The Application of Machine Learning to the Categorization of DASHR Head Impact Data

Sarah Glomski

Advisor: Jason F. Luck, Ph.D.

Duke University

Department of Biomedical Engineering

May 3, 2024

Table of Contents

I. *Background*

Quantifying Head Injury in Sports

Football is the most common organized sport to produce head injuries, making up 12.8% of all sports-related head injuries and 19.4% of all sports-related concussions [1]. A rise in the interest of sports-related concussion emerged from early neuropsychological studies, which found links between head impact exposure and long term cognitive function [2] [3]. It is now known that concussions exist on a spectrum [4], and that repeated subconcussive events may lead to long term neurodegeneration (CTE) [5] [6] [7] [8]. In response to this research, many governmental, academic, and private organizations have allocated resources to improving diagnostics [9] [10] [11], treatment [12] [13] [14], and legislation [15] [16] [17] in the area of sports-related brain injury.

Though the exact linear and angular acceleration quantities needed to cause concussive and subconcussive impacts are not agreed upon, several studies have aimed to quantify them. For example, by reconstructing video-taped NFL concussions with a Hybrid III dummy, a mean peak linear acceleration of 98 ± 28 g among concussed players was able to be discerned, with a half-sine duration of 15 ms [18]. A peak linear acceleration of 70-75 g was also suggested to delineate the concussions threshold for padded impacts [18]. Using the HIT System to create a large database of concussive (n=57) and subconcussive (n=300,977) events in football players, mean rotational features were also able to be discerned [19]. The average concussive impact resulted in a rotational acceleration of 5022 rad/s^2 and rotational velocity of 22.3 rad/s, while the average subconcussive impact yielded a rotational acceleration of 1230 rad/s^2 and rotational velocity of 5.5 rad/s [19]. Based on this data, a 50% concussion risk criterion was determined to be a rotational acceleration of 6383 rad/s^2 and a rotational velocity of 28.3 rad/s [19]. These injury risk values are useful for creating head injury risk curves [19], which help to characterize concussive and subconcussive impacts and inform safer design decisions. However, these values can easily be misinterpreted as an all-or-nothing threshold. In reality, the probability of head injury is non-deterministic in nature, but has been best quantified with a logarithmic spectrum.

The heightened interest in this field has led to the development of several kinematic devices that record head impact data. These include devices fastened to a helmet or headband, cast within a mouthguard or earpiece, or attached directly to the skin [20]. It is important to consider the coupling mechanism of the device to the head and to consider how accurately this represents the kinematic loading conditions of the brain, which can be approximated as the center of gravity of the head [21]. It is important that this data be cautiously analyzed and interpreted in a biomechanical context given the effect that published research can have on public safety in sports.

The Data Acquisition System for Head Response (DASHR) is an earpiece device that couples relatively well with the ear canal [22]. As with other devices, it measures linear acceleration and

angular velocity with an accelerometer and gyroscope [23]. However, rather than discarding data that does not meet a preset linear acceleration threshold, all data is stored in the DASHR for analysis. Although this eliminates the possibility for real-time impact analysis, the completeness of the kinematic data allows for various post-processing methods to be explored. This includes algorithmic and machine learning techniques, which could eventually allow for a real-time classifier like the Stanford mouthguard [24].

As with other sensors, the DASHR is prone to spikes in measured acceleration and rotation when it is handled outside the ear, re-seated in the ear canal, or even due to random glitches in the hardware [25]. These spikes in the vector quantities can masquerade as "impacts" in the kinematic data, which can lead to the over-reporting of head injury data and improper characterization of head impact biomechanics. One approach to combat this discrepancy is to cross-validate the wearables data by manually labeling the data obtained from the DASHRs in the players' ears during practice. However, this process is extremely time-consuming, as it requires researchers to watch many hours of video footage to label the data properly [26]. Rather than repeating this task to process data from every athletic exposure, it would be useful to develop a method to accurately classify the data and remove false positive impacts in a time-efficient manner.

Thresholding Algorithms

Thresholding is a common technique for excluding data that is likely erroneous [25][27]. In theory, if the natural distribution of the data is well understood, thresholds for improbable data could be set with reasonable confidence. Oftentimes, however, these thresholds are set based on an incomplete understanding of head injury biomechanics, which can lead to bias in the dataset [27]. In the context of head injury, many wearable sensors have a linear acceleration threshold that must be exceeded before data collection can occur. These linear acceleration thresholds (LATs) are not agreed upon in the literature, but are often set to 10 Gs [25][27] as a lower bound for what is considered a head impact.

By adding these lower bound thresholds, some of the low-g data is excluded from the dataset. To what extent this is valid depends on the goal of the data analysis. For the purposes of quantifying head injury accurately, it is important to consider how many of the excluded data are actually head impacts (ie. false negatives), versus how many of the included data are still non-impacts (ie. false positives) [28].

The true positive rate (TPR) and false positive rate (FPR) are metrics used to describe the efficacy of a classifier, which can be as simple as a thresholding algorithm. Their mathematical definitions are shown below, where TP is the number of true positives, TN is true negatives, FP is false negatives, and FN is false negatives. In the context of head injury, the FPR is of great concern.

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

It has been established that high-g non-impacts (caused by glitches or touching the device, for example) generally have lower peak durations than true impacts, and oftentimes have higher peak magnitudes [29]. Thus, if these high-g non-impacts are falsely included in the characterization of head impacts (which contributes to the FPR), there will not only be a higher number of recorded impacts, but also a higher average magnitude per impact. Therefore, to achieve an accurate representation of the typical head impact, it is important to try to minimize the FPR with the classifiers that are being developed.

Given the simplicity of thresholding scripts, it is unlikely that these classifiers will ever be perfect (ie. a TPR of 100% and an FPR of 0%). In an attempt to further minimize the FPR, more complex techniques can be applied to this classification task.

Machine Learning

Machine learning models have been developed to classify head impact data from football players [24] [26] [25]. These models were trained on kinematic data from other wearables, such as the Stanford Mouthguard. The purpose of this research is to develop a similar method for pre-processing raw DASHR data and feeding it into a machine learning model to distinguish between valid and invalid head impact data. Valid impact data is defined to be a true head impact, whereas invalid data includes high-G non-impact data that would arise from turning the DASHR on/off or situating/adjusting the DASHR in the ear. Other common activities such as running/walking and standing still have been designated their own output classes so as to aid in the overall labeling process.

Recent research has been done in the broader area of human activity tracking using advanced models [30][31][32]. One example is the 1D CNN-LSTM, which combines the feature recognition of a Convolutional Neural Network (CNN) with the longitudinal strengths of a Long Short Term Memory model (LSTM). With a fully-labeled set of practice data, more advanced models can be developed to find long-term trends that relate head impacts to behavioral activities. These models could provide insight into how certain variables, such as the player position, play called, and activity performed all lead into the risk of head impact and injury.

Machine learning relies on the existence of extensive datasets which allow the model to find subtle trends over time and revise its accuracy through an iterative process. A robust dataset is therefore of ultimate importance in training and testing machine learning models, and must be sufficiently representative of the target population. Any level of bias and skewing could lead to overfitting of the model to the training set, which limits the model's generalizability and

accuracy beyond the training set. When testing the model, a higher accuracy is clearly preferable, but a model with 100% accuracy would not be ideal as it would almost certainly be overfit to the training set, and would therefore have limited generalizability.

Another way of examining the extent to which a model is skewed by the dataset is to create a randomized model, which is trained on a training set with randomized label conditions, then tested on a test set with correct labels. For a model with a balanced dataset, the accuracy should statistically be around 25% (or 1 in 4 for the 4 classes). Datasets that yield a higher performance with randomized labels are inherently flawed in that they are biased to perform better, and so their reported accuracy from a fully-trained model will be artificially inflated.

DASHR Impact Classifier

Previous work has been made to develop a CNN that classifies DASHR data into the 4 classes: valid head impact, high-g non-impact, running/walking, and standing still [28]. The model was trained and tested using simulated data, as explained later in the methods section. Simulated data has been used in the literature to train models as a proof of concept prior to introducing field data, which is oftentimes more complex [24]. The reported accuracy of Liu's model was 84.6% ± 2.0% (n=10), which appears to be a decent level of accuracy, and is not high enough to indicate obvious overfitting. However, when trained on a randomized training set, the model achieved an accuracy of 62.3% ± 0.2%, which is well above the expected 25% for a perfectly balanced dataset. Part of this inflated performance is likely due to the fact that the dataset is heavily skewed to include more valid head impacts than any other condition, as shown in Figure 1.

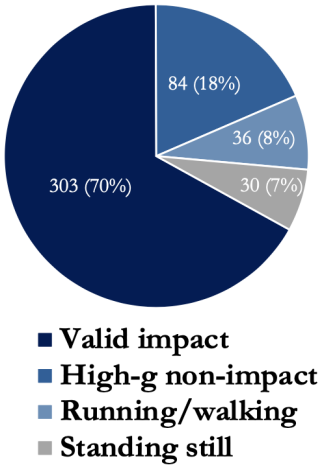

Figure 1. Distribution of the simulated dataset across the 4 classes.

With an inherently biased dataset, the reported accuracy is no longer an objective metric for evaluating the performance of the fully-trained CNN. A more in-depth analysis is required to determine how the model is achieving this level of accuracy, and whether the features it chooses to focus on during its predictions are correct or irrelevant to the context of this problem. This

involves opening the black box of machine learning to visualize results in an easily-interpretable manner for the human.

Opening the Black Box of Machine Learning

Since the creation of machine learning techniques, there has been significant interest in the interpretability of the results. Because machine learning is often intended to surpass human classifying capabilities, it provides an opportunity for researchers to learn from the model rather than the other way around. However, there exists a balance between the interpretability of the results and the faithfulness to the model's complexity. Viewing a multi-dimensional matrix of tensors is not easily interpretable, while viewing only the final class predictions is unfaithful to the model's complexity. A line must be drawn defining how much explanation is required for humans to make sense of the results, and many attempts at such a method have been made.

Selvaraju et al. (2017) introduced Gradient-weighted Class Activation Mapping (Grad-CAM) as an effective way to visualize CNN predictions in the context of input data, allowing a peek into the black box of machine learning [33]. Grad-CAM is a useful technique for identifying bias within a dataset and failure modes within a model. Although the technique was originally developed for 2D CNN models (image classifiers), the same principle can be applied to 1D CNN models (time-series classifiers).

Grad-CAM relies on the convolutional feature maps to provide insight into which features were most prominent in the time-series input and which features were used for predicting the output class. By generating a heatmap of relative importance of the input data, the prediction process can be more easily interpreted by the human.

Another topic of debate is the trustworthiness of the model's decision-making process. If there is no easily-interpretable explanation for why a model makes a certain prediction, does this discount the model's results? Selvaraju et al. (2017) discuss the value of Grad-CAM in the context of increasing the user's trust in the model to make the correct image classification based on the explanation it gives [33]. This, however, relies on the expertise of the user in his/her own visual classification technique, or in other words, having full confidence that he/she knows the ground truth label of the image. With less intuitive classification tasks like temporal kinematic data, it is important to think critically about the interpretability of the results, and whether this should inform the validity of the model's results.

II. *Methodology*

Simulated Data Collection

Liu had previously created a simulated dataset to serve as a preliminary CNN training set [28]. Data was simulated for each of the 4 conditions listed below:

Table 1. Classifications for CNN dataset.	
Class Number	Activity
0	High-g non-impacts
1	Valid head impacts
2	Standing still
3	Running/walking

To simulate the valid impact data, post-mortem human surrogate (PMHS) heads were used in drop tests inside lacrosse helmets. The drop tests were performed on 2 PMHS for 7 common impact locations and 3 drop heights, as shown in Table 2. 4 repeat trials were performed for each combination of impact location and drop height. 2 DASHRs were placed in each earpiece and set to record linear acceleration and rotational velocity at a sampling frequency of 1000 Hz.

