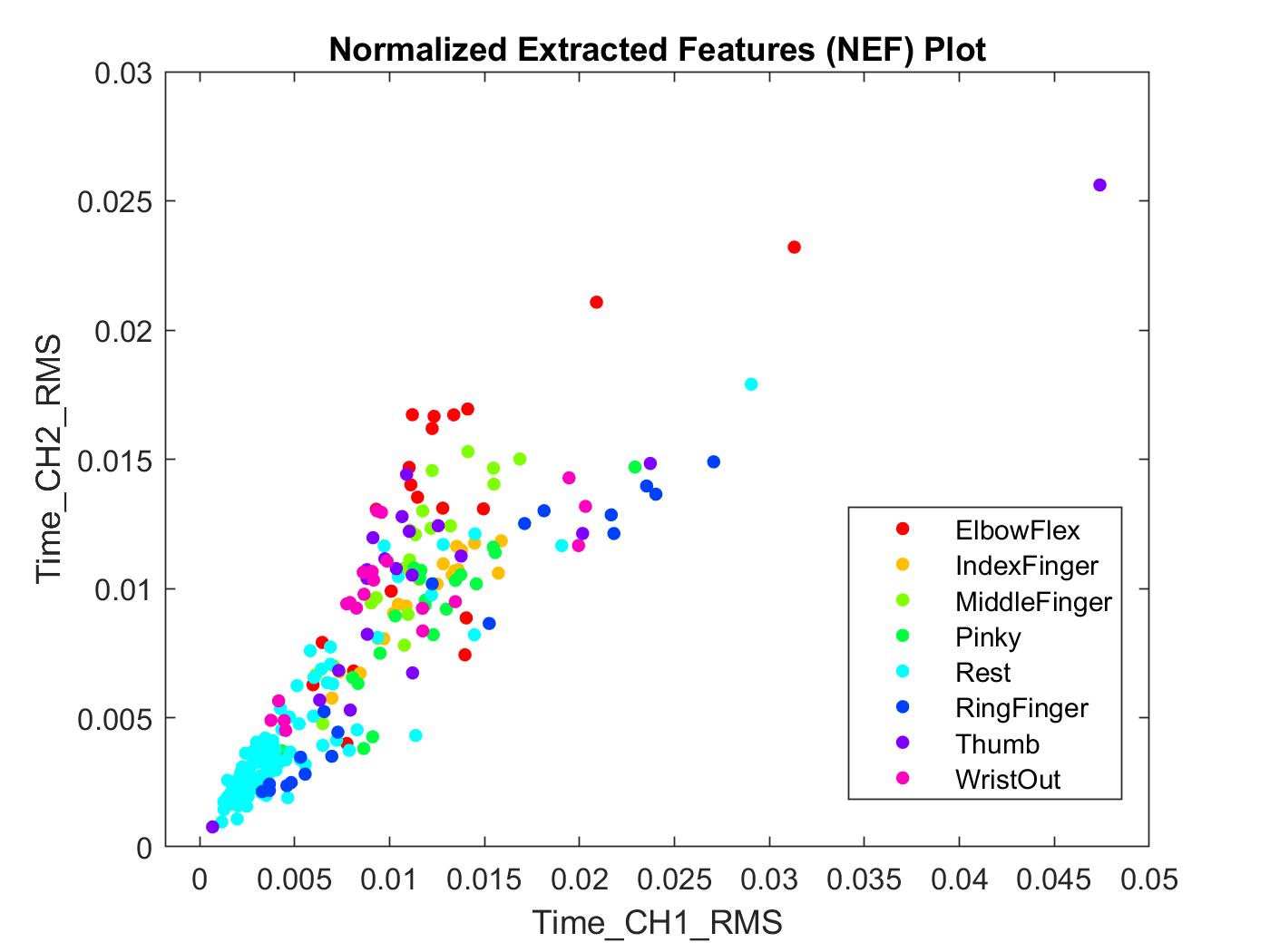


Figure 15: Normalized Feature Extraction Data

**6.2 COLLECTED SURFACE EMG DATA CLASSIFICATION RESULTS**

The SVM and RNN were trained and testing ten times separately, with a random selection of training and testing data collected for each session of the process. The predicted and expected classes were recorded and combined across all ten sessions for both classifiers. The confusion matrices, across all 10 sessions, for both the training and testing classification results for the SVM were recorded and represented in Figures 16 and 17.

