Data Science Capstone Final Report

Shobhit Asthana (sasthana@calpoly.edu)
Austin Schwarz (auschwar@calpoly.edu)
Roy Zawadzki (rzawadzk@calpoly.edu)

Introduction

For decades, people have been spending less time outside and exercising and more time sedentary in front of a TV or computer, leading to concerning health trends. According to the U.S. Department of Human and Health Services, 28% of Americans, or 80.2 million people, are physically inactive, with many more people barely meeting recommended activity levels [1]. As research has shown, increased activity level is linked to positive health outcomes both physical and mental. In order for kinesiologists and health experts to devise effective interventions to promote physical activity, quality data must be available for analysis.

As technology has progressed in the most recent decades, so has the ability for clinicians to accurately measure physical activity. Whereas questionnaires were once used to collect information about an individual's activity types and level, highly-sophisticated wearable devices and data processing methodology such as machine learning have become the leading methods to collect and analyze activity data [2]. We contribute to the advancing field of wearable technologies by employing machine learning to process and classify activities based on information collected by sensors on different parts of a participant's body.

We have collaborated with Dr. Sarah Keadle from the Cal Poly Kinesiology Department in a continuation of a project applying machine learning to wearable data, working to improve upon methodology developed by two capstone groups and a master's student. Like these groups, we answer the question: can we accurately predict what specific activity an individual is doing for an observed time period of interest? The five activities that we predicted were sitting or lying down, standing and moving, walking, running, and bicycling.

Last year's capstone group, or our starting point for our contributions, achieved good prediction ability with a neural network for each sensor. We iterate upon these models by further exploiting the structure of the longitudinal dataset of activity tracking. This diverges from the efforts of previous groups that have treated observations as independent from each other, that is as if this were a cross-sectional dataset.

In our improvements upon the work of previous groups, we consider two important aspects of the data. Firstly, that points either represent points where an individual is either transitioning from one activity to another, such as walking to sitting, or in the middle of an activity block such as walking. Secondly, the previous information about what an individual was doing in periods preceding the period of interest is highly relevant and contains helpful information to predict what an individual is doing in the period of interest. The improvements can be into two groups: model-agnostic improvements, as in improvements that are not contingent on prediction model choice, and specific modeling improvements.

Of the model-agnostic improvements, we implemented a new cross-validation methodology that assigns activities to train and test sets by two-hour activity blocks as opposed to simply randomizing the rows. In addition, we implemented a strategy to balance out class labels in the dataset by reducing the number of rows in the most common class, sitting and lying down, by half. Lastly, we devised a simple post-processing algorithm that smooths over model test set predictions by inferring identifying non-transition points and imputing divergent values with the majority of surrounding points.

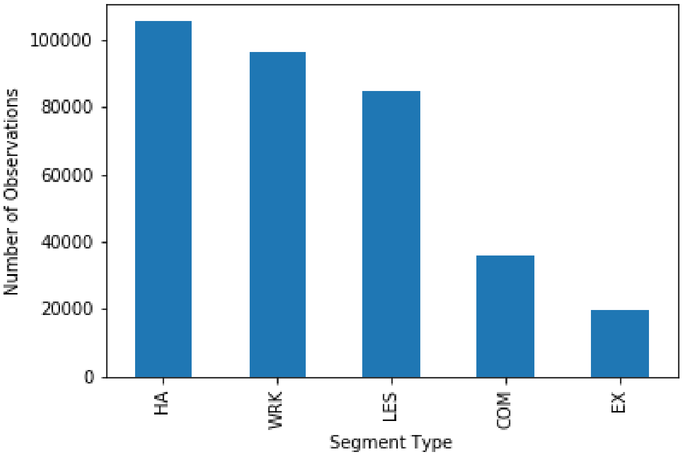
Of the model improvements, we created two new different types of models. The first new model we created a one dimensional convolutional neural network (CNN), which is a type of neural network that takes the closest points in time around an observation and derives features from them in order to make predictions. The second model architecture we used is a recurrent neural network (RNN), which also takes into account previous data points and evaluates them as a sequence rather than an independent observation.