<table>
<tr><td colspan="2">Table 2. Drop test conditions for helmeted PMHS heads.</td></tr>
<tr><td>Impact Location</td><td>Drop Height</td></tr>
<tr><td>Facemask</td><td rowspan="3">8 cm</td></tr>
<tr><td>Frontal</td></tr>
<tr><td rowspan="2">Frontal oblique right</td></tr>
<tr><td>50 cm</td></tr>
</table>

Parietal left	
Parietal right	
Occipital	90 cm
Vertex	

Invalid impacts, which include classes 0, 2, and 3, were simulated by a study author performing activities with the DASHR in a controlled environment. Running/walking and standing still data were simulated while wearing the DASHR. High-g non-impact data consisted of flicking, pressing, and re-situating the DASHR in the ear while either standing still or walking to simulate background noise.

Labeled Field Data

High school football practices were attended and DASHRs were distributed to the appropriate players. Throughout the practice, a handful of players were watched at any given time and their activities were recorded and timestamped.

The DASHR data was downloaded and converted so that it was prepared for analysis. Timestamps were aligned by inputting the start time when the DASHRs were turned on, and performing a visual verification that the resultant accelerometer trend matched the expected data given the time stamped activities. The resulting dataset had the distribution shown in Figure 2.

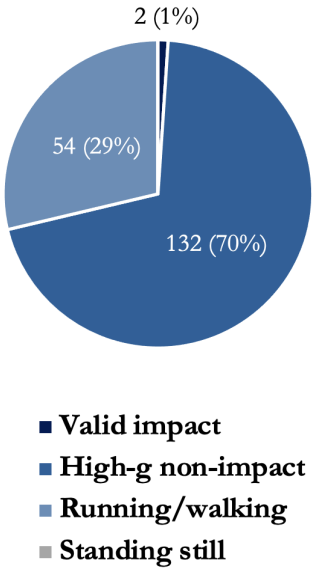

Figure 2. Distribution of the field dataset across the 4 classes.

Preprocessing Techniques

Raw DASHR data exists in the form of binary files, which must first be converted to MATLAB workspace variables. Once imported into MATLAB, the raw, continuous data can be plotted and examined using *plotfullfig*. MATLAB scripts were used to set linear acceleration thresholds, which isolated segments of data in which the threshold was met or exceeded. The duration of the impact was determined, and also had a threshold. Temperature and light were also checked to determine whether the DASHR had a high chance of being coupled to the ear.

Using an app developed by a Duke alumni, subsets of the DASHR data were extracted and placed into individual workspace variables–one for each impact. The option to specify linear acceleration threshold and linear acceleration spike duration allow for specific types of events to be isolated.

Since 10 Gs is a common linear acceleration trigger set on triggered kinematic recording devices, this was selected as the linear acceleration threshold (LAT) for impact data and high-G non-impact data. This LAT was chosen so that all isolated events, whether true impacts or false positives, would simulate those that are blindly extracted from practice data. This was done to take advantage of the fact that thresholding will likely be done prior to feeding real test data into the model. The data stored in the dataset was cropped such that 2000 ms before the peak in linear acceleration and 100 ms after the peak were included in the dataset.

Since it is known that impacts last a relatively long time, and many flukes in data collection occur as spikes over short periods of time, a linear acceleration spike duration threshold was set to 3 ms. It is important to keep thresholding procedures consistent between the training set data

and the standard preprocessing procedure so that the training set accurately represents the data that will be fed into the model during testing.

Basic Thresholding Algorithms

5 basic thresholding algorithms were created to attempt a binary classification task between Class 0 (high-g non-impacts) and Class 1 (valid head impacts). These algorithms were developed using either an upper bound or lower bound for peak linear acceleration, peak duration, or a combination of these characteristics. The algorithms were tested on both simulated and field data to determine if there were differences in performance between these two datasets.

For each thresholding algorithm, the threshold was varied to iterate over a sufficient range of values, and the thresholder accuracy was calculated for each value to determine which threshold is optimal. These optimal thresholds, along with the range of values tested, are shown for each thresholding algorithm in Table 3.

Table 3. Parameters and optimal thresholds for each of the 5 thresholding algorithms developed and tested on simulated and field data.

Number	Thresholder	Tested Value Range	Optimal Threshold Value (Simulated Data)	Optimal Threshold Value (Field Data)
1	Lower bound linear acceleration threshold (LB LAT)	10 Gs - 350 Gs	12.7 Gs	154.8 Gs
2	Upper bound linear acceleration threshold (UB LAT)	10 Gs - 350 Gs	158.2 Gs	10.0 Gs
3	Lower bound duration threshold (LB DT)	2 ms - 100 ms	7.0 Gs	15.0 ms

4	Combined LB LAT and LB DT	10 Gs - 350 Gs, 2 ms - 100 ms	12.1 Gs, 11.0 ms	12.1 Gs, 96.0 ms
5	Combined UB LAT and LB DT	10 Gs - 350 Gs, 2 ms - 100 ms	159.6 Gs, 7.2 ms	10.0 Gs, 3.0 ms

In the case of the binary classification task, these classifiers will produce 1 of possible 4 outcomes per input. These outcomes are shown in Table 4 below.

Table 4. Binary classification outcomes, where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

		Predicted	
		High-g non-impact	Valid head impact
Actual	High-g non-impact	TN	FP
	Valid head impact	FN	TP

The 5 classifiers were scored according to overall accuracy, which is defined below, where TP is the number of true positives, TN is true negatives, FP is false positives, and FN is false negatives.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

It is important to note that overall accuracy is not a complete measure of a classifier's efficacy. Receiver operating characteristic (ROC) curves were also generated to visualize the relationship between the TPR and FPR for each classifier. A perfect classifier will have 100% TPR and 0% FPR, but this is not realistic given the dataset. In practice, the best classifiers are those that show a steep rise in TPR for initial increases in FPR, followed by a plateau in TPR for subsequent increases in FPR. The area under the curve (AUC) for each ROC curve provides insight into how well the ROC curve follows the ideal shape, with a perfect ROC curve achieving an AUC of 1. An AUC of 0.5 represents a non-discriminating classifier, which has a linear ROC curve and cannot effectively classify between the two groups.

Grad-CAM

The principle of operation behind Grad-CAM is as follows:

1. A CNN is fully trained on a dataset of time-series data. The CNN is composed of several layers, including an input layer, 2 convolutional layers, a dropout layer, a max pooling layer, a flattening layer, and 2 dense layers. It accepts a series of linear acceleration and rotational velocity data (stacked) and outputs a single classification between classes 0 and 3.
2. A separate model is constructed, copying the fully-trained input layer and 2 convolutional layers from the original CNN. This model is called the Last Convolutional Layer model because it accepts the same input as the CNN, but outputs several convolutional feature maps that indicate which features are strongest in the input data, and where they occur. The number of convolutional feature maps, *f*, determines the number of features the model can look for.
3. The same time-series input is fed into both the CNN and the Last Convolutional Layer model. Both the predicted class and the convolutional feature maps are recorded while a function, tf.GradientTape, records extra information on the trainable backend variables for further analysis.
4. A function, tf.gradient, uses the information stored by GradientTape to calculate the gradient, or derivative, of the predicted class tensor with respect to the convolutional feature maps.
5. A weight for each channel of the gradient is calculated using the tf.reduce_mean function to take the global average pooling (GAP) for each convolutional feature map. Then, ReLU activation is used to determine which features were most relevant to the prediction process.
6. The gradient data is then normalized and reshaped such that it provides a heatmap of relative importance to the prediction process. If the input data is *t* time points in length and each convolutional layer has a kernel size *k*, the length of the heatmap will be *t-2*(k-1)*. This is slightly shorter than the input data because data is lost in a convolution on the edges of the input. The heatmap can be viewed by itself as a relative indicator of relevance over time, or it can be layered over the input data to provide specific insight into which features were relevant in the prediction process.

The following snippet of code was used to visualize the heatmap of relevance over time.

```
def grad_cam(last_conv_model, input_data, class_index,
print_grads=False):

 # Compute the gradients of the class score with respect to the
feature maps

 with tf.GradientTape() as tape:
```

```
    # tf will now start recording tensor values for automatic
differentiation
    last_conv_output, preds = last_conv_model(input_data) # shapes
(1, t-2*(k-1), f), (4,)
    class_output = preds[:, class_index] # shape (1,)
 if class_index != np.argmax(preds):
   print('Error: Prediction class index is wrong.')
   print(class_index, np.argmax(preds))
 grads = tape.gradient(class_output, last_conv_output)
 if print_grads:
   print(grads)
 # Apply global average pooling to the gradients
 pooled_grads = tf.reduce_mean(grads, axis=(0, 1))
 last_conv_output = last_conv_output[0]
 heatmap = last_conv_output @ pooled_grads[..., tf.newaxis]
 heatmap = tf.squeeze(heatmap)
 heatmap = tf.maximum(heatmap, 0)/tf.math.reduce_max(heatmap)
 return heatmap.numpy()
```

Grad-CAM often attempts to derive gradients and ends up with all 0s or all nans. This is due to the numerical instability of the gradient calculation, which utilizes the function tf.norm() to normalize the data. When tf.norm() attempts to divide 0 by 0, it returns nan instead of 1. Thus, several of the gradients were not calculable and were therefore excluded from the analysis.

An early hypothesis for the non-real gradients was that they occurred when the model did not particularly show a strong preference towards the predicted class relative to the other classes. However, this is likely not the case due to the high occurrence rate of this issue across all classes, and regardless of prediction correctness or model accuracy. There does seem to be high variability in the occurrence of this issue from model to model. In some models, nearly every gradient is non-real, whereas in others, all gradients are real. There does not appear to be a

pattern to the non-real gradient problem, however it should be examined more thoroughly with the implementation of stabilized gradient calculation functions.

Experimental Setup

There are two main questions that will be tested with the use of Grad-CAM on the current CNN dataset.

Experiment 1: How does cropping the input to exclude/include certain features affect the overall accuracy of the model?

In the GWDD report by Patrick Liu, it is stated that the time-series input is 2100 ms in length, which would include the full length of the data stored in the training set [28]. However, certain features of the simulated data make it highly predictable, such as the long pause in movement followed by the small uptick in rotational velocity that can be seen in the helmeted drop tests used for the valid impact condition, as shown in Figure 3.

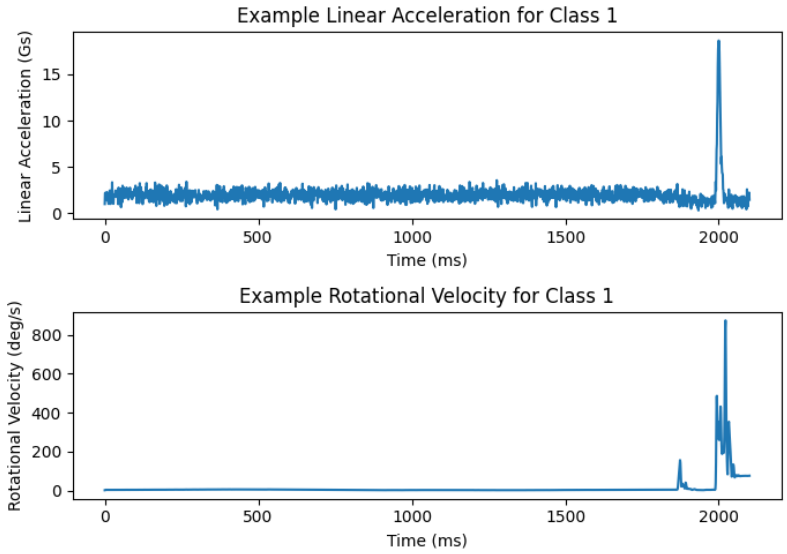

Figure 3. Example plot of linear acceleration and rotational velocity over 2100 ms for Class 1.