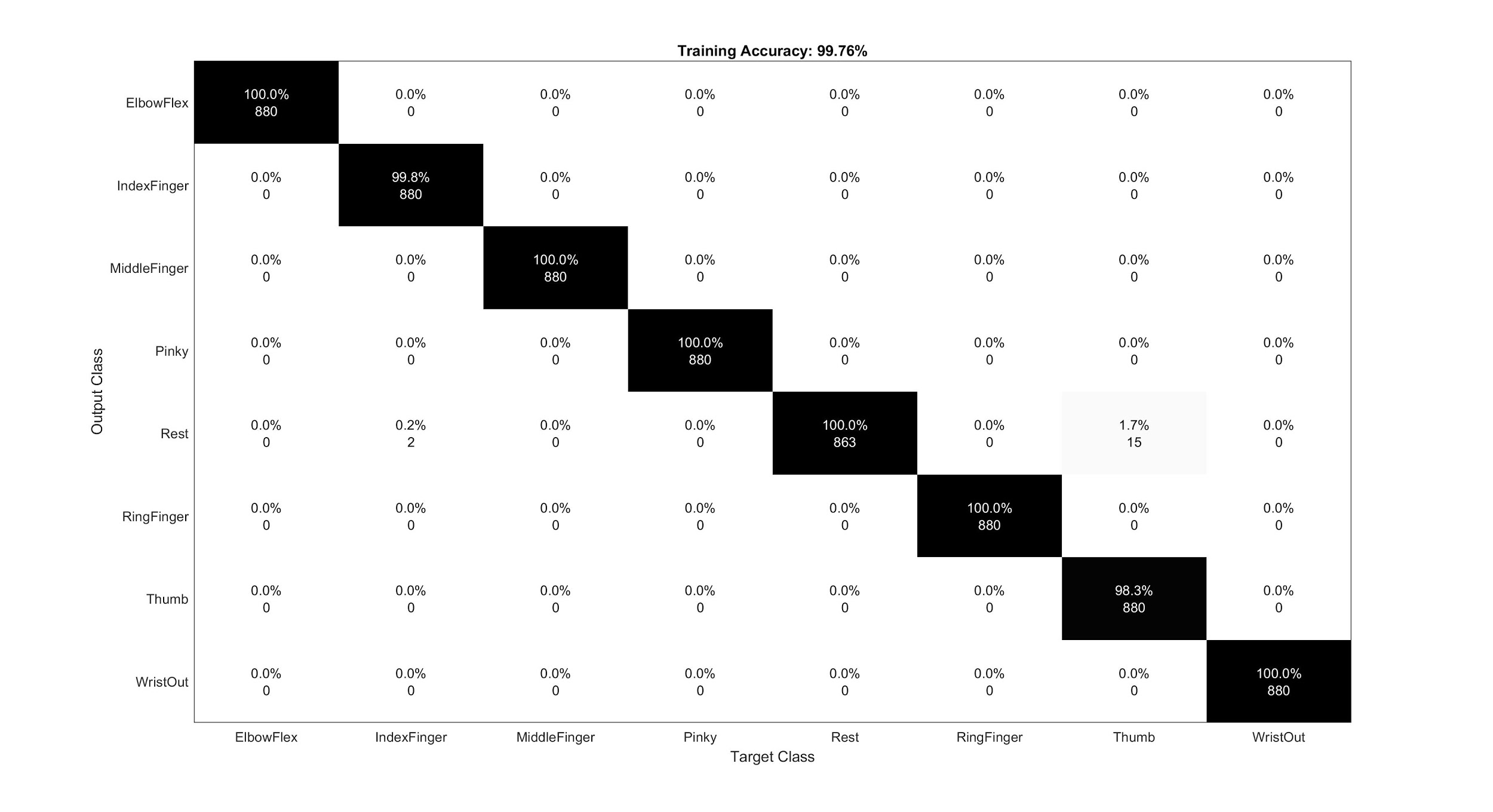


Figure 16: SVM Confusion Matrix of Training Accuracy

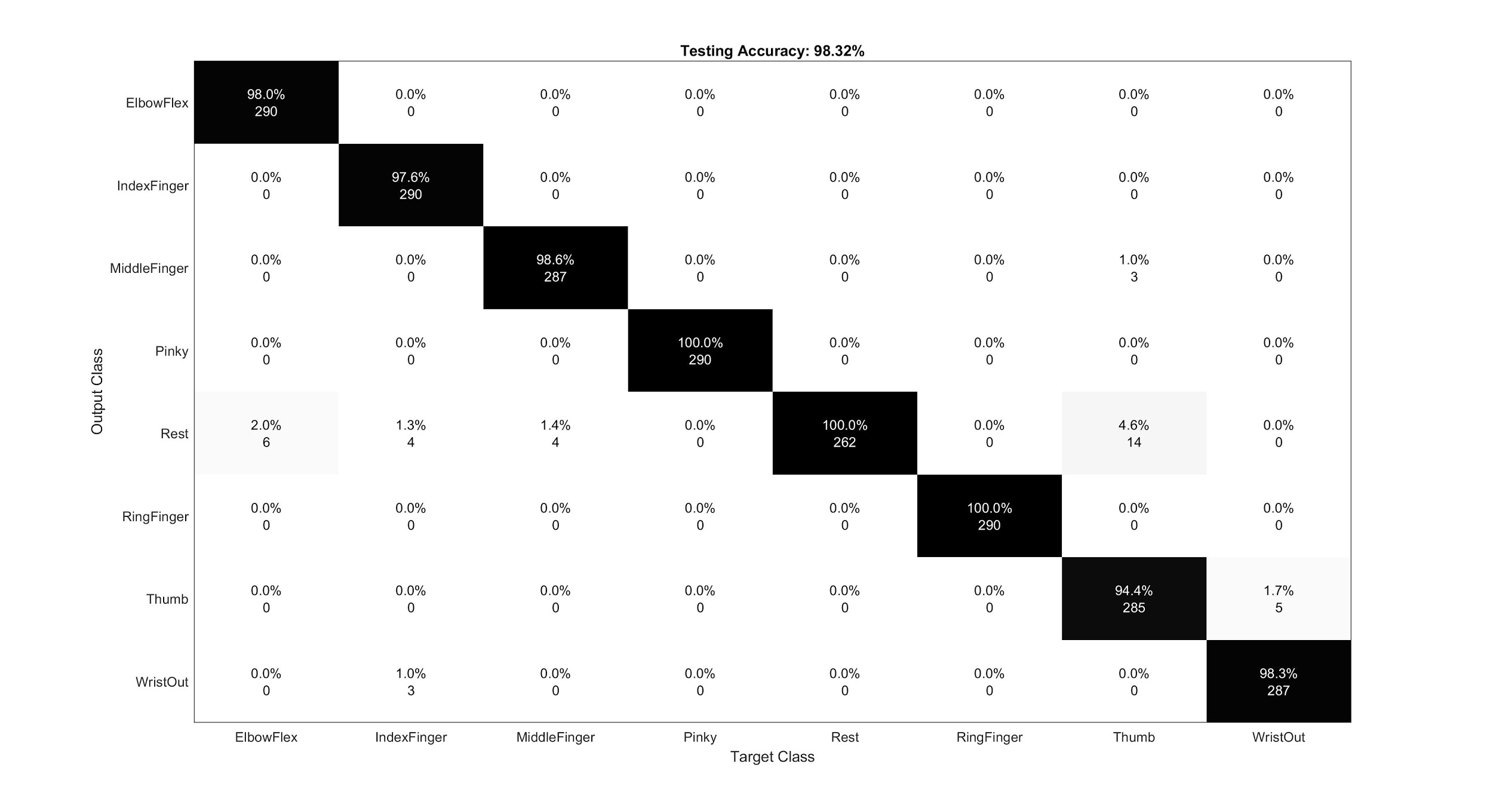


Figure 17: SVM Confusion Matrix of Testing Accuracy

Similarly, the confusion matrices for all 10 sessions of training and testing for the RNN were recorded and represented in Figures 19 and 20, while one of the ten training sessions was recorded in Figure 18.