Features

The model has a multitude of features ranging from basic statistics to very complex frequency features. The raw data has accelerometer measurements for the three axes and a timestamp. There was also a separate dataset consisting of a log of second by second encodings, which serve as the ground truth for the activities each person was performing. Because these files require merging, the data was aggregated on a second by second basis, meaning each second of time corresponds to a single observation in the dataset. The more basic statistical features are the minimum, maximum, twenty-fifth percentile, seventy-fifth percentile mean, and standard deviation for all three axes. There is also the mean value of axes after they were multiplied together. Some of the more complex features were mean and standard deviation of vector magnitude and the angle of acceleration relative to the vertical. Lastly, the model has selected fast fourier transforms to measure frequency level statistics. Table 1 contains more complex information about each of these features.

The primary class labels the model tries to predict are walking, running, bicycling, sitting or lying down, and standing and moving. Standing and moving is movement that is not long enough to classify as walking.

There were five different types of activities that we called segments. The first one was household activities, which are basic chores a person would do around the house like meal prep or cleaning their room. The second activity was work activities, which were typical things someone would do at work, like computer work or going to meetings. The third activity included errands and transportation, which were activities like shopping or riding the bus. The fourth activity was sedentary leisure, which was activities like watching television or playing video games. The last activity was active leisure, which was activities like playing sports or riding a bike. All participants wore a monitor for two hours and did a combination of the above activities.



Methods

For a baseline of different improvement strategies, as it relates to this prediction problem, we use a basic feed forward neural network for each sensor with standard k-fold, independent rows cross-validation. In the analysis, this is called the "Basic Neural Network."

Model-Agnostic Improvements

New Cross-Validation Methodology

All model training up until this point implemented standard k-fold cross-validation where the rows were randomly assigned to training and testing. This assumes that rows are independent of each other, which is not a reasonable assumption; many times, rows from the same subject and even the same two-hour activity block will be in train and test sets, giving the algorithm partial information about what is going to predict. This can lead to underestimates of modeling testing error [3]. Moreover, we wanted to ensure equal representation of each segment type in the training and testing set.

As a result, we constructed a stratified cross-validation mechanism where we would select whole two-hour segment blocks to be placed into the training and testing. In this paper, this practice is called the "New CV." The algorithm goes as follows:

1. Enumerate all segment blocks (i.e. label all k segment blocks 1,2,...,k-1,k)
2. Assign enumerated segment blocks into datasets by segment type (we will have five datasets for the five different segment types: HA, WRK, LES, COM, and EX)

3. Within each segregated dataset, select the first 20% of the segments as testing and the last 80% of the segments as training.
4. Coalesce the 20% testing sets from each segregated dataset into one testing set and the 80% training sets from each segregated dataset into one training set
5. Run the model and record down metrics.
6. Repeat steps 3-5 but with the next 20% for each segregated dataset as the test set (i.e. 5-fold cross-validation)

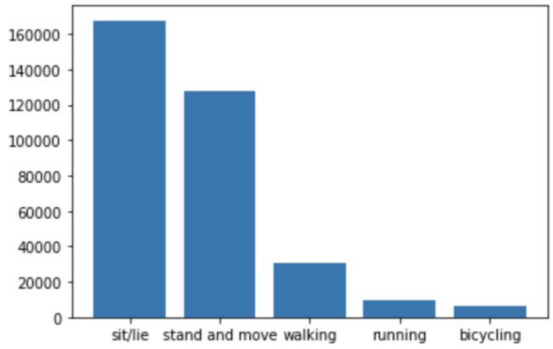
Post-processing

The distinction of data points into activity transition points and non-transition points offers the chance to correct likely prediction mistakes that are found in non-transition blocks. To formally define transition and non-transition points, transition points are the data points that encompass transitioning from one activity from another (e.g. the few seconds to transfer walking to sitting) while non-transition points are the points that make up the core of an activity block, away from transitions. For example, if one goes for an uninterrupted walk, all points excluding the few seconds that make up the beginning and end of the walk are non-transition points. In this sense, we utilize the fact that non-transition points make up many of the points in the prediction problem and that there exists activity inertia. That is, given someone is engaged in a particular activity, it is overwhelmingly likely that they will continue to do this same activity.

We use this concept of activity inertia in our post-processing algorithm. This algorithm takes the predictions made by the model on the test and imputes points that diverge from the points that surround it as the mode of these surrounding points. More specifically, the algorithm first sweeps through the predictions, takes the mode of the surrounding four points (two in the back and two in the front). Then it compares this mode to the point of interest. If it is different, the point of interest is replaced by the mode. See Figure 1 in the appendix for an example. This essentially corrects mistakes where, within a non-transition point block, the model predicts some other activity.