These repetitive, artificial features are a result of the testing environment and pre-processing techniques used, and are not representative of the target environment. It is hypothesized that cropping out these features will reduce the accuracy of the model, indicating that the model was focusing on features that are irrelevant to the target population.

Liu previously explored the idea of cropping the 2100 ms dataset to only include that last 100 ms, as shown in Figure 4 [28]. This would, in theory, eliminate the repetitive features described above. However, because the pre-processing technique crops every input to have the peak acceleration occur at 2000 ms, the first half of the peak is cropped out in this scenario. Plus, with two convolutional layers that each reduce the length of the time-series data by *k-1*, this cropping

technique would eliminate some seemingly very important features for classification. Thus, it is hypothesized that this version of the dataset will result in a low performance due to the loss of pertinent information.

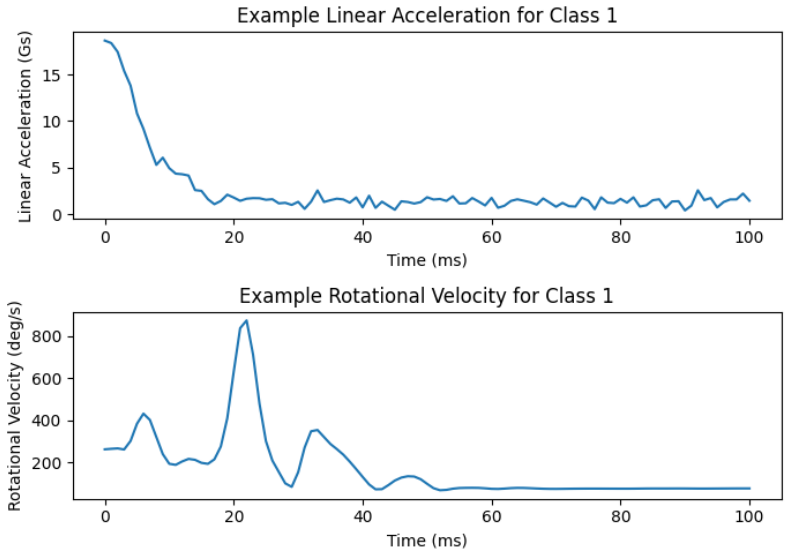

Figure 4. Example plot of linear acceleration and rotational velocity over 100 ms for Class 1.

The idea of centering and zooming in on the data is then explored, with 100 ms and 50 ms subsets of the data centered around the peak accelerations. An 2100 ms example of a high-g non-impact is shown in Figure 5 below. Some of the high-g non-impact samples have long peak durations that last more than 50 ms, as shown in Figure 6 for the 50 ms cropped and centered condition, which crops out the latter end of the peak. It is hypothesized that without a full view of this information, the model will perform worse, and so the accuracy of the 50 ms model will be lower than that of the 100 ms model.

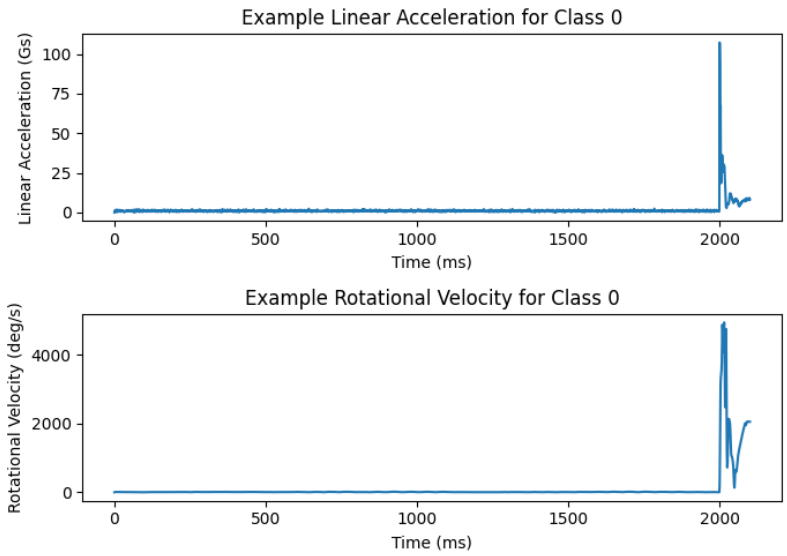

Figure 5. Example plot of linear acceleration and rotational velocity over 2100 ms for Class 0.

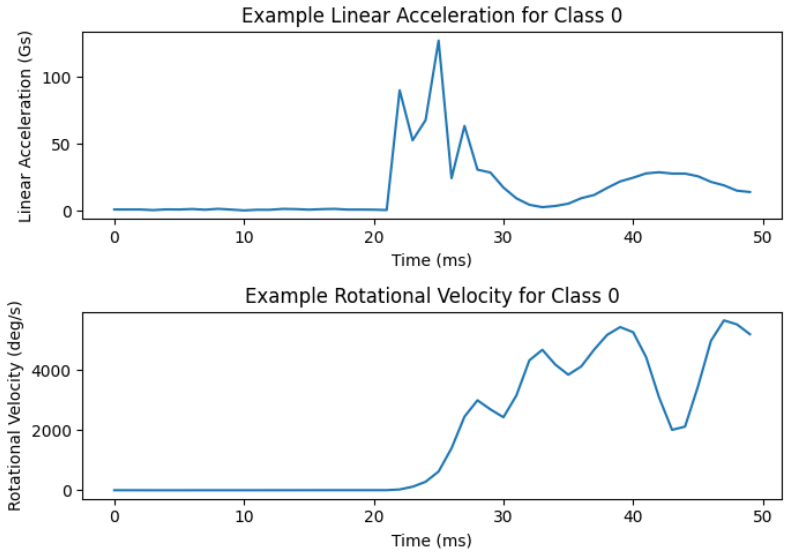

Figure 6. Example plot of linear acceleration and rotational velocity over 50 ms (centered) for Class 0.

The running/walking data is often periodic in shape, with a period of ~350 ms, as shown below in Figure 7. Figure 8 shows a 100 ms cropping of this same input, where the periodic nature is no longer seen. Because these periods are longer than the cropped input lengths, it is hypothesized that the accuracy of the model will be lower with respect to class 3 as the length of the input decreases.

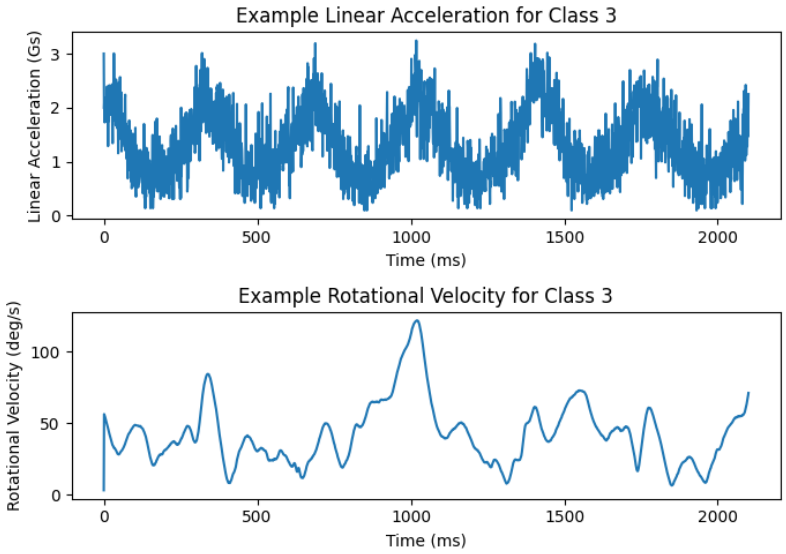

Figure 7. Example plot of linear acceleration and rotational velocity over 2100 ms for Class 3.

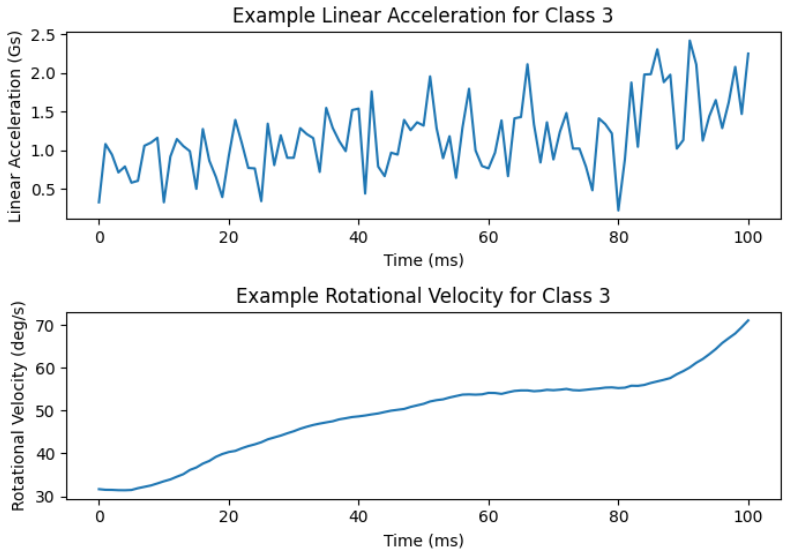

Figure 8. Example plot of linear acceleration and rotational velocity over 100 ms for Class 3.

Class 2 is generally composed of random noise, as shown in Figure 9. Cropping this to 100 ms yields an input similar to Figure 10, where the linear acceleration still looks random, but the rotational velocity appears to have a slight trend that the model may mistake for the feature of another class.

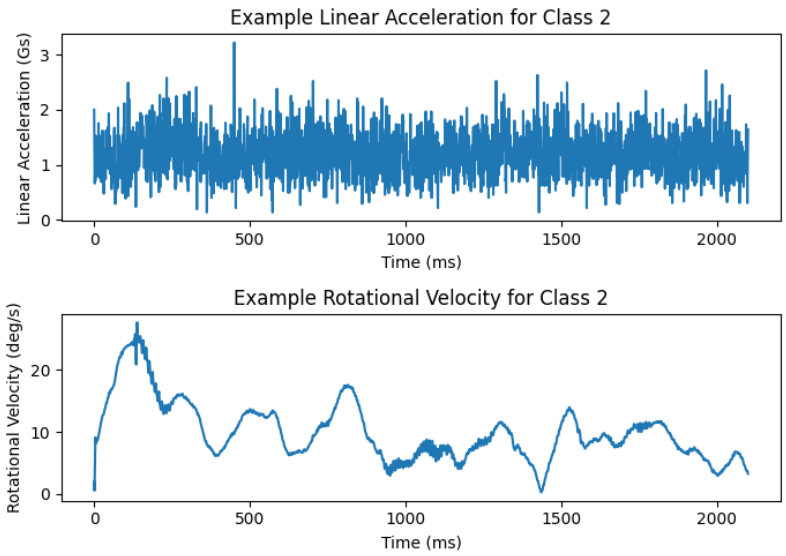

Figure 9. Example plot of linear acceleration and rotational velocity over 2100 ms for Class 2.

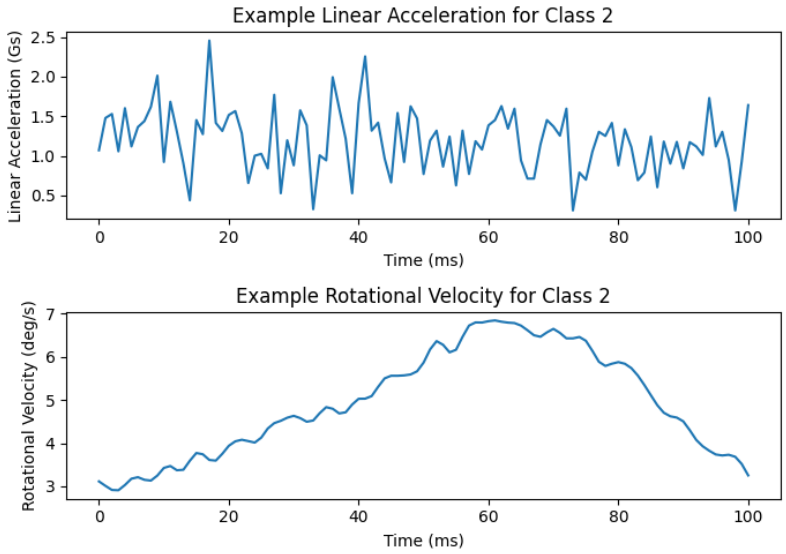

Figure 10. Example plot of linear acceleration and rotational velocity over 100 ms for Class 2.