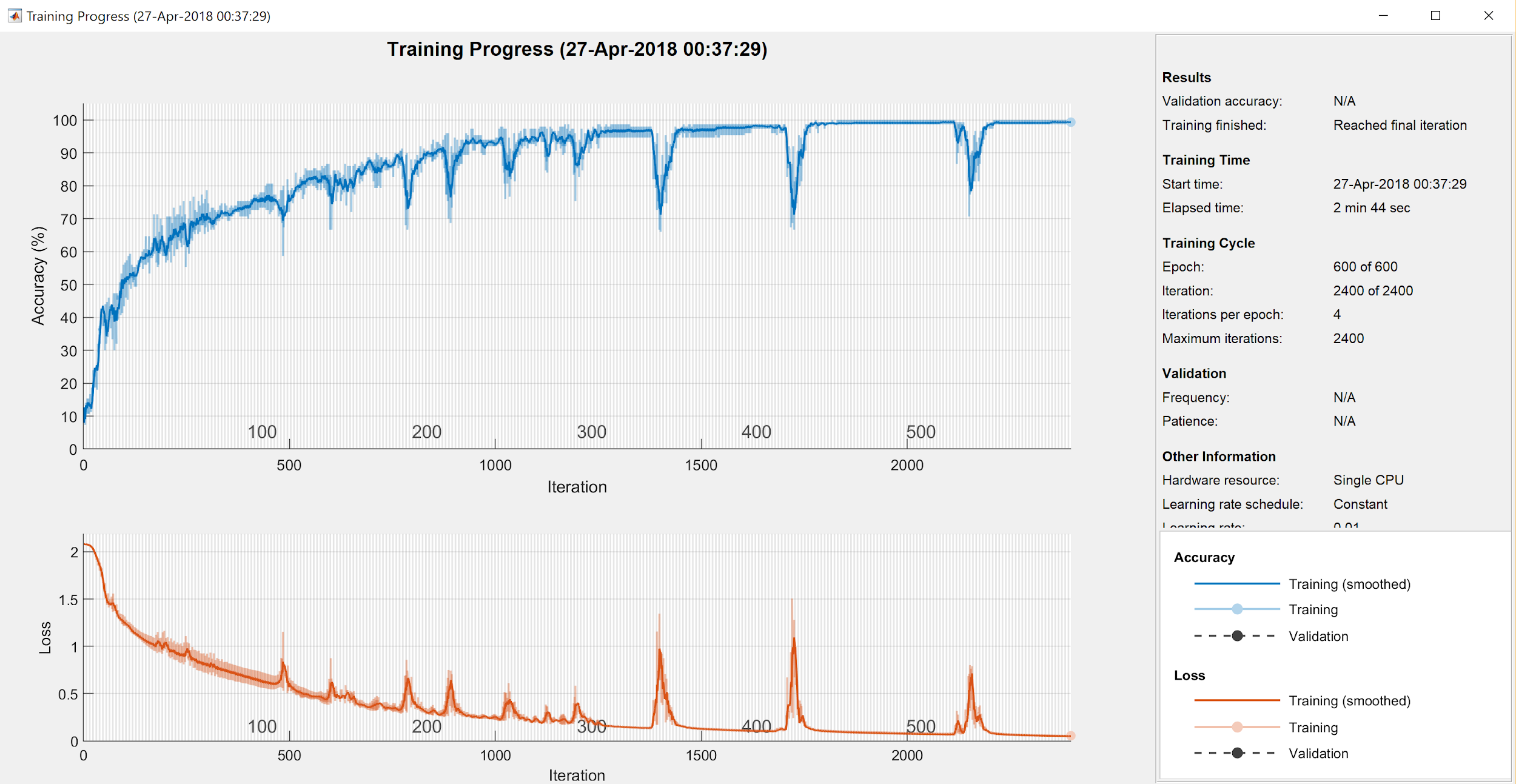


Figure 18: RNN Training Plot of 10th Session

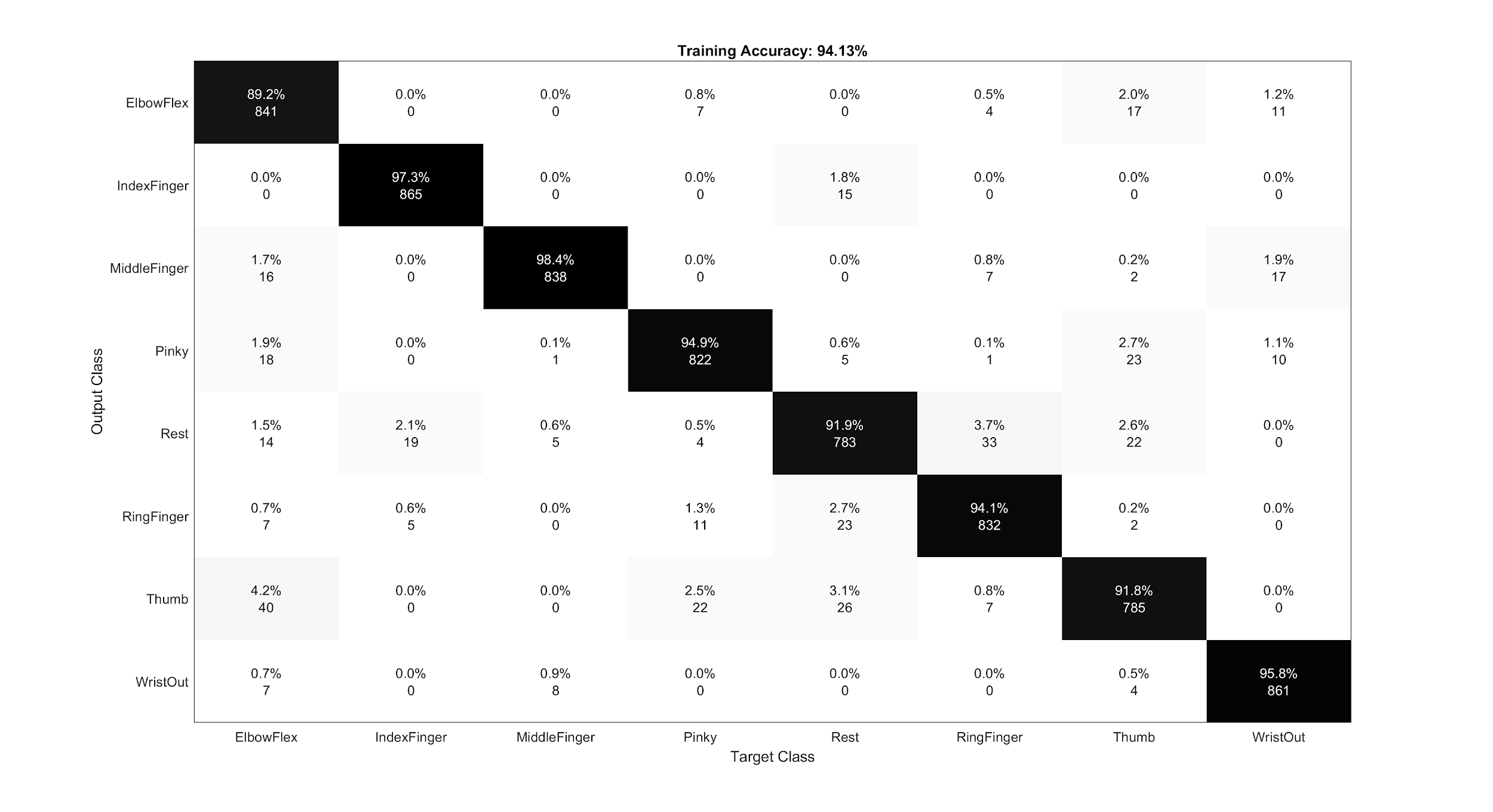


Figure 19: RNN Confusion Matrix of Training Accuracy

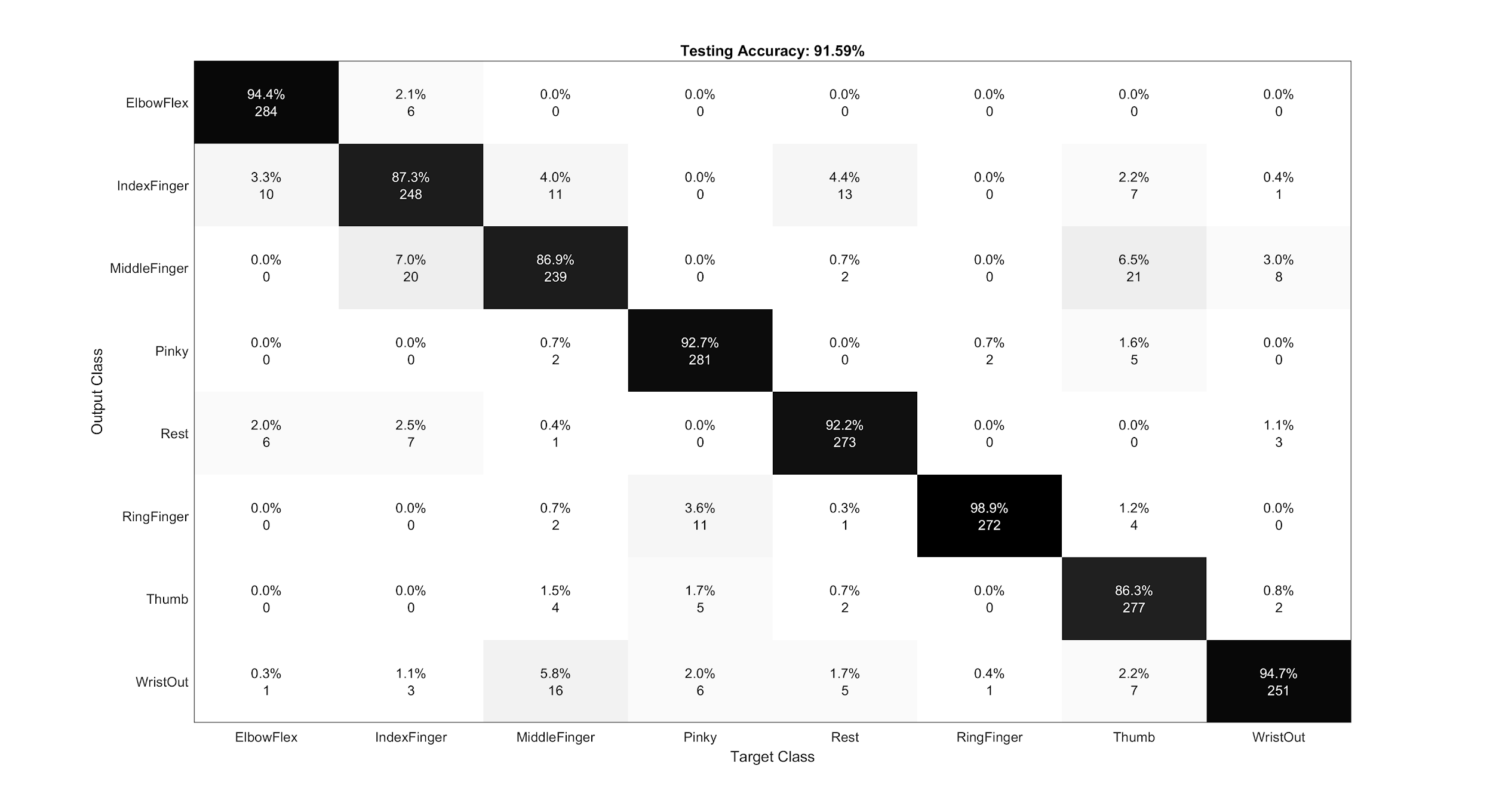


Figure 20: RNN Confusion Matrix of Testing Accuracy

**6.3 VALIDATION DATA PREPROCESSING VISUALIZATIONS**

Some graphical depictions of the raw data in the time-domain, the filtered frequency-domain data, the extracted features from both the time and filtered frequency domains combined and the normalized extracted features all generated from the validation dataset were recorded and represented in Figures 21, 22, 23, and 24, respectively. Similar to the plots in Figures 14 and 15, the plots in Figures 23 and 24 again serve to illustrate how the data for each class of action may be discriminated between by the RNN and SVM classifiers using the extracted features. The classes in the validation set are far easier to group than the classes of the collected data with the features extracted for these experiments, as illustrated by comparing Figures 14 and 23.