Class Rebalancing

In ideal conditions, a classification would perform best when there is an even distribution of classes in the training set. This means the model has a sufficient number of observations of each class to train over. However, across all sensors, we noticed the “sit/lie” activity segment represented a large proportion of the data (see Table 2).



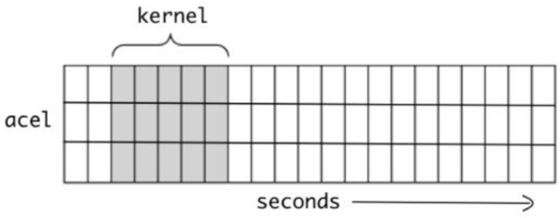
As you can see, “sit/lie” makes up nearly 50% of the total observations for each monitor. Additionally, the “bicycling” class was the least represented. The imbalance of classes led us to believe that our model may be more biased toward predicting sit/lie over other classes.

To test this hypothesis, we removed half of the “sit/lie” observations from the dataset so that the classes would be more evenly distributed. To maintain time sequences, we removed the first half of “sit/lie” observations in order to keep the sequence dependencies of the second half intact. Then we trained the model using the same new cross validation methodology.

Modeling Improvements

Convolutional Neural Network

One model that we used to predict the different activities was a convolutional neural network. Usually, convolutional neural networks are used in image processing to detect certain patterns to predict classifications. These are two dimensional. However, convolutional neural networks are also useful for working with one dimensional time series sensor data, like data from an accelerometer. A one dimensional convolutional neural network has a kernel of a set width that slides from the first data points all the way to the last, aggregating features together from windows and making a smaller dataset. Here is a visual representation:



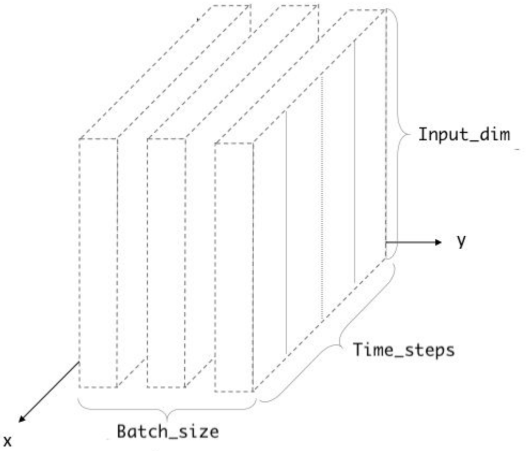
The shaded kernel represents the data points being aggregated. They will become a single point in the next layer. Eventually, the network will detect advanced features and use them to classify data points as one of the activities.

The model has four one dimensional convolutional neural network layers and two dense layers. At the end, the model outputs a matrix of size number of observations by the number of possible classifications. Due to the nature of our cross-validation, this could have either four or five classifications, based randomly on how the data was split into the folds. To get the predictions, we take the classification column with the largest number for each data point and classify the data point as one of the five activities.

Recurrent Neural Network

The most complex deep learning model we implemented was a recurrent neural network

(RNN). RNN's are especially useful for time series data since each prediction takes into account previous sequences of data. The data inputted into a RNN is three dimensional, representing samples, time steps, and features where a sample represents a time sequence, time steps are the number of observations made in the sample, and features are the data or statistics collected. For this application, each sample consisted of 5 second sequences within an activity segment. Here is a visualization of the input shape:



In this image, the time steps represent one point of observation in the sample and the features are statistics and measurements collected at each observation. For the application of predicting movement patterns, we reshaped the data to reflect 5 time steps per sample representing a 5 second sequence of movement.

The specific implementation of RNN we used is called a Long Short Term Memory (LSTM) Network, which has feedback connections that allows the network to process entire sequences. The mechanics of how an LSTM function is outside the scope of this paper, but the idea is that the network can store the results of previous timesteps in a sample that then affect future predictions. For more detailed information about LSTMs [4].

Step Study

In addition to the AM Study data we had, we also received some raw data from a different study called "Step Study". Every participant in "Step Study" did one of two options. The first option was a twenty-five minute session on the treadmill followed by eighteen minutes of self paced walking. The second option was sitting for six minutes, followed by a short self-paced walk to a vehicle, and finally a ten minute drive. We received both a log of all of the activities that each of the participants had done and four sets of raw accelerometer data for four sensors: hip, wrist, chest, and thigh.