Experiment 2: How do the kernel size and number of convolutional feature maps affect the accuracy of the model?

After determining the optimal input data length, the kernel size will be varied from a standard size of 5 to a large size of 11, as previously explored by Liu [28]. Generally, larger kernel sizes are used for identifying features with longer periods, and are worse at identifying short features. The number of feature maps outputted by the convolutional layers of the CNN were also varied from a standard size of 128 to a large size of 256. More feature maps generally means that the model can scan for more features, but comes at the cost of higher computational requirements.

Because the dataset is composed of both short features (ie. a head impact) and long features (ie. running/walking), the size of the kernel may affect which features are more easily identified. Varying the number of feature maps is generally done to see whether the model needs more complexity to achieve better accuracy.

III. *Results*

Classifier Accuracy

Figure 11 shows the resulting accuracy of each classifier using the optimized threshold values for both simulated and field data. Both the thresholding scripts and the ML model results are included, which illustrates the relative efficacy of each. Among the simulated data, the classifier with the highest overall accuracy was the combined LB LAT and LB DT (LAT=12.1 Gs, DT=11.0 ms), with an accuracy of 92.5%. This outperformed the ML model, which achieved an accuracy of 89.5%.

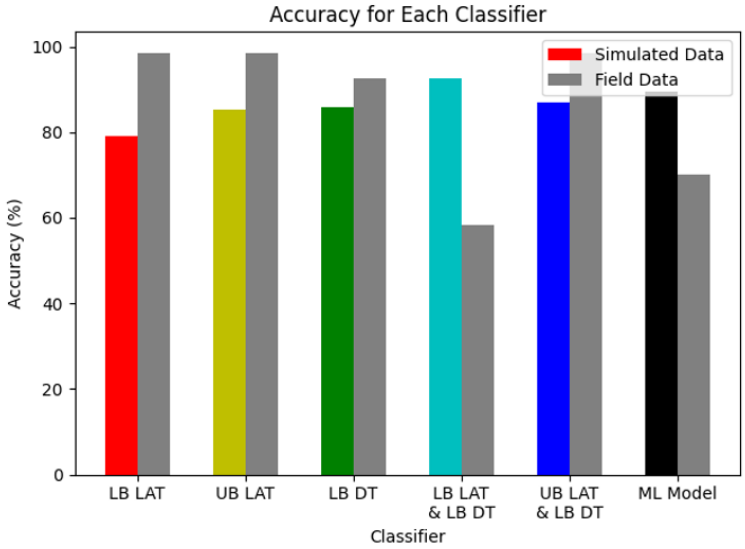

Figure 11. Accuracy for each classifier for simulated and field data.

Comparing the results for simulated and field data, there is not an apparent trend in the data. In some cases, accuracy is higher for the field data (ie. for the LB LAT and LB DT separately), but in others this trend is flipped (ie. the combined LB LAT and LB DT). In the case of the LB LAT classifier, an overall accuracy of 98.5% was achieved, which is seemingly great. However, Table 3 shows that 154.8 Gs was selected as the optimal threshold for field data, which is much higher than the 12.7 Gs selected for the simulated data. The distribution of the field data is such that the true head impacts made up only 1.06% of the overall dataset, as opposed to making up 66.9% of the simulated dataset. This results in a skewed accuracy value, as the size of the classes are not accounted for. When the threshold is blindly varied over a range of values in cases such as this, it can result in an optimal threshold that performs well in terms of theoretical accuracy, but would not perform as well with a balanced dataset. This phenomenon illustrates how the uneven distribution of the field dataset likely skewed the results of the classifiers. Therefore, the field data was excluded from further analysis, and the simulated data was analyzed as a proof of concept for what could be done once the field data is more balanced.

ROC Curves

Further analysis was performed on the thresholding algorithms to determine whether the overall accuracy values were indicative of good classifier performance. Figure 12 shows ROC curves that were plotted for each of the 5 thresholding algorithms using simulated data. The AUC is shown in the legend for each classifier.

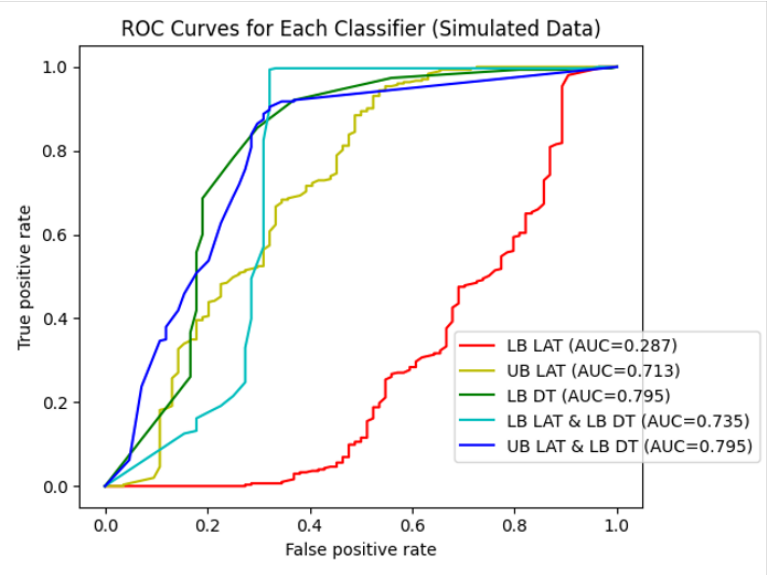

Figure 12. ROC curve and AUC for each classifier for simulated data.

The two thresholding algorithms that tied with the highest AUC were the LB DT and the combined UB LAT and LB DT with an AUC of 0.795. The combined LB LAT and LB DT performed slightly worse, with an AUC of 0.735, despite having the highest overall accuracy of the thresholding algorithms.

Interestingly, the LB LAT and UB LAT had inverted ROC curves, which makes sense given that the directionality of the threshold was all that changed between the two algorithms. The UB LAT achieved an AUC of 0.713, which is the complement of the LB LAT's AUC of 0.287. This illustrates that the UB LAT was much better at completing the binary classification task than the LB LAT, despite only having an accuracy that was 6.2% higher (85.27% vs 79.07%).

ML Experiments with Grad-CAM

Experiment 1

The effect of varying the input data length can be seen in Figure 13 from the mean accuracy across 10 models. The hypothesized trends were seen to hold true for the experiment.

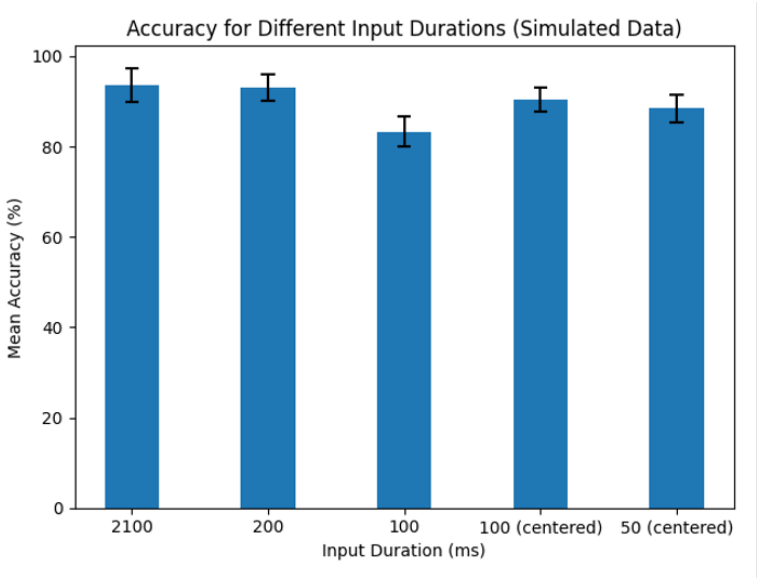

Figure 13. ML model accuracy for different input data lengths (k=5, f=128, n=10).

A one-way ANOVA revealed the accuracy of the 100 ms group was significantly different from both the 200 ms and 2100 ms groups ($p<0.0001$ and $p<0.0001$, respectively; see Appendix A for statistics). These results show that decreasing the length of the input alone significantly hindered the model's performance, assuming the full feature was still within the cropping window.

When the main peak was cut off from the cropping window, the model performed significantly worse. The 100 ms group was statistically different from both the 100 ms (centered) and 50 ms (centered) groups ($p=0.0001$ and $p=0.0058$, respectively). This shows that the 100 ms group must have lost pertinent information due to cropping, and that by re-centering the window around the peak acceleration, the model was more easily able to distinguish between classes. However, the 100 ms (centered) and 50 ms (centered) groups were not significantly different, indicating that cropping closer to the peak acceleration did not significantly hinder the accuracy of the model. These results indicate that the model is likely focusing on the peaks of the high-g non-impact samples rather than the features after the peak.

To investigate these trends, Grad-CAM was used to examine which features the model was focusing on. Example heatmaps were generated for each class under each input condition, indicating the relevance of each time point in determining the predicted classification. To understand the dataset better, input data was averaged over all test cases from 10 randomly generated models, each with randomly partitioned training/test sets. Relevance was also averaged over these tests for each class. These average relevance over time graphs are shown for each class.

A few key trends in Experiment 1 were visualized with Grad-CAM. First, the suspicion that the model was focusing on the long pause in movement for Class 1 was confirmed, as shown in Figure 14. The model also sometimes focused on the small uptick in rotational velocity (which

occurred due to breaking the string), which corresponds with a drop in relevance after this point in Figure 15. However, this was not as common as the model focusing on the larger peak to determine the classification. It can also be seen in Figure 15 that the model rarely misclassified Class 1 as any other classes, as shown by the relatively small relevance traces for all other classes.

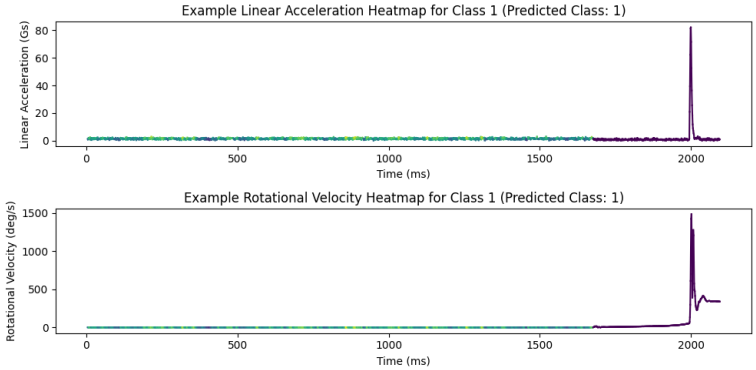

Figure 14. Example heatmap over 2100 ms for Class 1.

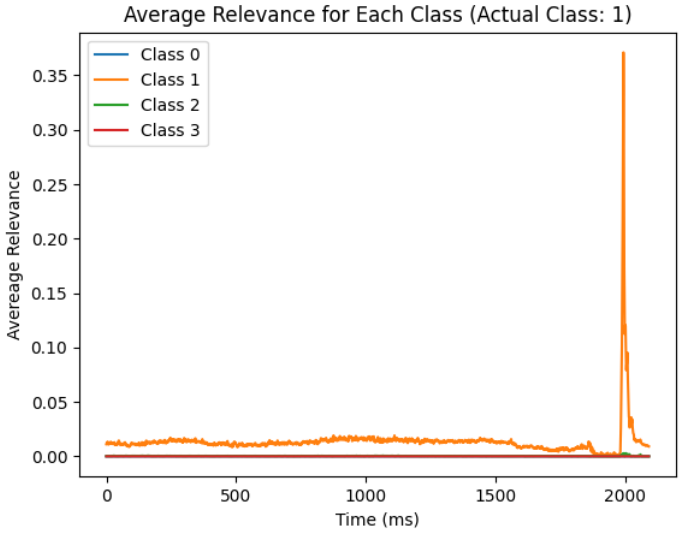

Figure 15. Average relevance over 2100 ms for Class 1.