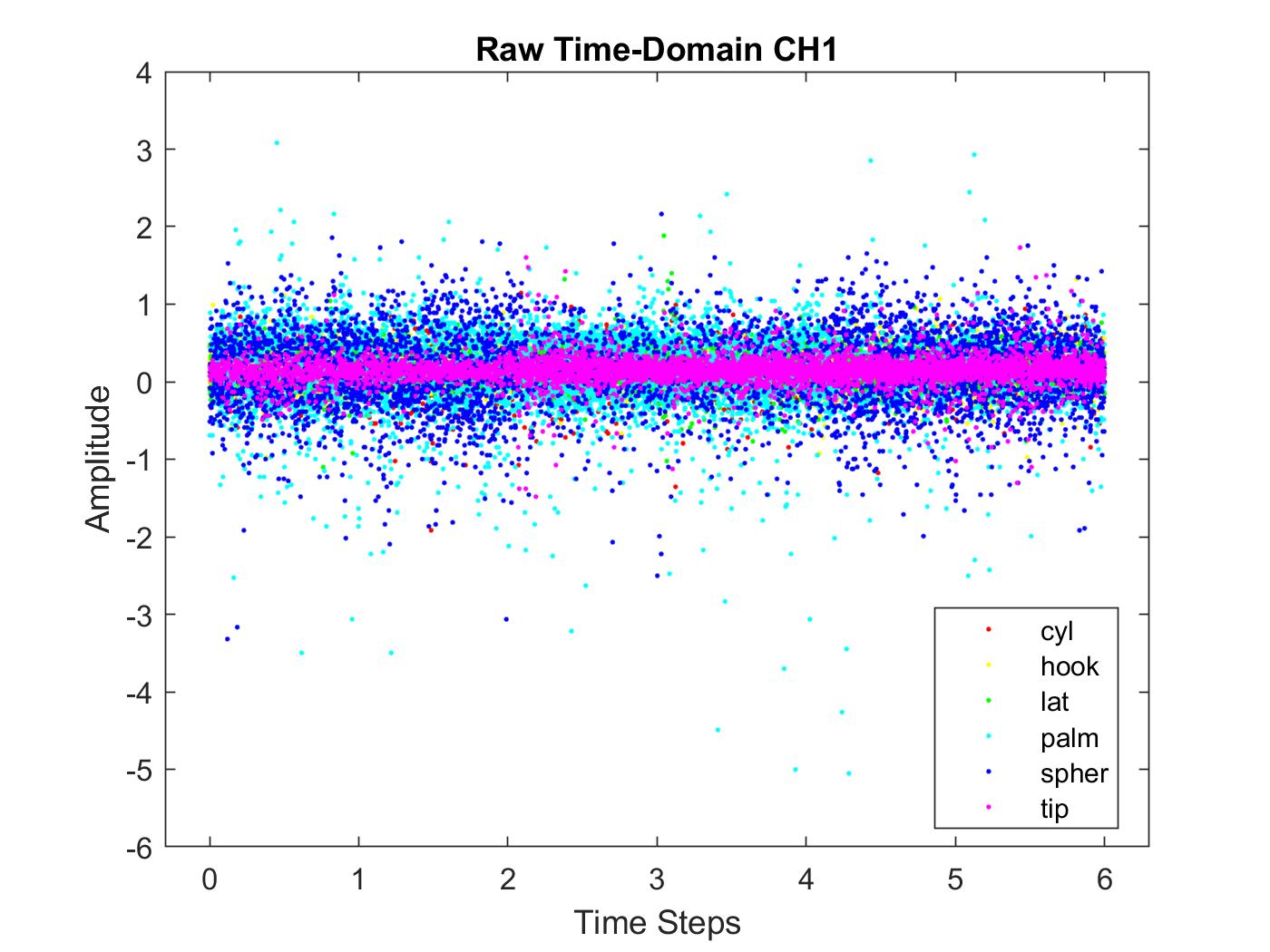


Figure 21: Channel 1 Raw Time-Domain EMG Validation Data

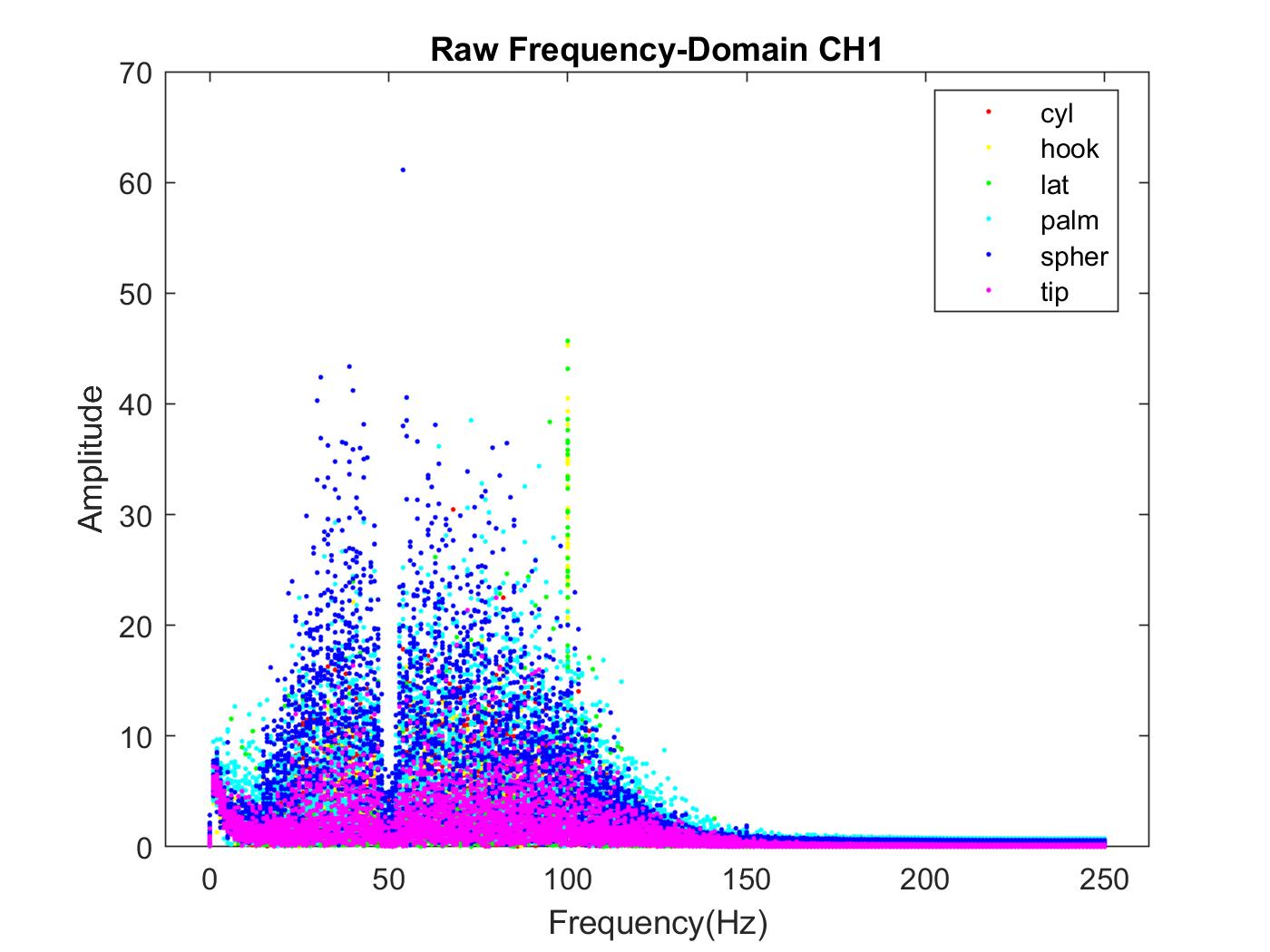


Figure 22: Channel 1 Raw Spectrum EMG Validation Data

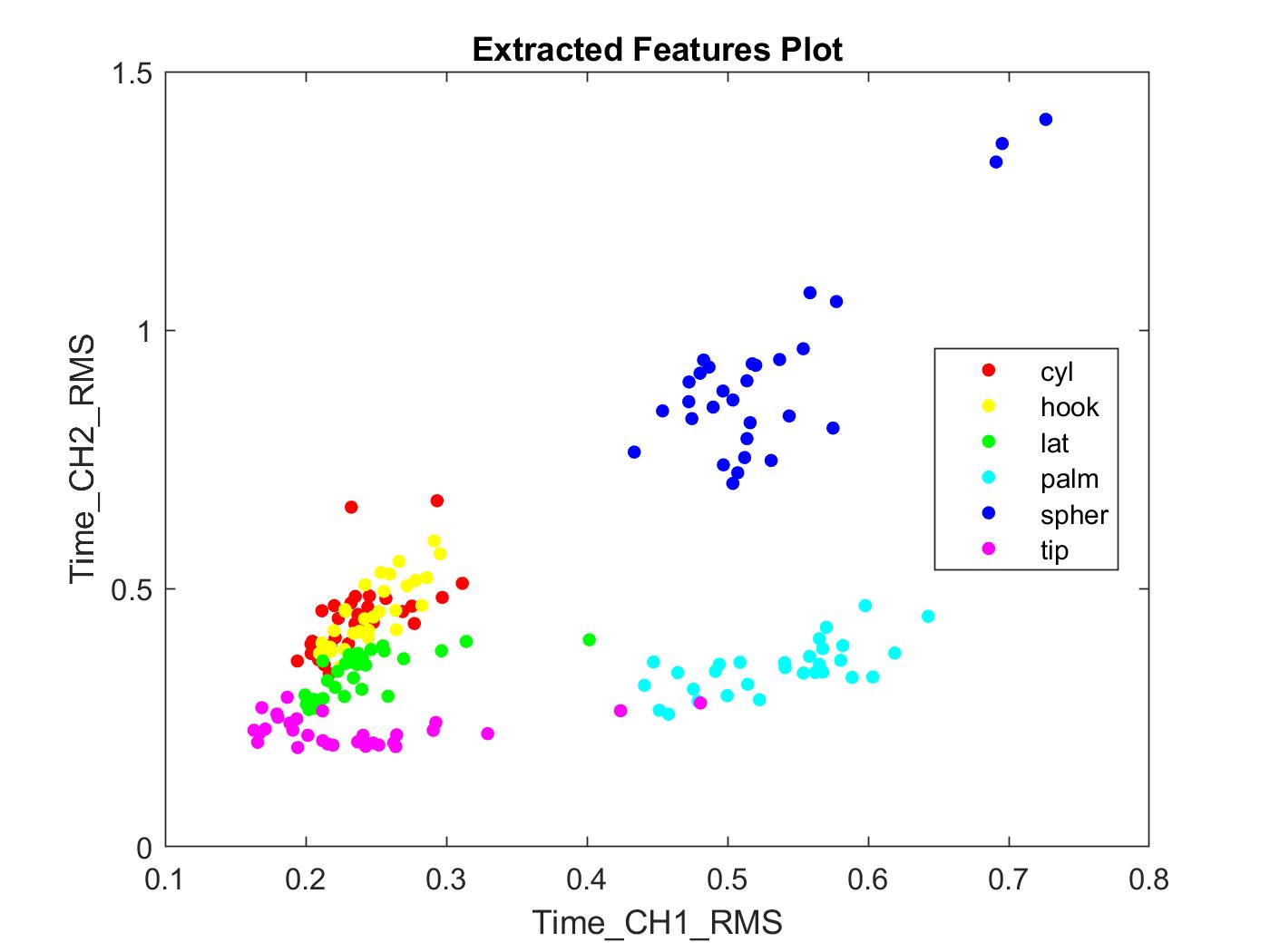


Figure 23: Feature Extraction Validation data

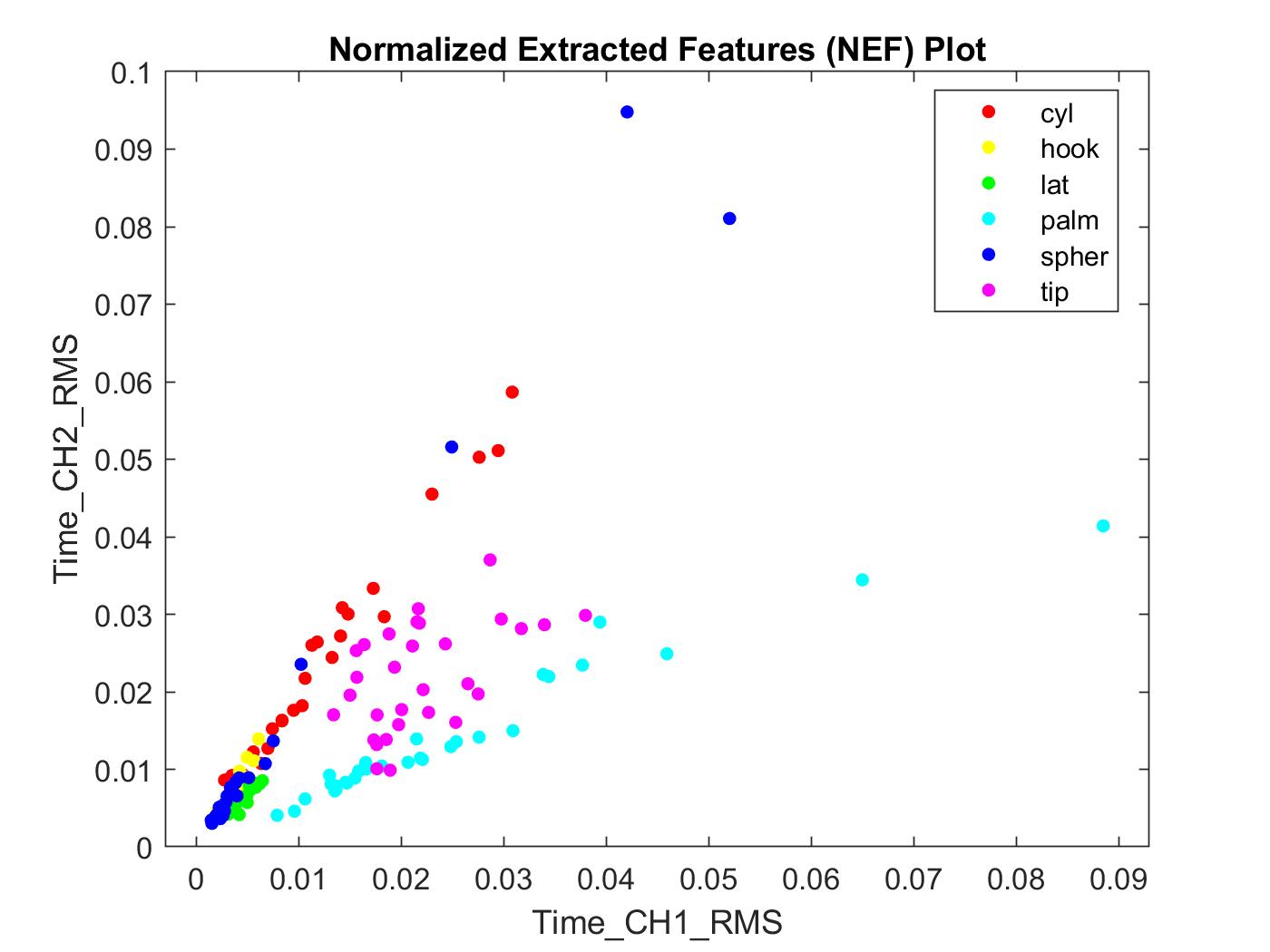


Figure 24: Normalized Feature Extraction Validation Data

**6.4 VALIDATION DATA CLASSIFICATION RESULTS**

The accuracies recorded by the researchers who generated the third-party dataset were