This data was then aggregated in the same way that the "AM Study" was and turned into four comma-separated value files for each of the sensors with all of the features present in the AM Study and multiple columns of classifications. While we ran out of time to do this ourselves, the "Step Study" data should be able to be easily combined with the current "AM Study" data that is used in the current model. The "Step Study" data will boost the number of data points with walking classifications while also creating a new prediction label, driving.

Results and Discussion

Basic neural network results are shown in Table 3. The baseline model performed well at an overall accuracy of 81.5% for hip, 92.5% for chest, 68.4% for wrist, and 94.2% for thigh. Combining the new CV with the basic neural network did not change bottom-line accuracy by a noteworthy amount. However, it did collapse misclassifications in confusion matrices into certain categories such that errors were not as spread across classes (see Table 4). Ultimately, our best performing model was the RNN with class rebalancing, which yielded a cross validated accuracy of 92.5% for hip, 92.9% for chest, 90.1% for wrist, and 91.7% for thigh. Though performance for the thigh sensor decreased a little bit, we saw a drastic improvement in the wrist sensor which served to be the sensor with the highest variance.

Model Agnostic Results

Class Rebalancing

Class rebalancing had moderate to high gains across models and sensors (see Table 5). The biggest gain can be found in predicting for the wrist sensor where the RNN with the new cross-validation and class rebalancing improved 20.4% to a 90.1% accuracy when compared to an RNN with no class rebalancing. By employing our class rebalancing strategy, combined with an RNN, we are able to take a sensor that models have had trouble predicting for and boost its predictive ability past 90% accuracy. For the already accurate sensors, namely Chest and Wrist, there was little to no improvement, but for the sensors that were previously inaccurate, accuracies were boosted to over 90%. This indicates that the dominance of sit/lie in the dataset was perhaps skewing the model in the form of high false positives for this class.

Basic Postprocessing

Basic post-processing, like class rebalancing, contained the most gains for the sensors that had initial poor predictive accuracy, Hip and Wrist, though this was dependent on model type (see Table 6). The RNN saw the most gains by post-processing, while CNN contained considerable losses. For the Hip sensor, however, there were no losses from post-processing. These mixed gains suggest that there are still considerable tweaks left to be made for such an approach, but the post-processing of non-transition points offers potential gains over leaving this strategy out.

Modeling Results

The convolutional neural network was a little bit less accurate than our other networks. On average, the network was five percent worse at predicting the correct classification compared to the neural network with the new cross validation. It was around five percent worse than the recurrent neural network for hip, chest, and thigh, but significantly worse for the wrist. The best convolutional neural network was the model with class rebalancing and no post-processing. The model performed best on the chest and thigh sensors and the worst on the wrist sensors. All of the results are in the appendix Table 7.

The recurrent neural network served to be our best performing and most robust model architecture. We saw the largest improvement in accuracy in the wrist sensor, the worst performing sensor, as well as the model improved accuracy in the hip and chest sensors, which already performed well. The best performing model was the RNN with class rebalancing. In shaping the data for the RNN to train on, we found the best time step interval was between 5-7 seconds. Interestingly, post-processing had an insignificant effect on model performance. However, the main downside to the RNN is that it requires an extremely long time to train and run the model. This made cross validation especially difficult as it would require more than 1 hour to run for each sensor. Overall, we believe a LSTM network architecture is best suited for time-series applications. The results are explained in more detail in the appendix at Table 8 Recurrent Neural Network.

Future Directions

There are several further improvements that can serve as future directions for predicting activity type with accelerometer data. Beginning with the post-processing algorithm, the one that we implemented in this analysis was fairly basic and can be increased in complexity with larger gains in predictive power hopefully following. One method, like the one we implemented, can look at the points surrounding the point of interest and given that this point diverges from the other points. Then, a score representing how confident we are that this point is incorrect can be computed using both information about the class of the point of interest and the proportion of points of each class surrounding the point. The optimal number of surrounding points to look at and a threshold for the confidence score can be added as hyperparameters to the model training process.

Additionally, based on our initial results on class rebalancing, there could be significant opportunities to improve model performance by applying different methods of balancing classes. We identified two approaches that would help with rebalancing. The first is to simply calculate the number of observations to remove from each segment to reduce the disparities in the number of observations across the different activities. This method is the simplest to do and would extend the work our group did. However, this approach may not be the best as it is essentially removing a significant amount of observations to train on. Thus, a preferred method would be to simulate more observations so that classes are represented more equally. The challenge with this approach is that it is difficult to ensure the time series nature of the data remains intact. A possible solution is to take an entire time sequence and vary the data points by a small amount. This would ensure that some of the dynamics of the time sequence would be the same.