When the input data is shortened to 200 ms, the model can no longer focus on as much of the long pause in movement, but it still focuses a bit on the early lack of movement, as shown by the small peak in relevance in Figure 17. As shown in Figure 16, the model focuses more on the latter parts of the peak when compared to Figure 14. The slightly worse performance of the model can be seen by the larger relevance traces for classes 0 and 2 in Figure 17.

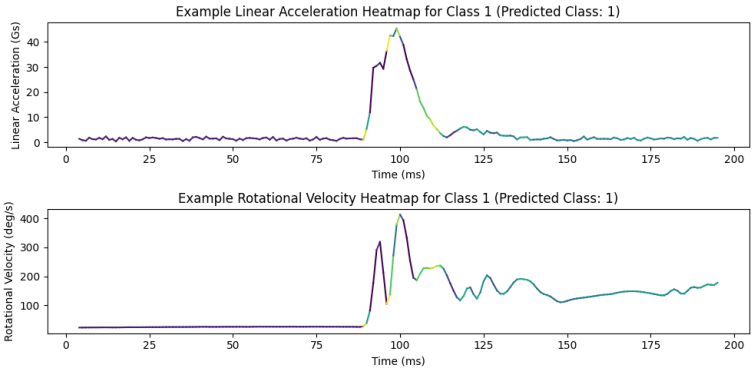

Figure 16. Example heatmap over 200 ms for Class 1.

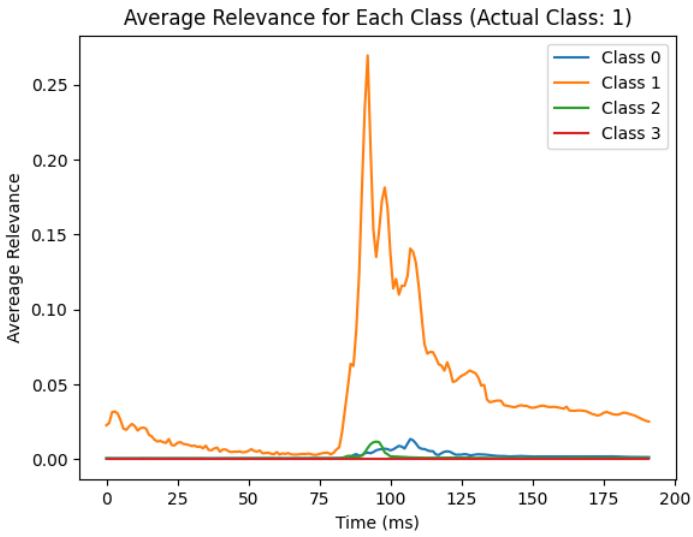

Figure 17. Average relevance over 200 ms for Class 1.

When the large peak was cropped out in the 100 ms condition, the model was forced to focus more on the latter half of the peak, as shown by the larger amplitude in Figure 18 compared to Figures 17 and 15. When looking at the average relevance for Class 0 in Figure 19, it is clear that the model is easily confused between classes 0 and 1. This means that both the precision and recall of the model are lower for the 100 ms input compared to the longer inputs.

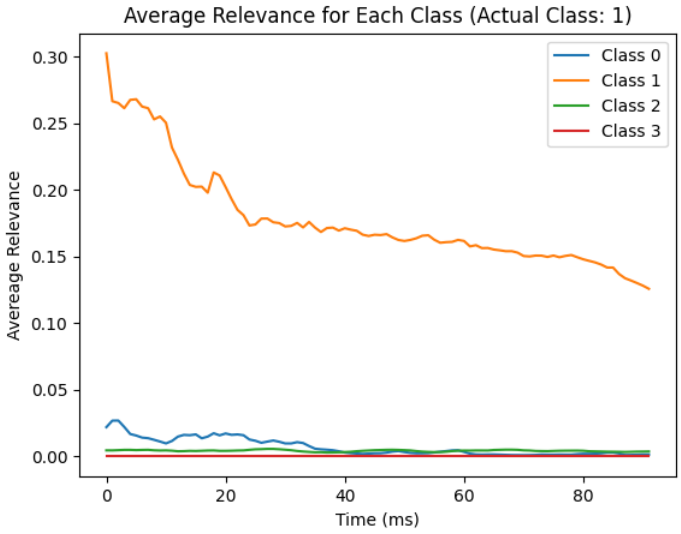

Figure 18. Average relevance over 100 ms for Class 1.

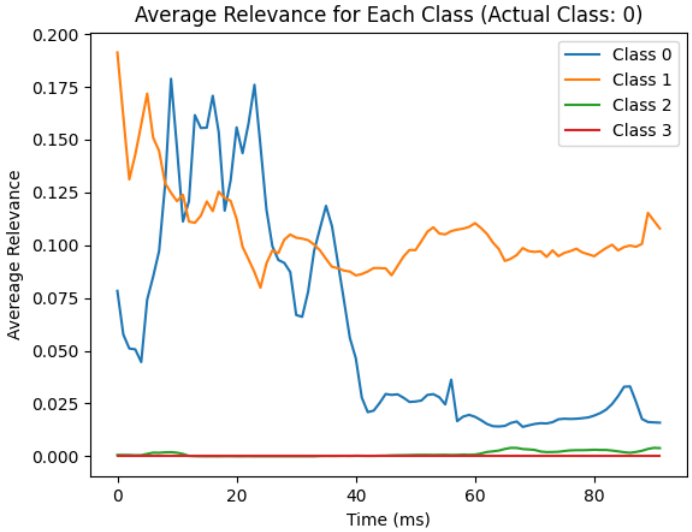

Figure 19. Average relevance over 100 ms for Class 0.

When considering the role of centering and cropping around the peak, it is easiest to examine Class 0. Figure 20 shows that centering the input around the main peak allows the model to make a much more accurate differentiation between classes 0 and 1, since the relevance of Class 1 is much lower than it was in Figure 19. However, classes 2 and 3 begin to appear in misclassifications as well. Figure 21 shows an example heatmap for a correct classification of Class 0, whereas Figures 22 and 23 show misclassifications of classes 2 and 3 as Class 0. It is apparent that zooming in on the walking/jogging data causes the loss of pertinent features that are needed by the model to make accurate classifications.

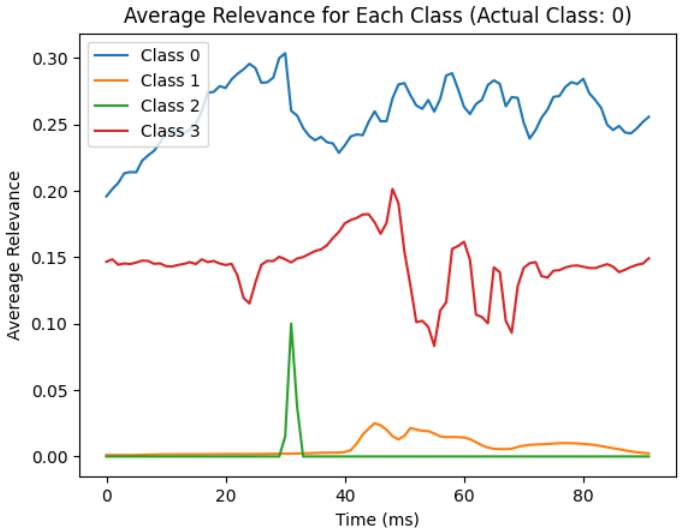

Figure 20. Average relevance over 100 ms (centered) for Class 0.

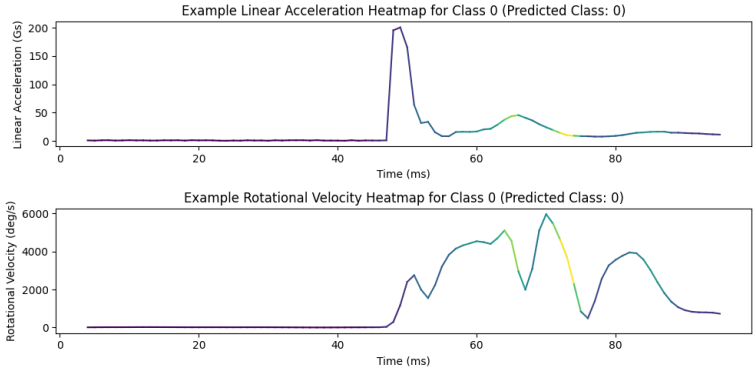

Figure 21. Example heatmap over 100 ms (centered) for Class 0.

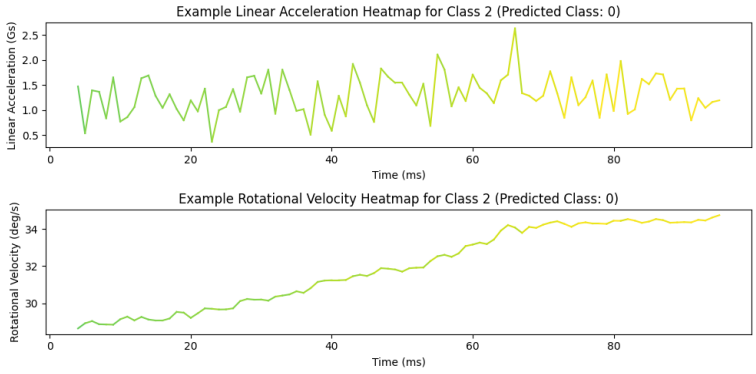

Figure 22. Example heatmap over 100 ms (centered) for Class 2 (misclassified).

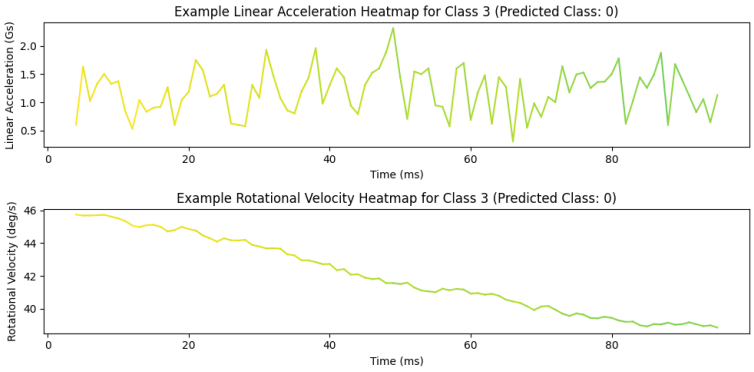

Figure 23. Example heatmap over 100 ms (centered) for Class 3 (misclassified).

Because the period of Class 3 is so large, the time shift added by centering caused a different portion of the periodic data to be used as an input. Evidently, this section of the data is more easily mistaken by the model for the behavior seen in Class 0. The effect of further cropping around the peak can be seen in Figure 24, where the model unexpectedly performs better with less data.

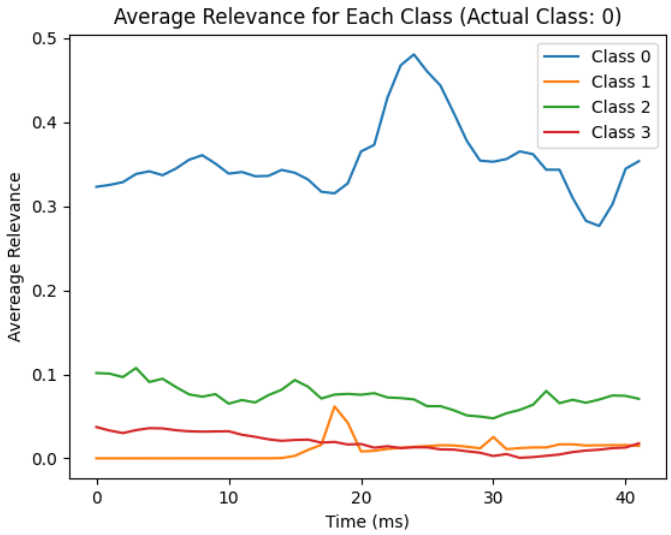

Figure 24. Average relevance over 50 ms (centered) for Class 0.