depicted in Figure 25. The **male 2** dataset and respective classification accuracy of 94.8% were of interest for these experiments, because only the action data for this subject was used for validation of the designs of the RNN and SVM classifiers.

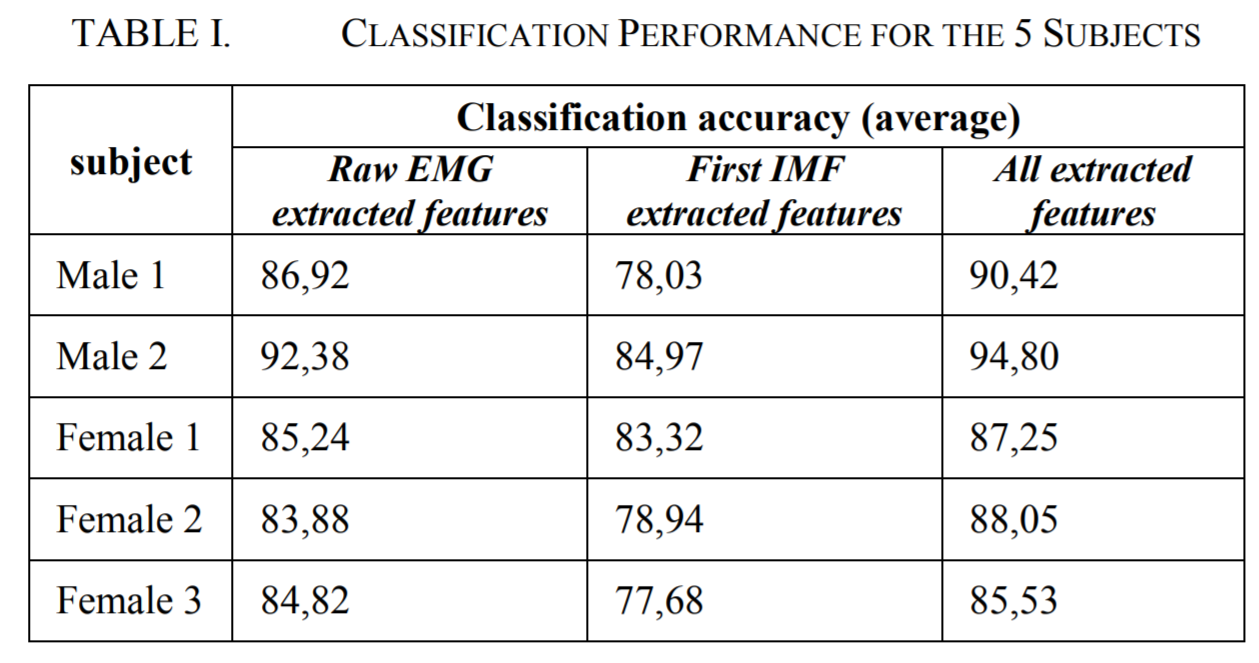


Figure 25: EMBC Published Validation Set Classification Accuracy [32]

The training and testing confusion matrices of the SVM when trained and tested on the validation dataset were recorded and represented in Figures 26 and 27.