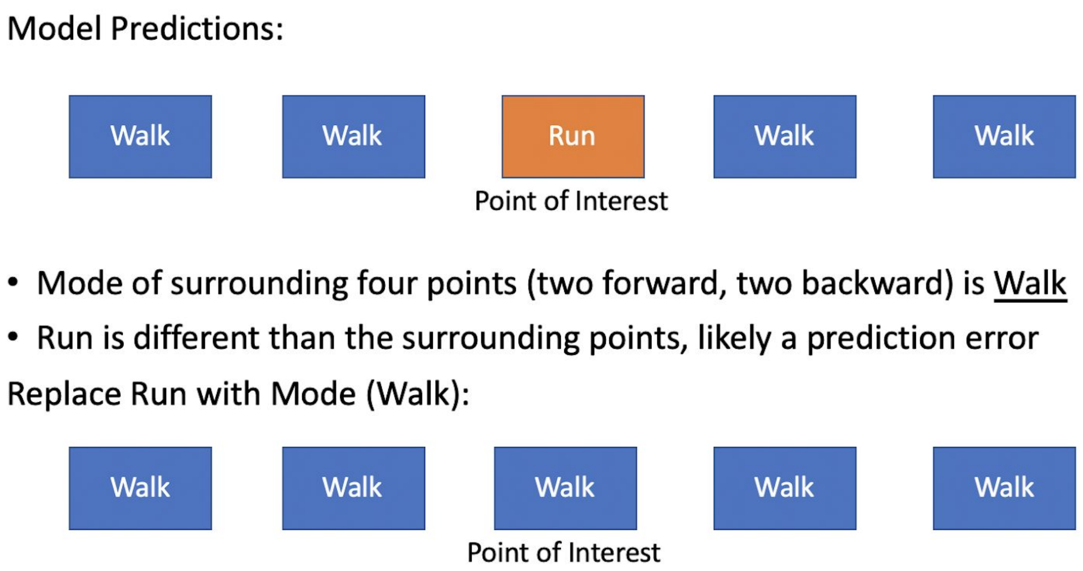
Lastly, adding in the extra data from the “Step Study” will help to make our model more robust. The “Step Study” data also has a new classification, “driving”, which helps expand the scope of the project. We hope that adding in the new data will improve the model performance and increase its scope.

Conclusion

Overall, with the new stratified cross validation method, we are now able to better evaluate model performance. As illustrated by our results, a RNN network architecture combined with class rebalancing yielded the best overall performance with the most significant improvement in the wrist sensor. While this network architecture does serve to be extremely time consuming to run, we feel that it is the best model to handle the time series nature of the data. Additionally, we found that class rebalancing provided significant improvement in accuracy. We hope that this work can provide a more robust model and evaluation framework for predicting movement patterns.

Tables and Figures

Figure 1: Post-processing example



Model Predictions:

- Mode of surrounding four points (two forward, two backward) is Walk
- Run is different than the surrounding points, likely a prediction error

Replace Run with Mode (Walk):

Table 1: Features

Feature	Description

X.mean, y.mean, z.mean	Mean on each axis for each second on each axis
X.min, y.min, z.min	Minimum on each axis for each second
X.max, y.max, z.max	Maximum on each axis for each second
Xy.mean, xz.mean, yz.mean, xyz.mean	Mean products of the axis’ acceleration for each second
X.sd, y.sd, z.sd	Standard Deviation of acceleration for each axis for each second
mean.vm	Mean vector magnitude for each axis for each second (vm = $(x^2 + y^2 + z^2)^{0.5}$)
sd.vm	Standard deviation of vector magnitude for each second
Pct25, pct75	Twenty-fifth ands seventy-fifth percentile of vector magnitude for each second
Mean.ang, sd.ang	Mean and standard deviation of angle of acceleration relative to the vertical (ang = 90arcsin(x/vm)(pi/2))
Xfft, yfft, zfft, mfft	Fast Fourier Transforms(1-15 Hz) for the x, y, and z axis and for the vector magnitude

Table 2: Class Proportions

Sensor	Sit/Lie (proportion)	Stand and Move	Walking	Running	Bicycling
Hip	49.2	37.4	8.9	2.7	1.8
Chest	49.8	36.5	9.3	2.6	1.7
Wrist	49.2	37.4	8.9	2.7	1.8
Thigh	46.7	36.1	12.2	2.6	2.3