Returning back to the 2100 ms duration, the period pattern of Class 3 is easily recognized by the model, as shown in Figure 25. An example heatmap is shown in Figure 24, where the relevance is repeatedly higher around the peaks.

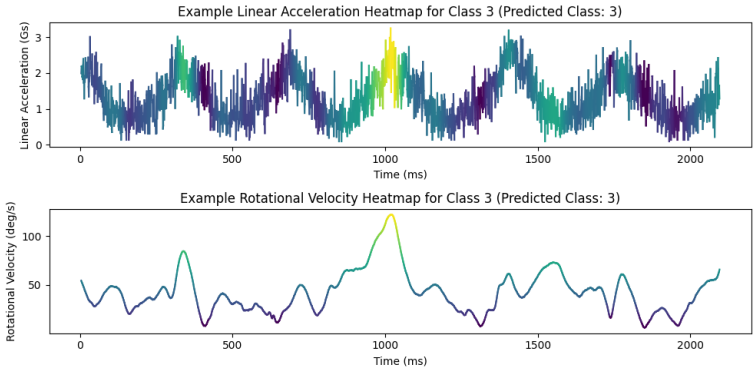

Figure 24. Example heatmap over 2100 ms for Class 3.

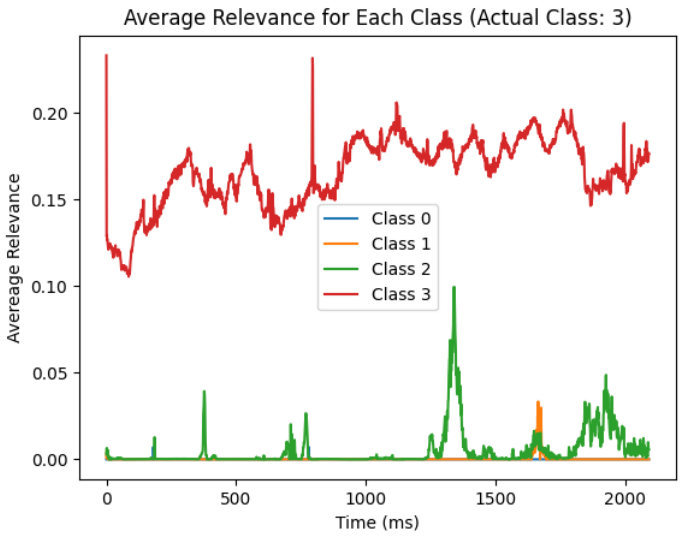

Figure 25. Average relevance over 2100 ms for Class 3.

Figure 26I shows the average relevance for Class 3 over 100 ms, where the recall of Class 3 is extremely poor, showing the effect of cropping out pertinent information. However, it is unclear whether the model performed worse due to the change in information for Class 3 or the other classes with discrete features (like classes 0 and 1). It is likely a combination of both, as this did not occur for Class 2.

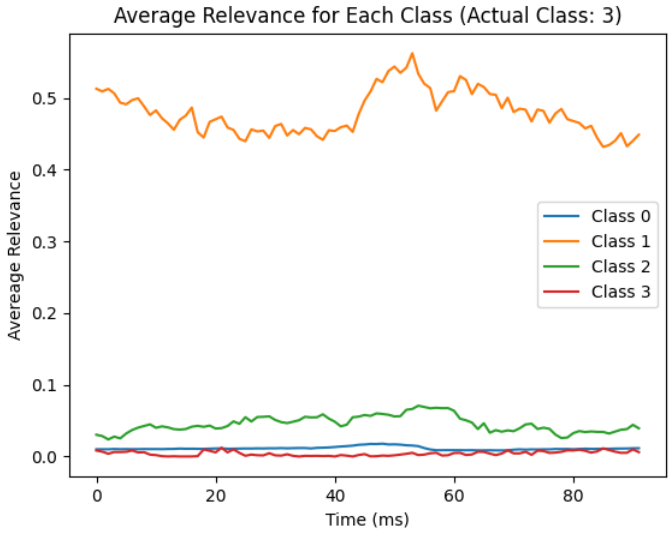

Figure 26. Average relevance over 100 ms for Class 3.

Figure 27 shows the average relevance for Class 3 over 100 ms (centered). As seen before, the effect of shifting the cropping window was drastic for Class 3, causing the model to now mistake Class 0 for Class 3 instead of Class 1. The effect of increased centering can be seen in Figure 28, where the general misclassification trends remain the same, but the recall of Class 3 drops even lower.

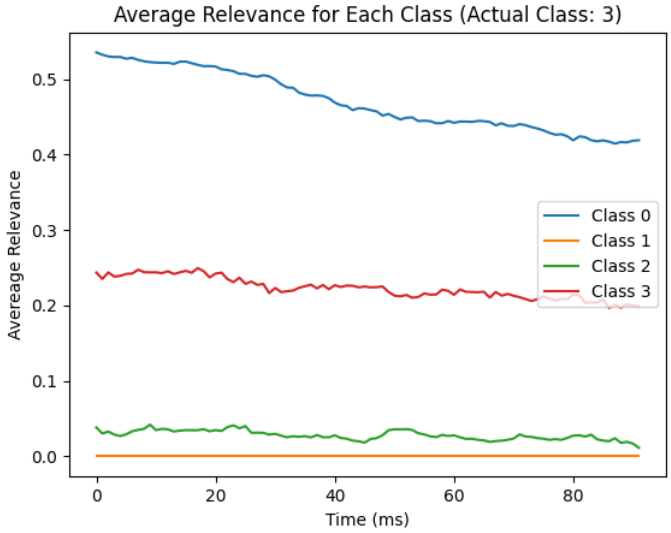

Figure 27. Average relevance over 100 ms (centered) for Class 3.

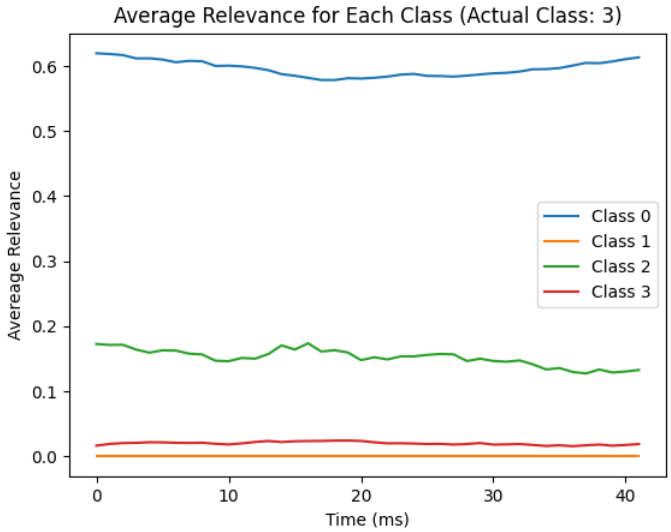

Figure 28. Average relevance over 50 ms (centered) for Class 3.

Experiment 2

Table 5 shows the model's accuracy for each combination of kernel size and number of feature maps. A 2-factor ANOVA revealed that there were no significant differences in model accuracy as a result of kernel size or number of feature maps (Appendix A).

Table 5. Mean model accuracy (± standard deviation) for different kernel sizes and number of feature maps (input length=100 centered, n=10).

		Number of feature maps	
		128	256
Kernel size	5	90.4 ± 2.6	89.7 ± 3.9
	11	90.1 ± 3.4	89.7 ± 3.5

Model accuracy is commonly divided into 2 components: precision and recall, as shown in the equations below. Precision is a measure of the correctness of the model's classifications, while recall measures the completeness of these classifications.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

To further examine the performance of the model in the context of kernel size and number of feature maps, the model's recall was calculated for each class, as shown in Table 6. Because the percentage values were cumulative over 10 models, a 2-factor ANOVA could not be performed. However, general trends could still be observed in the data.

Table 6. Mean recall for different kernel sizes and number of feature maps (input length=100 centered, n=10).

		Number of feature maps			
		128		256	
Kernel size	5	Class 0	Class 1	Class 0	Class 1
		73.3%	98.4%	78.0%	98.1%
		Class 2	Class 3	Class 2	Class 3
		66.2%	25.3%	81.8%	25.0%
	11	Class 0	Class 1	Class 0	Class 1
		63.9%	95.8%	75.0%	99.3%
		Class 2	Class 3	Class 2	Class 3
		55.7%	44.6%	54.9%	38.2%

The Grad-CAM results do not show a significant variation in the classifications for 128 feature maps versus 256. The kernel size showed little effect on the results as well, except for in Class 3, where the model performed better with a larger kernel size of 11. An example heatmap for Class 3 with a kernel size of 5 is shown in Figure 29, and the average relevance is shown in Figure 30. As discussed earlier, Class 3 showed a poor recall of only 25.3% with this condition. However, by increasing the kernel size to 11, the recall increased to 44.6%, as shown in Table 6. Figure 31 shows an example heatmap with the larger kernel size, and Figure 32 shows the average relevance over time compared to other classes. The largest difference between Figures 29 and 31

appears to be the width of the most relevant region, which is higher for the larger kernel size. This makes sense given the role of the kernel is to find features within a given window, so a larger kernel is able to find larger features more easily. Comparing Figures 30 and 32 qualitatively, the latter appears to have a smoother trace for relevance over time, which makes sense given the larger kernel size.

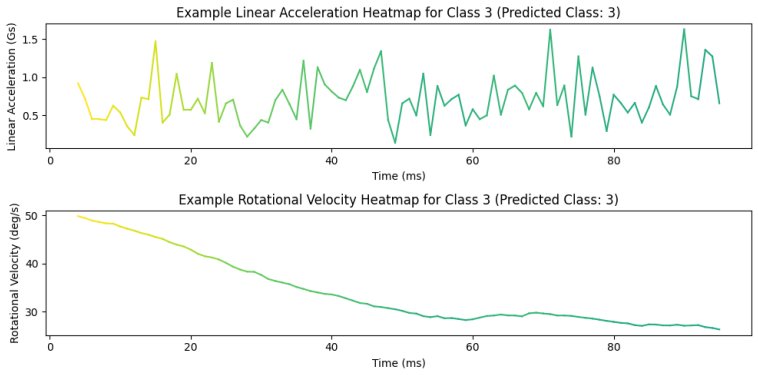

Figure 29. Example heatmap over 100 ms (centered) for Class 3 (k=5, f=128).

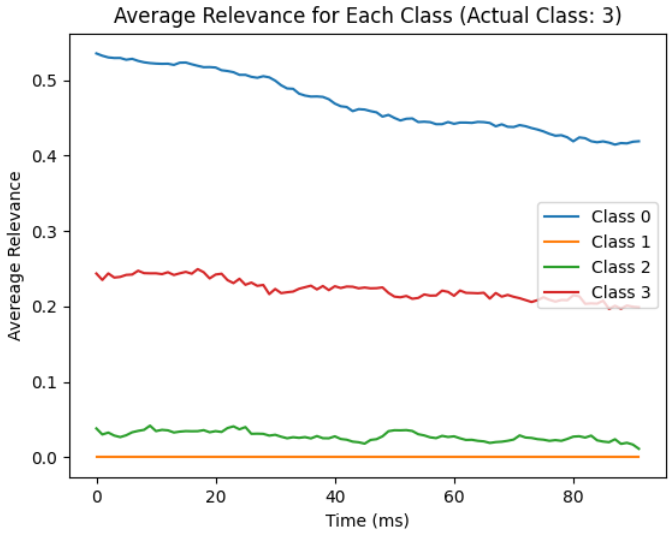

Figure 30. Average relevance over 100 ms (centered) for Class 3 (k=5, f=128).