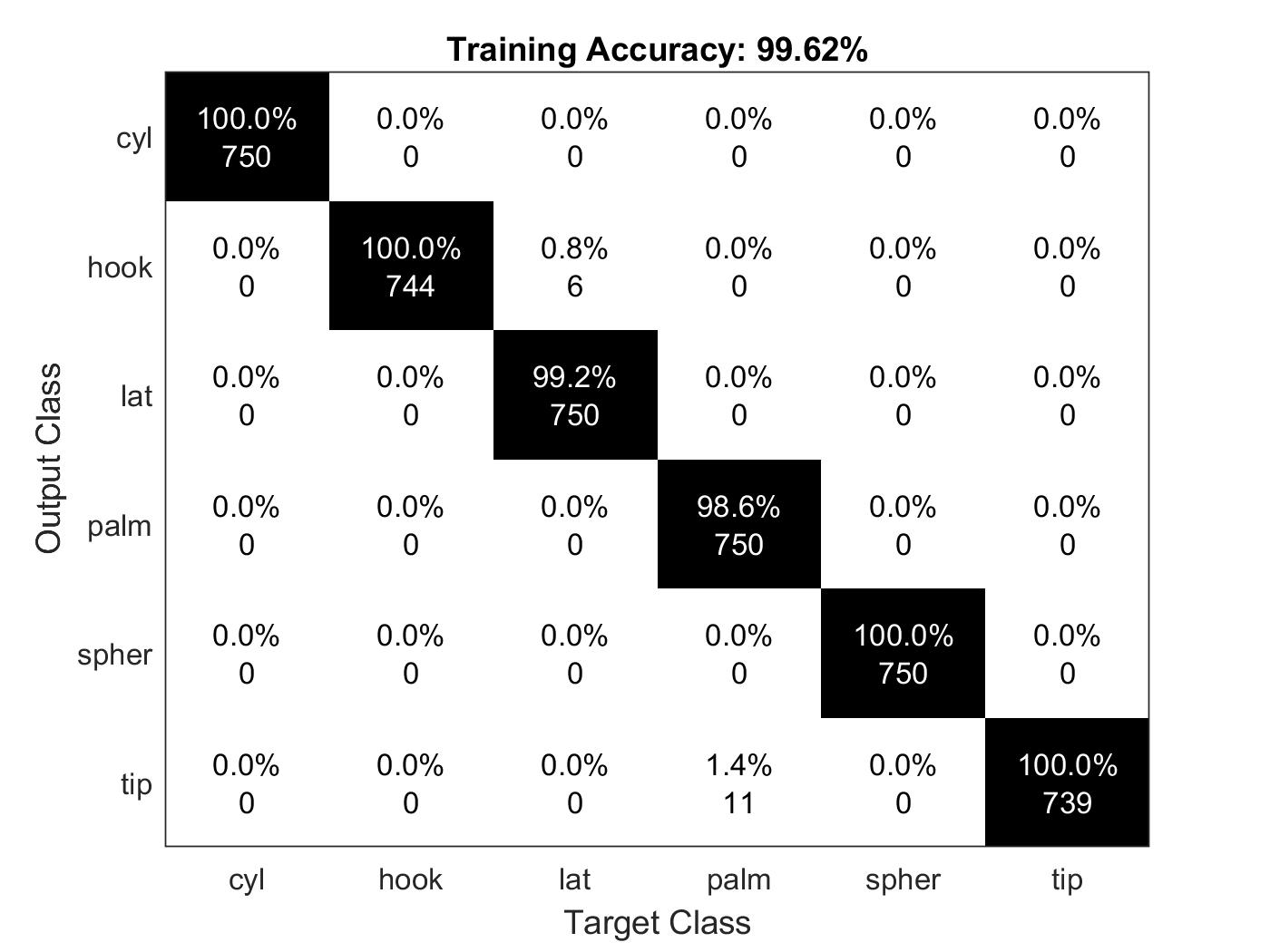


Figure 26: SVM Validation Confusion Matrix Training Accuracy

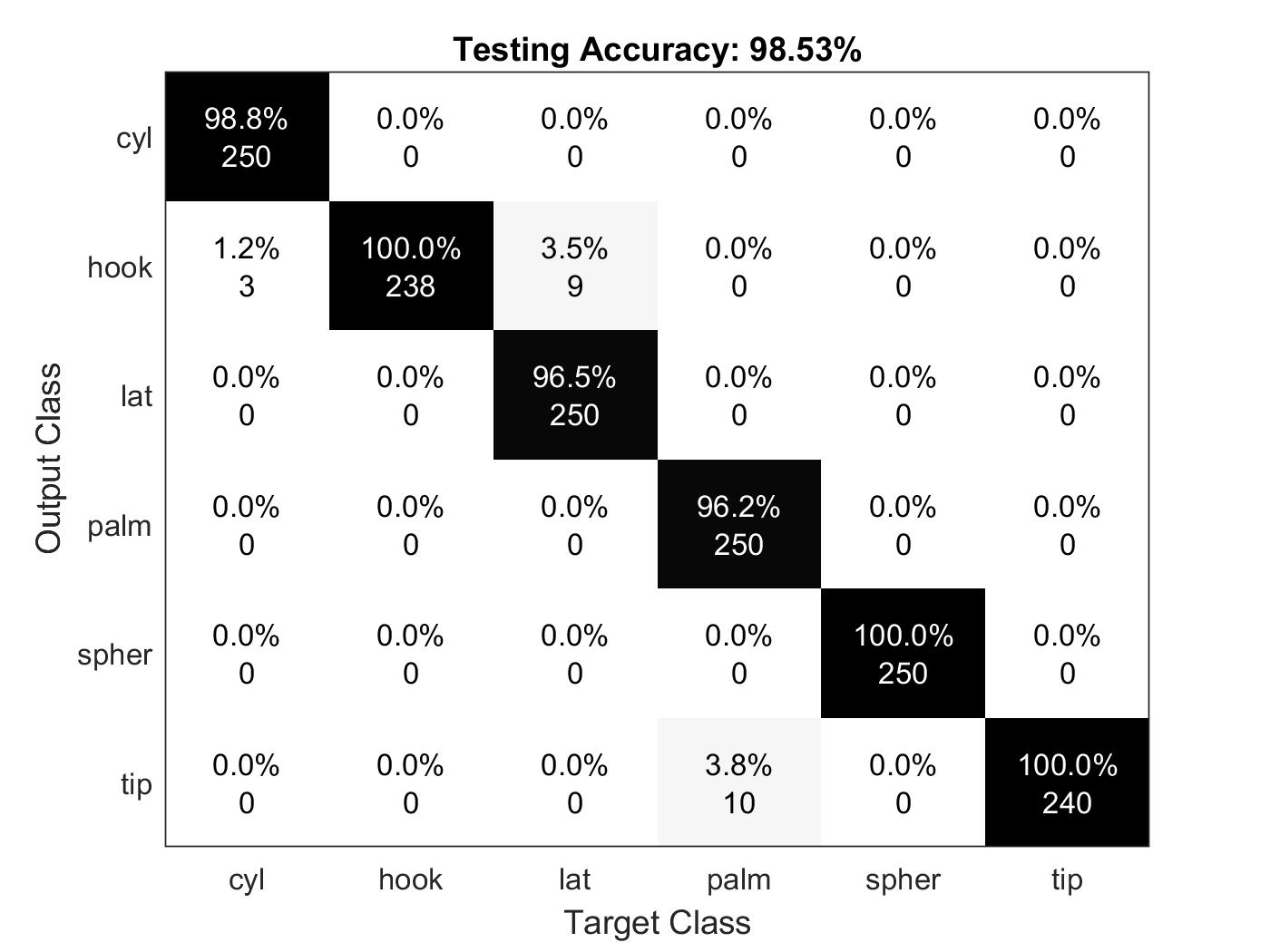


Figure 27: SVM Validation Confusion Matrix Testing Accuracy

Similarly, the training and testing confusion matrices for the RNN, when the validation dataset was used, were plotted and recorded in Figures 29 and 30. One of the training session plots for the RNN when the validation data was used was plotted and recorded in Figure 28.