Table 3: Basic Neural Network

Hip
Accuracy: 81.5%

	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Precision	0.894	0.776	0.857	0.972	0.000
Recall	0.862	0.814	0.686	0.935	0.000
F1-Score	0.871	0.781	0.752	0.952	0.000

Confusion Matrix

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	140000	23000	150	0	84
Stand and Move	19000	100000	4200	220	240
Walking	87	9300	21000	120	49
Running	1	300	290	8600	10
Bicycling	630	5000	440	8	0

Chest
Accuracy: 0.925

	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Precision	0.981	0.880	0.884	0.975	0.986
Recall	0.960	0.937	0.747	0.927	0.628
F1-Score	0.970	0.906	0.805	0.950	0.766

Confusion Matrix

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	170000	7200	79	3	100
Stand and Move	300	120000	4600	380	230
Walking	46	8500	25000	25	16
Running	13	540	94	8600	32
Bicycling	320	1900	27	1	3800

Wrist
Accuracy: 68.4%

	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Precision	0.784	0.630	0.792	0.896	0.000
Recall	0.748	0.665	0.500	0.825	0.000
F1-Score	0.758	0.630	0.790	0.896	0.000

Confusion Matrix

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	130000	42000	320	21	150
Stand and Move	38000	85000	4400	390	420
Walking	1300	13000	15000	410	110
Running	12	1200	190	7600	170
Bicycling	370	5300	300	140	0

Thigh

Accuracy: 94.2%

	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Precision	0.991	0.924	0.850	0.969	0.994
Recall	0.986	0.939	0.831	0.888	0.770
F1-Score	0.988	0.930	0.836	0.925	0.866

Confusion Matrix

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	120000	1500	88	0	170
Stand and Move	790	9000	4800	140	170
Walking	7	5400	27000	50	61
Running	18	200	550	6300	24
Bicycling	320	650	370	59	4700

Table 4: Basic NN New CV Confusion Matrices

Hip

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	71000	15000	26	0	27
Stand and Move	3700	99000	3700	2200	31
Walking	4	4700	2000	14	8
Running	0	53	53	9100	0

Bicycling	21	5000	4500	5	0

Chest

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	8000	6400	26	0	32
Stand and Move	4903	87000	3700	320	71
Walking	3	4200	26000	13	0
Running	0	240	8	900	13
Bicycling	5	350	0	1	4700

Wrist

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	53000	32000	47	7	32
Stand and Move	15000	89000	2300	300	97
Walking	400	7600	17000	34	49
Running	0	140	130	8900	5
Bicycling	21	500	85	330	0

Thigh

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	460000	960	35	0	100
Stand and Move	550	69000	2900	180	93

Walking	5	3600	210000	6	26
Running	0	19	22	5800	6
Bicycling	130	220	0	8	5800

Table 5: Class-rebalancing results by model (accuracy)

	Basic NN w/ New CV **Before**	Basic NN w/ New CV **After**	CNN w/ New CV **Before**	CNN w/ New CV **After**	RNN w/ New CV **Before**	RNN w/ New CV **After**
Hip	80.6%	86.4%	79.3%	80.3%	80.9%	92.5%
Chest	92.4%	90.7%	86.6%	87.6%	92.8%	92.9%
Wrist	67.8%	72.5%	62.1%	63.5%	69.7%	90.1%
Thigh	94.1%	94.2%	88.5%	90.3%	94.3%	91.7%

Table 6: Post-processing resulting by model (accuracy)

	Basic NN w/ New CV **Before**	Basic NN w/ New CV **After**	CNN w/ New CV **Before**	CNN w/ New CV **After**	RNN w/ New CV **Before**	RNN w/ New CV **After**
Hip	80.6%	81.1%	79.3%	80.4%	80.9%	82.2%
Chest	92.4%	93.9%	86.6%	81.3%	92.8%	90.4%
Wrist	67.8%	67.2%	62.1%	55.7%	69.7%	73.3%
Thigh	94.1%	95.0%	88.5%	85.1%	94.3%	94.0%

Table 7: CNN New CV Results

Hip

Accuracy = 79.2%

	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Precision	0.7279	0.7781	0.3704	0.9152	0
Recall	0.8512	0.7934	0.7279	0.8281	0
F1-Score	0.7809	0.7785	0.4909	0.8676	0