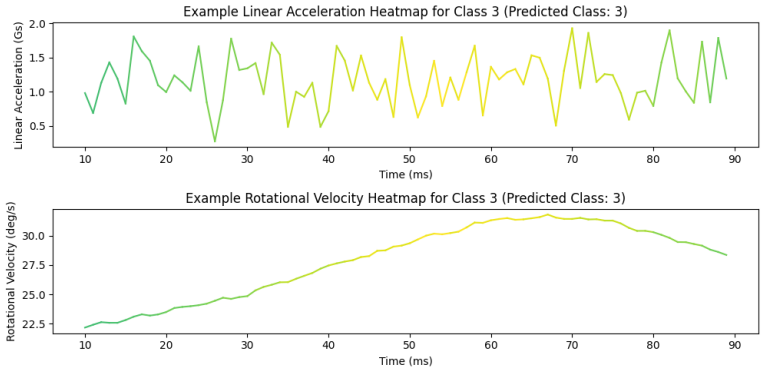

Figure 31. Example heatmap over 100 ms (centered) for Class 3 (k=11, f=128).

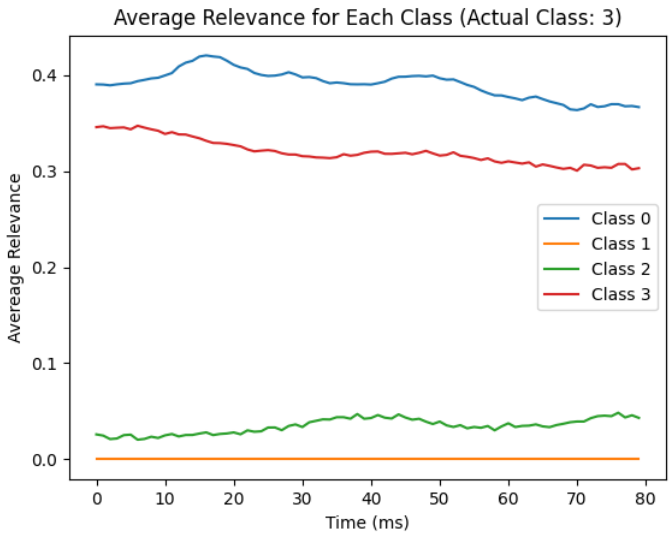

Figure 32. Average relevance over 100 ms (centered) for Class 3 (k=11, f=128).

A common concern with increasing the kernel size is that it may have a harder time finding smaller features, such as the peaks in classes 0 and 1. Table 6 shows that the recall of the model dropped slightly for classes 0 and 1 on average, but not considerably. This resulted in an overall higher accuracy of the model, indicating that it may be an ideal choice for the model moving forward.

IV. *Discussion*

Classifier Performance

In terms of overall accuracy, the ML model performed roughly the same, or in some cases, worse than the basic thresholding scripts. ML requires more resources (ie. computing power, data storage, and data to train on, etc) than basic scripts, so if there is not a considerable increase in the performance, these tradeoffs may not be worth the investment. In these experiments, the marginal gain of using ML over simpler methods has indicated that it may not be ideal for this task. However, there is not enough evidence to discourage the further development of more complex models to complete the binary classification task. Further analysis of the ROC curve should also be done for the ML model to determine if its performance is truly consistent with the overall accuracy.

Applications to Injury Risk Curves

Currently, these classifiers could be implemented to help to reduce the inflation of HIE reports in the literature. Doing so would help the development of more accurate injury risk curves and characterizations of head injury, as fewer false positive impacts would be included. The combined LB LAT and LB DT thresholder is a simple algorithm that could be implemented to lower the number of false positive impacts. The algorithm could either be directly programmed into the wearable sensor device, or incorporated into the pre-processing phase of data analysis. A combined UB LAT and LB DT thresholder has the potential to further lower the number of false positives, but the two populations are not understood well enough for an exact threshold value to be suggested. Thus, it is safer to rely on the LB DT thresholding component to eliminate false positive events such as glitching.

With a maximum AUC of 0.795, the thresholding algorithms did an acceptable job of completing the binary classification task (with an AUC between 0.8 and 0.9 being considered “excellent”) [34]. However, the FPR for each classifier was still higher than ideal at their respective maximum accuracies. For example, the combined LB LAT and LB DT had the highest accuracy at 92.5%, but the FPR at this threshold value was still 32%. It is important to minimize the FPR, since false positive impacts are often high-g and therefore skew the head impact characterization to be higher in magnitude. These high FPRs show that there is still considerable room for improvement with classifier performance. One such improvement may be to optimize the thresholds based on the lowest FPR for that dataset.

Comparison of Simulated and Field Data

The simulated and field datasets were considerably different in terms of distribution, which made it difficult to draw accurate comparisons between the resulting performance for each dataset. The collection of more field data would help increase the reliability of the data, as it is currently only

representative of two high school football players on one day of practice. Redundant data could also be excluded from the dataset so that class sizes are more equal.

Although the field data is limited, the relative frequency of true positive head impacts as compared to high-g non-impacts shows the importance of minimizing the FPR. If one were to blindly include all the collected field data, the number of reported impacts would be 67 times higher than in reality (134 vs 2, respectively). In the literature, it was generally seen that 21 times the number of impacts would be detected for a NCAA lineman (1050 vs 50, respectively) [24]. Additionally, the characterization of head injury would be higher in magnitude, with an increase of 4.4 Gs in peak linear acceleration (25.91 Gs vs 21.51 Gs, respectively) and a decrease of 190 deg/s in peak rotational velocity (1630 deg/s vs 1820 deg/s, respectively). These mischaracterizations of head impact and HIE emphasize the importance of implementing a classifier to exclude false positive impact events.

The overall simplicity of the simulated dataset was both a weakness and a strength of the data. The ideal laboratory conditions in which the simulated data was collected likely contributed to overfitting of the model to specific features, such as the low noise levels in the valid impacts or the specific quirks of the study author's gait. However, these ideal conditions may be applicable in augmenting a dataset of field data, in the same sense that computer-generated data has been used to augment physics-informed machine learning models (PIML) [26].

Grad-CAM Insights

In the context of the current problem, where the data has been simulated, there are unavoidable differences between it and a real dataset. For example, because the valid impact data came from drop tests, a small uptick in rotational velocity can be seen at the time that the wire is cut for each drop test. This uptick would not be present in a real dataset, and so focusing on this confounding variable is a sign that the model may perform worse on real data, where this trend is not present. Other circumstantial features, such as a long period of rest before flicking the DASHR, should ideally be cropped out of the training set to discourage the model from finding correlations rather than causations. Results indicate that the model performs better with longer input data. However, this is likely due to the circumstantial features surrounding the simulated data, as shown through Grad-CAM. A qualitative observation was made that the longer input data also took much longer to train a model with. This was due to the higher number of computations that had to take place to forward/backpropagate the input data through the model. Thus, shorter input data would be preferable if there is not a significant loss in accuracy.

The optimal input length was determined to be 200 ms, as it included enough pertinent information for the model to perform well (and not statistically different from the full 2100 ms data), but did not include circumstantial information from the simulated data. The 100 ms (centered) condition was the second most accurate, but introduced unexpected misclassifications involving Class 3 (running/walking) due to the time shift. This raises the question of whether the

dataset should be augmented to include test cases where the data is time-shifted, so that the model focuses more heavily on the frequency content of the data. This would likely increase the robustness and generalizability of the model, but may lower model performance. Grad-CAM could be used to determine whether the model is still able to find time-shifted features within an augmented dataset.

The Class 2 (standing still) results show the need for "out of set" classification, which is a catch-all class for any test case that the model is not familiar with. The CNN also has a hard time recognizing low/DC frequencies, which make it hard to detect the absence of a feature. The addition of an "out of set" class would also reduce the occurrence of random guesses that the model makes when it does not find a strong connection to any class.

Results show that there are two main types of class features: discrete/high-frequency (ie. classes 0 and 1) and continuous/low-frequency (ie. classes 2 and 3). Continuous features were better classified with longer input data, likely because the period of these features was large compared to the input data length. In contrast, discrete features were oftentimes incorrectly/irrelevantly classified with longer input data. It is difficult to detect these two feature classes with the same kernel size. A multi-headed model would likely perform better at differentiating between these classes, and should be implemented in a future iteration of the model.

Currently, half of the dataset relies on thresholding, while the other half does not. This pre-processing technique is used to isolate discrete activities with peak detection, which causes an innate bias that the model may take advantage of. For classes with discrete events (such as Class 0 and Class 1), the peak is located at the same point in time for all samples, with a cutoff for amplitude and duration. Effectively, this pre-processing technique reduces the amount of work the model has to do to classify a head impact. To some extent, it is ideal to take advantage of this, since thresholding is commonly used in pre-processing. However, if the goal is to eventually move away from the closed-minded idea that head injuries occur at or above a specific threshold, then ideally, the model could make classifications without the need for such pre-processing techniques. This would make more sense for why the same model is being asked to label both discrete and continuous events.

The kernel size and number of feature maps did not significantly affect the accuracy of the model. However, these parameters did appear to have some effects on the accuracy of specific classes. Class 3, for example, performed better with a larger kernel size, with a higher recall score. The larger kernel size also appeared to discern features of the high-frequency classes (Classes 0 and 1) as well, indicating that it may be an ideal choice for the model moving forward.

For classes that did not have a timed event (ie. classes 2 and 3), averaging the input data sometimes caused features to cancel out (destructive interference). Instead of averaging these input data over time to display the "typical" input sequence, it may be more valuable to perform

a cross-correlation between the input data sequences to emphasize any time-shifted features that they share.

V. *Future Work*

Despite the marginal gain of using the current ML model, there is not enough evidence to discourage further exploration of more advanced machine learning models. Advanced models, such as the Long Short-Term Memory model, or LSTM, should be investigated in parallel with simpler classifiers, such as the Support Vector Machine, or SVM. The addition of more field data would help create a more robust dataset for advanced models to train on. Once more field data is available, simulated data should be added in different amounts to determine if it can be used to augment the training set. The binary classification task should continue to be addressed separately from the advanced classification task, as certain model parameters, like the kernel size, can lead to different performance on these two tasks.

Improvements to the current ML model should be made cautiously, with the risk of overfitting increasing as the model accuracy approaches 100%. The next step towards improving the accuracy of the current ML model should be to implement out-of-set classification, with Grad-CAM used to examine the confidence levels during the decision-making process. This would allow researchers to discard outliers and minimize the false positive rate, creating a pipeline for reliable data. Grad-CAM could be used to further improve the transparency of the model's confidence in decision-making, and researchers could use video footage to reexamine the classifications made with low confidence. These changes would help improve the current model while other classifiers are explored in parallel.

VI. *Conclusions*

The purpose of this research was to explore the details surrounding the ML model that was created by Liu [28], and validate or invalidate these methods. Through comparisons to simpler classifiers, it was shown that the ML model was not adding a significant increase in the performance on the binary classification task. Thus, the methods of Liu were not entirely necessary. However, the same approach with a simpler thresholding algorithm would likely work as a screening technique to remove invalid head impacts from the practice data.

The use of Grad-CAM helped to identify several areas within the simulated dataset that could be improved upon to increase the robustness of the current CNN, such as kernel size and cropping. In the future, Grad-CAM can be used to explore scenarios such as multi-label classification, where the input data contains more than one correct classification. With this change, more complex models, such as a multi-headed CNN or a CNN-LSTM, can be used to label long sequences of practice data without the need for thresholding in pre-processing.

Appendix A

Experiment 1 – Statistics

Table 7. Summary Data.