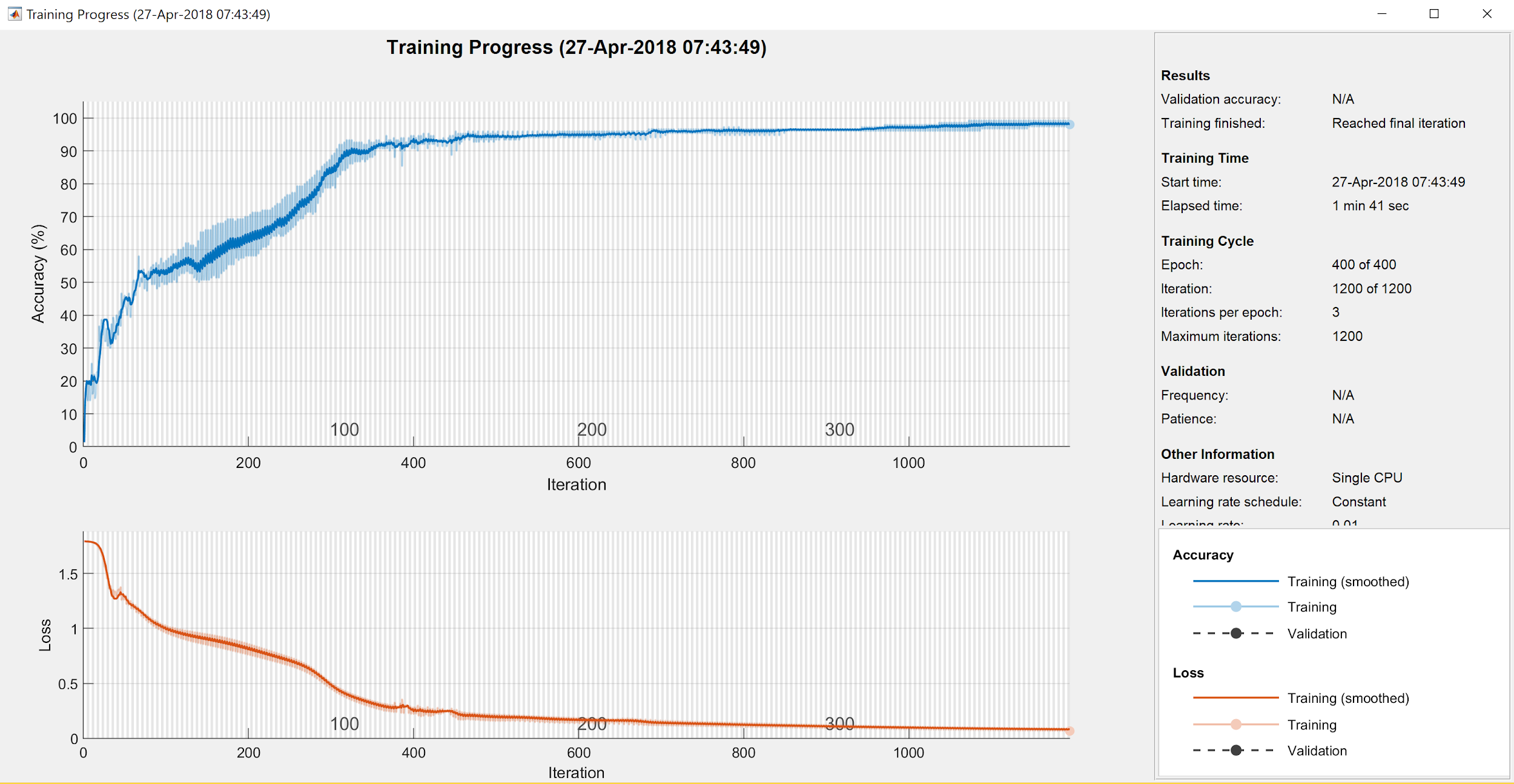


Figure 28: RNN Validation Training Plot 10th Iteration

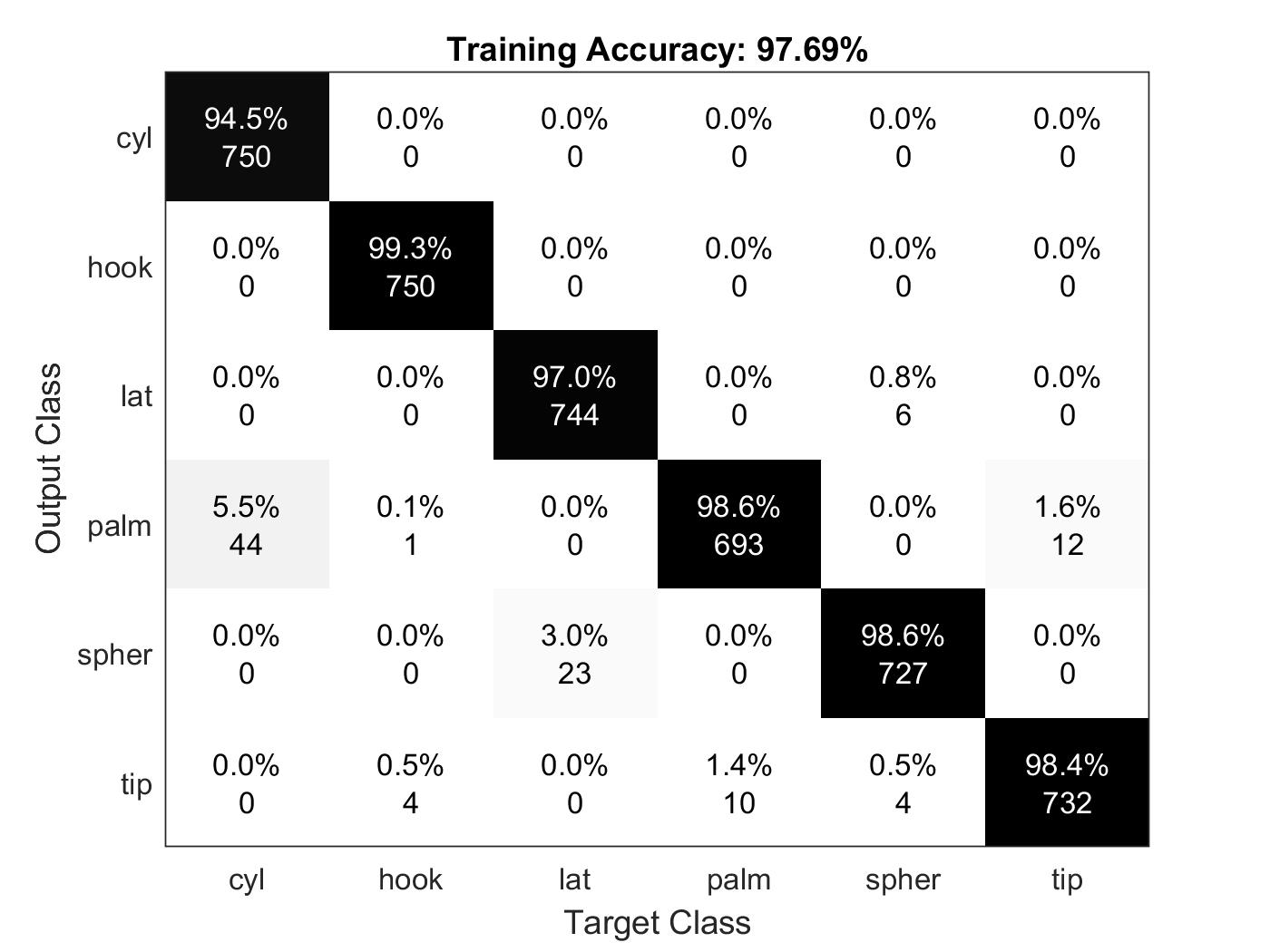


Figure 29: RNN Validation Confusion Matrix Training Accuracy

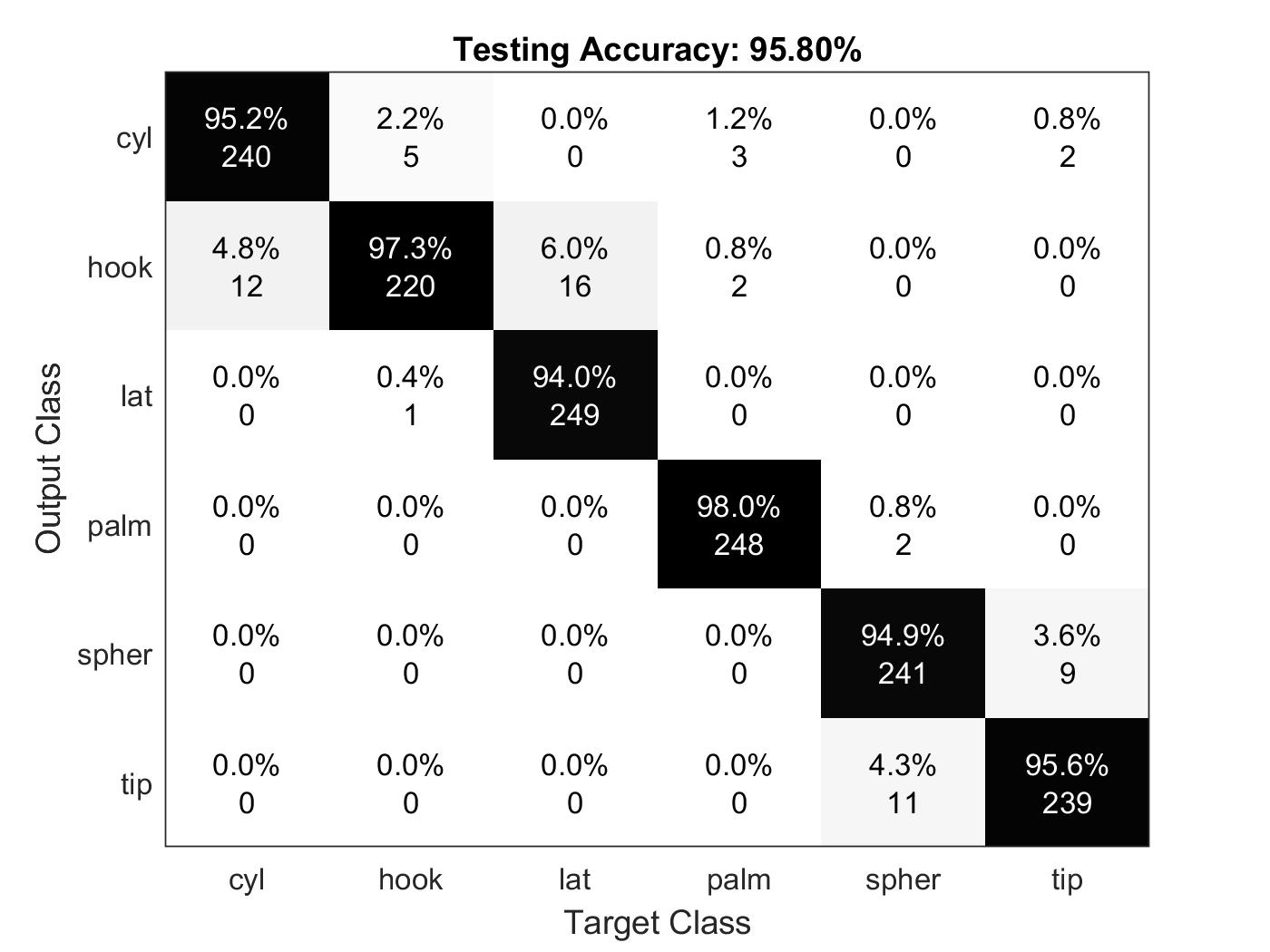


Figure 30: RNN Validation Confusion Matrix Testing Accuracy

**6.5 DISCUSSION**

The training and testing accuracies of the SVM - when the collected data was processed - were 99.76% and 98.32% [Figures 16-17]. Given that the training and testing accuracies were similar, the model was likely not overtrained. The SVM classifier mainly seemed to have some trouble distinguishing the actions of thumb flex, index finger flex, middle finger flex, and elbow flex from rest. One possible reason for this discrimination issue was that these actions were so slight and the amplitudes of their sEMG signals across the six channels so small between all the actions relative to the rest class that occasionally they appeared to the SVM classifier as though the subject was at rest.

Similarly, the RNN achieved a training accuracy of 94.13% and a testing accuracy of 91.59% [Figures 18-20]. The RNN seemed to have far more difficulty discriminating between some classes than others, as compared to the SVM. For example, in training the RNN predicted the rest class, when the actual class was ring finger, thumb, or index finger. The error rates for such mistakes in training were minimal, but they did stick out from the classification errors between other classes which were even smaller. These errors were somewhat expected based on the overlap between rest and many of the other classes in Figures 12 through 15. The RNN seemed to have especial trouble discriminating between the rest class and various finger flex classes and the elbow flex class and various finger flex classes. This failure to discriminate between these classes might suggest better placement of the electrodes or more accurate sEMG collection equipment would be required in order to increase discrimination between these similar actions. Comparing the training and testing confusion matrices of the RNN, the RNN had trouble discriminating between the thumb and elbow flexes and the elbow flex and index finger actions far more so in the testing portion than in the training portion, suggesting that the RNN may have been slightly overtrained in a few of the 10 sessions.

When trained and tested on the validation set, the SVM achieved a training accuracy of 99.62% and a testing accuracy of 98.53% [Figures 26-27]. The SVM only seemed to have trouble distinguishing between palm and hook, cyl and hook, and lat and hook. A quick examination of the plots in Figures 21 through 24 shows that some of the aforementioned classes do have overlapping features that may have made discrimination difficult for the classifier. Given that the same classification errors present during the training process were also present during the testing process for the SVM, overtraining of the SVM probably did not occur.

Similar to the SVM, the RNN was trained and tested on the validation dataset for **male 2** and it achieved a training accuracy of 97.69% and a testing accuracy of 95.80% [Figures 28-30]. The RNN primarily had trouble discriminating between the cyl and palm, lat and spher, and tip and palm class combinations during training. During testing, the RNN had trouble discriminating between the hook and cyl, lat and hook, and spher and tip classes. The cyl and hook and tip and spher class combination discrimination problems were largely present in the testing results rather than the training results, again suggesting that the RNN may have been slightly overtrained.