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	97743	8452	47	8560	25
Stand and Move	12883	54959	1283	18	126
Walking	3	766	2058	0	0
Running	21771	167	2125	115900	0
Bicycling	4404	8426	43	306	0

Chest

Accuracy = 88.3%

	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Precision	0.8897	0.8651	0.6338	0.9082	0.1994
Recall	0.8611	0.8965	0.7874	0.9459	0.1826
F1-Score	0.8691	0.8778	0.6953	0.9241	0.1907

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	116863	6100	70	12548	138
Stand and Move	4794	67401	2793	184	8
Walking	50	1209	4740	21	0

Running	6797	115	95	122663	3
Bicycling	4515	4078	1	857	2112

Wrist

Accuracy = 61.1%

	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Precision	0.3943	0.5101	0.3035	0.6514	0.2107
Recall	0.4717	0.3779	0.6523	0.8756	0.2188
F1-Score	0.4261	0.4244	0.4143	0.7270	0.2146

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	54161	10716	209	49580	161
Stand and Move	23609	26176	1895	17575	14
Walking	217	766	1844	0	0
Running	14752	494	2125	122550	42
Bicycling	412	1971	2	7911	2883

Thigh

Accuracy = 90.3%

	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Precision	0.9377	0.8676	0.5170	0.9272	0.0
Recall	0.9518	0.9089	0.7979	0.9711	0.0
F1-Score	0.9445	0.8856	0.6272	0.9475	0.0

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	87172	3711	59	642	0
Stand and Move	4061	63266	2263	13	0
Walking	6	1065	4228	0	0
Running	826	77	1612	84581	0
Bicycling	1030	5772	0	4756	0

Table 8: Recurrent Neural Network New CV

Hip

Accuracy = 91.9%

	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Precision	0.0.9965	0.8949	0.8474	0.9724	0
Recall	0.9991	0.9677	0.6880	0.882	0
F1-Score	0.9977	0.9278	0.7552	0.9198	0

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	84000	77	0	0	3
Stand and Move	290	120000	3500	210	140
Walking	0	9400	21000	53	25
Running	0	460	390	8200	180
Bicycling	5	6000	130	0	0

Chest

Accuracy = 93.2%

	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Precision	0.9975	0.8994	0.8695	0.9647	0
Recall	0.9991	0.9658	0.7494	0.9469	0
F1-Score	0.9983	0.9302	0.8003	0.9552	0

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	90000	87	0	0	0
Stand and Move	210	130000	3900	200	84
Walking	15	8400	25000	51	4
Running	0	240	150	8700	100
Bicycling	5	5900	110	84	0

Wrist

Accuracy = 90.2%

	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Precision	0.0.9944	0.8722	0.7919	0.9277	0
Recall	0.9939	0.9612	0.5848	0.8139	0
F1-Score	0.9942	0.9124	0.6651	0.8355	0

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling

Sit/Lie	83000	500	2	0	0
Stand and Move	460	120000	3900	300	3200
Walking	10	12000	18000	97	62
Running	0	710	970	7500	29
Bicycling	2	5700	310	74	0

Thigh

Accuracy = 92.2%

	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Precision	0.9944	0.9174	0.8535	0.9639	0.0
Recall	0.9999	0.9588	0.8038	0.9811	0.0
F1-Score	0.9969	0.9381	0.8156	0.9722	0.0

True / Predicted	Sit/Lie	Stand and Move	Walking	Running	Bicycling
Sit/Lie	62000	29	0	0	0
Stand and Move	350	92000	3300	220	59
Walking	1	6300	26000	58	31
Running	0	84	36	6900	14
Bicycling	1	2100	4000	11	0

References

1. Physical Activity Council. *2014 Participation Report.* Available at: http://www.physicalactivitycouncil.com/PDFs/current.pdf - PDF
2. Keadle, Sarah Kozey, et al. "A Framework to Evaluate Devices That Assess Physical Behavior." *Exercise and sport sciences reviews* 47.4 (2019): 206-214.

3. Bergmeir, Christoph, Rob J. Hyndman, and Bonsoo Koo. "A note on the validity of cross-validation for evaluating autoregressive time series prediction." *Computational Statistics & Data Analysis* 120 (2018): 70-83.
4. Sherstinsky, Alex. “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network.” Physica D: Nonlinear Phenomena 404 (2020): 132306. Crossref. Web