Input Length (ms)	Mean Accuracy (%)	Standard Deviation (%)
2100	93.684	3.717
200	92.982	2.909
100	83.246	3.362
100 (centered)	90.351	2.602
50 (centered)	88.421	3.133

Table 8. One-way ANOVA.

Source of Variation	Sum of Squares	d.f.	Variance	F	p
Between Groups:	703.5077	4	175.8769	17.529	0
Within Groups:	451.5076	45	10.0335		
Total:	1155.0152	49			

Table 9. Tukey HSD Post-hoc Test.

Group 1 vs Group 2: Diff=-0.7020, 95%CI=-4.7271 to 3.3231, p=0.9874
Group 1 vs Group 3: Diff=-10.4380, 95%CI=-14.4631 to -6.4129, p=0.0000

Group 1 vs Group 4: Diff=-3.3330, 95%CI=-7.3581 to 0.6921, p=0.1475
Group 1 vs Group 5: Diff=-5.2630, 95%CI=-9.2881 to -1.2379, p=0.0048
Group 2 vs Group 3: Diff=-9.7360, 95%CI=-13.7611 to -5.7109, p=0.0000
Group 2 vs Group 4: Diff=-2.6310, 95%CI=-6.6561 to 1.3941, p=0.3547
Group 2 vs Group 5: Diff=-4.5610, 95%CI=-8.5861 to -0.5359, p=0.0192
Group 3 vs Group 4: Diff=7.1050, 95%CI=3.0799 to 11.1301, p=0.0001
Group 3 vs Group 5: Diff=5.1750, 95%CI=1.1499 to 9.2001, p=0.0058
Group 4 vs Group 5: Diff=-1.9300, 95%CI=-5.9551 to 2.0951, p=0.6543

Experiment 2 – Statistics

Table 10. Descriptive Statistics.

COUNT	balanced		
	128	256	
5	10	10	20
11	10	10	20
	20	20	40
MEAN			
	128	256	
5	90.422	89.969	90.1955
11	90.127	89.658	89.8925

	90.2745	89.8135	90.044
VARIANCE			
	128	256	
5	5.01177333	13.6869211	8.91127868
11	13.1311567	10.74184	11.3661461
	8.61692079	11.5969713	9.90228615

Table 11. Two Factor Anova.

ANOVA				Alpha	0.05	
	SS	*df*	*MS*	*F*	*p-value*	*p eta-sq*
Rows	0.91809	1	0.91809	0.08626296	0.77066957	0.00239047
Columns	2.12521	1	2.12521	0.19968293	0.65765375	0.00551615
Inter	0.00064	1	0.00064	6.0134E-05	0.9938556	1.6704E-06
Within	383.14522	36	10.6429228			
Total	386.18916	39	9.90228615			

References

1. C. E. Gaw and M. R. Zonfrillo, “Emergency department visits for head trauma in the United States,” *BMC Emergency Medicine*, vol. 16, no. 1, Jan. 2016.
2. R. J. Echemendia and L. J. Julian, “Mild Traumatic Brain Injury in Sports: Neuropsychology's Contribution to a Developing Field,” *Neuropsychology Review*, vol. 11, no. 2, pp. 69–88, Jun. 2001.
3. M. R. Lovell and M. W. Collins, “Neuropsychological assessment of the college football player,” *Journal of Head Trauma Rehabilitation*, vol. 13, no. 2, pp. 9–26, Apr. 1998.
4. A. C. McKee, T. D. Stein, C. J. Nowinski, R. A. Stern, D. H. Daneshvar, V. E. Alvarez, H.-S. Lee, G. Hall, S. M. Wojtowicz, C. M. Baugh, D. O. Riley, C. A. Kubilus, K. A. Cormier, M. A. Jacobs, B. R. Martin, C. R. Abraham, T. Ikezu, R. R. Reichard, B. L. Wolozin, A. E. Budson, L. E. Goldstein, N. W. Kowall, and R. C. Cantu, “The spectrum of disease in chronic traumatic encephalopathy,” *Brain*, vol. 136, no. 1, pp. 43–64, Jan. 2013.
5. A. C. McKee, R. C. Cantu, C. J. Nowinski, E. T. Hedley-Whyte, B. E. Gavett, A. E. Budson, V. E. Santini, H.-S. Lee, C. A. Kubilus, and R. A. Stern, “Chronic traumatic encephalopathy in athletes: Progressive tauopathy after repetitive head injury,” *Journal of Neuropathology & Experimental Neurology*, vol. 68, no. 7, pp. 709–735, Jul. 2009.
6. B. E. Gavett, R. A. Stern, and A. C. McKee, “Chronic traumatic encephalopathy: A potential late effect of sport-related concussive and subconcussive head trauma,” *Clinics in Sports Medicine*, vol. 30, no. 1, pp. 179–188, Jan. 2011.
7. A. M. Spiotta, J. H. Shin, A. J. Bartsch, and E. C. Benzel, “Subconcussive impact in sports: A new era of awareness,” *World Neurosurgery*, vol. 75, no. 2, pp. 175–178, Feb. 2011.
8. Y. Stern, “Long-term consequences of repetitive brain trauma: chronic traumatic encephalopathy,” in *Cognitive Reserve: Theory and applications*, New York: Taylor & Francis, 2007, pp. 460–467.
9. P. McCrory et al., “Consensus statement on concussion in sport-the 5th International Conference on Concussion in sport held in Berlin, October 2016,” *British Journal of Sports Medicine*, 01-Jun-2017. [Online]. Available: https://bjsm.bmj.com/content/51/11/838?source=post_page.
10. P. McCrory, N. Feddermann-Demont, J. Dvořák, J. D. Cassidy, A. McIntosh, P. E. Vos, R. J. Echemendia, W. Meeuwisse, and A. A. Tarnutzer, “What is the definition of sports-related concussion: A systematic review,” *British Journal of Sports Medicine*, 01-Jun-2017. [Online]. Available: https://bjsm.bmj.com/content/51/11/877.abstract.
11. A. Mann, C. H. Tator, and J. D. Carson, “Concussion diagnosis and management,” *The College of Family Physicians of Canada*, 01-Jun-2017. [Online]. Available: https://www.cfp.ca/content/63/6/460.abstract.

12. J. J. Leddy, H. Sandhu, V. Sodhi, J. G. Baker, and B. Willer, “Rehabilitation of concussion and post-concussion syndrome,” *Sports Health: A Multidisciplinary Approach*, vol. 4, no. 2, pp. 147–154, 2012.
13. K. J. Schneider, J. J. Leddy, K. M. Guskiewicz, T. Seifert, M. McCrea, N. D. Silverberg, N. Feddermann-Demont, G. L. Iverson, A. Hayden, and M. Makdissi, “Rest and treatment/rehabilitation following sport-related concussion: A systematic review,” *British Journal of Sports Medicine*, 01-Jun-2017. [Online]. Available: https://bjsm.bmj.com/content/51/12/930.abstract.
14. K. J. Schneider, G. L. Iverson, C. A. Emery, P. McCrory, S. A. Herring, and W. H. Meeuwisse, “The effects of rest and treatment following sport-related concussion: A systematic review of the literature,” *British Journal of Sports Medicine*, 01-Apr-2013. [Online]. Available: https://bjsm.bmj.com/content/47/5/304.short.
15. K. L. Tomei, C. Doe, C. J. Prestigiacomo, and C. D. Gandhi, “Comparative analysis of state-level concussion legislation and review of current practices in concussion,” *focus*, 01-Dec-2012. [Online]. Available: https://thejns.org/focus/view/journals/neurosurg-focus/33/6/article-pE11.xml.
16. S. P. Chrismas and M. A. Schiff, “Implementation of concussion legislation and extent of concussion education for athletes, parents, and coaches in Washington State,” *The American journal of sports medicine*, 2014. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/24510067/.
17. L. G. Concannon, “Effects of legislation on sports-related concussion,” *Physical Medicine and Rehabilitation Clinics of North America*, vol. 27, no. 2, pp. 513–527, Feb. 2016.
18. E. Pellman, D. Viano, A. Tucker, I. Casson, and J. Waeckerle, “Concussion in professional football: Reconstruction of game impacts and injuries: In reply,” *Neurosurgery*, Oct. 2003.
19. S. Rowson, S. M. Duma, J. G. Beckwith, J. J. Chu, R. M. Greenwald, J. J. Crisco, P. G. Brolinson, A.-C. Duhaime, T. W. McAllister, and A. C. Maerlender, “Rotational head kinematics in football impacts: An injury risk function for concussion,” *Annals of Biomedical Engineering*, vol. 40, no. 1, pp. 1–13, Jan. 2012.
20. K. L. O'Connor, S. Rowson, S. M. Duma, and S. P. Broglio, “Head-impact–measurement devices: A systematic review,” *Journal of Athletic Training*, vol. 52, no. 3, pp. 206–227, Mar. 2017.
21. S. Manoogian, D. McNeely, S. Duma, G. Brolinson, and R. Greenwald, “Head acceleration is less than 10 percent of helmet acceleration in football impacts.,” *Europe PMC*, 01-Jan-2006. [Online]. Available: https://europepmc.org/article/med/16817638.
22. C. Kuo, "Measurement and Modeling of Head Impact Kinematics." Order No. 28115233, Stanford University, United States, California, 2018.
23. J. Luck, J. Shrridharani, K. Matthews, J. Kiat, and C. Bass, "A system for measuring head acceleration for impact biomechanics," in *World Congress of Biomechanics*, 2014.

24. L. C. Wu, L. Zarnescu, V. Nangia, B. Cam, and D. B. Camarillo, “A head impact detection system using SVM classification and proximity sensing in an instrumented mouthguard,” *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 11, pp. 2659–2668, Nov. 2014.
25. Wu, L. C., Kuo, C., Loza, J., Kurt, M., Laksari, K., Yanez, L. Z., … Camarillo, D. B. (2017). Detection of American football head impacts using biomechanical features and support Vector Machine Classification. *Scientific Reports*, *8*(1). doi:10.1038/s41598-017-17864-3
26. Raymond, S. J., Cecchi, N. J., Alizadeh, H. V., Callan, A. A., Rice, E., Liu, Y., … Camarillo, D. B. (2022). Physics-informed machine learning improves detection of head impacts. *Annals of Biomedical Engineering*, *50*(11), 1534–1545. doi:10.1007/s10439-022-02911-6
27. Wang, T., Kenny, R., & Wu, L. C. (2021). Head impact sensor triggering bias introduced by linear acceleration thresholding. *Annals of Biomedical Engineering*, *49*(12), 3189–3199. https://doi.org/10.1007/s10439-021-02868-y
28. Liu, P. (2021). *Quantifying Head Impact Exposure and Classifying Relevant Impacts with Machine Learning in High School Football* [Unpublished Graduation With Departmental Distinction Thesis]. Duke University.
29. Campbell, K. R., et al. (2020). Head impact telemetry system’s video-based impact detection and location accuracy. *Medicine & Science in Sports & Exercise*, *52*(10), 2198–2206. https://doi.org/10.1249/mss.0000000000002371
30. Schuldhaus, D., Jakob, C., Zwick, C., Koerger, H., & Eskofier, B. M. (2016). Your personal movie producer. *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. https://doi.org/10.1145/2971763.2971772
31. Holleczek, T., Ru, A., Harms, H., & Tro, G. (2010). Textile pressure sensors for sports applications. *2010 IEEE Sensors*. https://doi.org/10.1109/icsens.2010.5690041
32. Bedri, A., Li, R., Haynes, M., Kosaraju, R. P., Grover, I., Prioleau, T., Beh, M. Y., Goel, M., Starner, T., & Abowd, G. (2017). Earbit. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *1*(3), 1–20. https://doi.org/10.1145/3130902
33. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2019). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, *128*(2), 336–359. https://doi.org/10.1007/s11263-019-01228-7
34. Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, *5*(9), 1315–1316. https://doi.org/10.1097/jto.0b013e3181ec173d