During this experiment there were issues with achieving the high accuracy percentages finally achieved during some of the initial trial and error when developing the SVM and RNN models. The SVM and RNN specifically seemed to have trouble with the imbalance of class samples - as in the case of the collected sEMG data where there were 117 samples for the rest class and at most 20 samples for each of the other classes or the small amount of available training data, such as for the validation set, which had only 30 samples for each of the six classes. The solution found was to oversample the data. The collected data was oversampled, so that there were 117 samples for each of the eight classes including the rest class. **Male 2**, given that the samples were already even, originally the oversampling was taken out of the process. This gave poor results and in return was revalidated implementing an oversampling which brought the 3rd parties data samples from 30 per action to 100. The oversampling process was simply a random sampling of the available data for each class until the desired number of samples for each of the classes had been generated. Once this was implemented the accuracies were greater.

In summation, the goal of achieving 90-95% accuracy for the SVM and RNN classifiers was achieved after the data had been passed through the time and frequency-domain feature extraction techniques to extract the mean, max, variance and RMS of each channel. The training and testing accuracies of both models were comparable, suggesting that both models were minimally overtrained. On closer examination of the SVM, there seemed to be almost no overtraining, while for the RNN there was relatively more overtraining. Both the SVM and RNN models along with their respective preprocessing procedures were validated with a third-party dataset and both produced accuracies exceeding those of the third-party research paper, suggesting that the models were valid. There are many differences in the processes which could have affected these accuracies. One of the main differences is the research paper used all analytic methods, there is not any machine learning. This may have given some better results, but it was stated in the research that the advantage to using a simple linear classifier is that machine learning tuning parameters do not need to be troubleshot [32].

If these experiments were conducted again, more data collected over multiple days should be collected across male and female subjects in order to build more robust classification models. More features should also probably be extracted in order to increase the amount of separability. The RNN classifier seemed to have some trouble with overtraining, so processes should be put in place to protect against this problem. The data should also probably be taken over multiple days in order to increase the robustness of the model.

**7. REFERENCES**

[1] Došen, S., Cipriani, C., Kostić, M., Controzzi, M., Carrozza, M. C., and Popović, D. B. (2010). Cognitive vision system for control of dexterous prosthetic hands: experimental evaluation. J. Neuroeng. Rehabil. 7:42. doi: 10.1186/1743-0003-7-42

[2] Markovic, M., Dosen, S., Cipriani, C., Popovic, D., and Farina, D. (2014). Stereovision and augmented reality for closed-loop control of grasping in hand prostheses. J. Neural Eng. 11:46001. doi: 10.1088/1741-2560/11/4/046001

[3] Atzori, M., Gijsberts, A., Müller, H., and Caputo, B. (2014c). “Classification of hand movements in amputated subjects by sEMG and accelerometers,” in Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), (Chicago, IL), 3545–3549.

[4] Gijsberts, A., Atzori, M., Castellini, C., Muller, H., and Caputo, B. (2014). The movement error rate for evaluation of machine learning methods for sEMG-based hand movement classification. IEEE Trans. Neural Syst. Rehabil. Eng. 22, 735–744. doi: 10.1109/tnsre.2014.2303394

[5] Krasoulis, A., Vijayakumar, S., and Nazarpour, K. (2015). “Evaluation of regression methods for the continuous decoding of finger movement from surface EMG and accelerometry,” in Neural Engineering (NER), 2015 7th International IEEE/EMBS Conference (Montpellier), 631–634.

[6] Park, K., and Lee, S. (2016). “Movement intention decoding based on deep learning for multiuser myoelectric interfaces,” in 2016 4th International Winter Conference on Brain-Computer Interface (BCI), 1–2.

[7] Fukuda, O., Tsuji, T., Kaneko, M., Otsuka, A., and Tsuji, O. F. T. (2003). A human-assisting manipulator teleoperated by EMG signals and arm motions. IEEE Trans. Robot. Autom. 19, 210–222. doi: 10.1109/tra.2003.808873

[8] Li, G., Schultz, A. E., and Kuiken, T. A. (2010). Quantifying pattern recognition-based myoelectric control of multifunctional transradial prostheses. IEEE Trans. Neural Syst. Rehabil. Eng. 18, 185–192. doi: 10.1109/TNSRE.2009.2039619

[9] Tenore, F. V. G., Ramos, A., Fahmy, A., Acharya, S., Etienne-Cummings, R., and Thakor, N. V. (2009). Decoding of individuated finger movements using surface electromyography. IEEE Trans. Biomed. Eng. 56, 1427–1434. doi: 10.1109/TBME.2008.2005485

[10] Atzori, M., Cognolato, M., & Müller, H. (2016). Deep learning with convolutional neural networks applied to electromyography data: a resource for the classification of movements for prosthetic hands. Frontiers in neurorobotics, 10, 9.

[11] Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural network. In Advances in neural information processing systems (pp. 1135-1143).

[12] Allard, U. C., Nougarou, F., Fall, C. L., Giguère, P., Gosselin, C., Laviolette, F., & Gosselin, B. (2016, October). A convolutional neural network for robotic arm guidance using sEMG based frequency-features. In Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on(pp. 2464-2470). IEEE.

[13] B. Hudgins, P. Parker, R. Scott, "A new strategy for multifunction myoelectric control", IEEE Transaction on Biomedical Engineering, vol. 40, no. 1, pp. 82-94, 1993.

[14] T. Farrell, R. Weir, "The optimal controller delay for myoelectric prostheses", IEEE Transaction on Neural systems and rehabilitation engineering, vol. 15, no. 1, pp. 111-118, 2007.

[15] B. Peerdeman, D. Boere, H. Witteveen, R. H. in ‘t Veld, H. Hermens, S. Stramigioli, H. Rietman, P. Veltink, S. Misra, "Myoelectric forearm prostheses: State of the art from a user-centered perspective", Journal of Rehabilitation Research & Development, vol. 48, no. 6, pp. 719-738, 2011.

[16] Yao, X. (1999). Evolving artificial neural networks. Proceedings of the IEEE, 87(9), 1423-1447.

[17] “Convolutional Neural Network.” Unsupervised Feature Learning and Deep Learning Tutorial, ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/.

[18] N.F. Guler, S. Koçer, Classification of EMG signals using PCA and FFT, J. Neurosci. Methods, Vol. 29, Iss. 3, 2005, 241-250

[19] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discov., Vol. 2, 1998, 121-167

[20] G. Horvath, J. Suykens, S. Basu, Advances in Learning Theory: Methods Models And Applications, 2003, IOS Press, Amsterdam

[21] Chen, X., & Wang, Z. J. (2013). Pattern recognition of number gestures based on a wireless surface EMG system. Biomedical Signal Processing and Control, 8(2), 184-192. 10.1016/j.bspc.2012.08.005

[22] Stango, A., Negro, F., & Farina, D. (2015). Spatial correlation of high density EMG signals provides features robust to electrode number and shift in pattern recognition for myocontrol.IEEE Transactions on Neural Systems and Rehabilitation Engineering, 23(2), 189-198. 10.1109/TNSRE.2014.2366752

[23] M.A. Oskoei, H. Hu  
Support vector machine-based classification scheme for myoelectric control applied to upper limb  
IEEE Trans. Biomed. Eng., 55 (2008), pp. 1956-1965

[24] P. Kaufmann, K. Englehart, M. PlatznerFluctuating EMG signals: investigating long-term effects of pattern matching algorithms

Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Buenos Aires(2010), pp. 6357-6360

[25] Kamali, T., Boostani, R., & Parsaei, H. (2014). A multi-classifier approach to MUAP classification for diagnosis of neuromuscular disorders. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 22(1), 191-200. 10.1109/TNSRE.2013.2291322

[26] Xu, X., Zhang, Y., Luo, Y., & Chen, D. (2013). Robust bio-signal based control of an intelligent wheelchair. Robotics, 2(4), 187-197. 10.3390/robotics2040187

[27] C. Hsu, C. Lin, "A comparison of methods for multi-class support vector machines", IEEE Trans. Neural Netw., vol. 13, no. 2, pp. 415-425, 2002.

[28] Agarap, A. F. (2017). An Architecture Combining Convolutional Neural Network (RNN) and Support Vector Machine (SVM) for Image Classification. *arXiv preprint arXiv:1712.03541*.

[29] Tidwell, R., Akumalla, S., Karlaputi, S., Akl, R., Kavi, K., & Struble, D. (2013, July). Evaluating the feasibility of EMG and bend sensors for classifying hand gestures. In Proceedings of the International Conference on Multimedia and Human Computer Interaction (Vol. 63, pp. 1-8).

[30] Cannan J, Hu H. Automatic user identification by using forearm biometrics. IEEE; 2013.

[31] Oliver, A. S., Maheswari, N., & Samraj, A. (2014). Stable and critical gesture recognition in children and pregnant women by SVM classification with FFT features of signals from wearable attires. Research Journal of Applied Sciences, Engineering and Technology, 7(23), 4917-4926.

[32] C. Sapsanis, G. Georgoulas, A. Tzes, D. Lymberopoulos, Improving EMG based classification of basic hand movements using EMD in 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society 13 (EMBC 13), July 3-7, pp. 5754 - 5757, 2013.

[33] Pineda, F. J. (1987). Generalization of back-propagation to recurrent neural networks. *Physical review letters*, *59*(19), 2229.