AI-Enabled Micro Motion Sensors for Revealing the Random Daily Activities of Caged Mice

Yifan Liu, Meng Chen, Chishing Chan, Ho-yin Chan, Jianping Wang, Xinge Yu, Xinyue Li, and Wen Jung Li*

More than 120 million mice and rats are used yearly for scientific purposes. While tracking their motion behaviors has been an essential issue for the past decade, present techniques, such as video-tracking and IMU-tracking have considerable problems, including requiring a complex setup or relatively large IMU modules that cause stress to the animals. Here, we introduce a wireless IoT motion sensor (i.e., weighing only 2 g) that can be attached and carried by mice to collect motion data continuously for several days. We also introduce a combined segmentation method and an imbalanced learning process that are critical for enabling the recognition of common but random mouse behaviors (i.e., resting, walking, rearing, digging, eating, grooming, drinking water, and scratching) in cages with a macro-recall of 94.55%. An interactive preprint version of the article can be found at: https://doi.org/10.22541/au.166005321.10787501/v1.

1. Introduction

Because their genome is similar to those of humans,[1–3] mice have been widely used in the preclinical stage of drug/vaccine discovery and mammalian organ injury and recovery mechanism research.[4,5] Mouse responses could help researchers predict drug effects in humans.[6] Moreover, mouse responses could also help understand different genome expressions.[7,8] Hence, the efficacy and safety of drugs or genome expressions would often require continuous long-term monitoring of mouse behaviors and responses through live and recorded videos.[9,10] Thus, prolonged observation, such as a behavioral test, is necessary since it would be utilized to measure the recovery or injury level of the mice. Preclinical trials and behavior tests on mice are widely used.[11] However, there is an extremely critical problem: it is highly time-consuming for researchers to study the recorded videos and observe and analyze mouse behaviors and responses.

With the rise of machine learning and behavior recognition algorithms, the above problem could be effectively solved if the entire monitoring process of mice could be automated without much human involvement. One commonly used approach is to apply computer vision techniques to analyze the recorded videos and classify mouse behaviors.[12,13] Another approach, which is discussed in this paper, is to capture mouse motion (e.g., x, y, z-accelerations and angular velocities) directly by micro-inertial measurement unit (μIMU) sensors. Due to the advancements in microelectromechanical systems (MEMS) technology in the past two decades, the small-size and considerably low-cost μIMUs have led to the popularity of applying these sensing devices in sectors such as robotics, sports, and navigation.[14,15] These compact sensing devices have also been widely utilized in animal studies for drug discovery and monitoring animal health.

Thirteen recent animal behavior research[16–27] involving twelve different animals of diverse physical sizes with μIMUs are shown in Figure 1. The twelve animals are subdivided into three categories according to their weight. Most of the μIMUs have been deployed on relatively large animals (i.e., canine and broiler) and megafauna[28] (i.e., elephant and cow), and only a few studies have been conducted with small lab animals such as guinea pigs and mice. Venkatraman et al.[16] built a wireless accelerometer and tested it on rats (in 2007), which could recognize three behaviors of rats (i.e., eating, standing, and grooming).
with an average recall of approximately 94.3%. However, the sensor size was large (32 mm x 25 mm) and heavy (10.2 g, i.e., 25–33% of a typical full-grown laboratory mouse). Then, in 2010, Venkatraman et al. designed a headstage that contained an accelerometer to measure an animal’s acceleration, but it needed a connected cable to transmit collected data. Currently, the major limitation in applying μMUs for small animal motion tracking and recognition is still the overall size and running time of the sensing device. However, with the continual advancement of microfabrication technology, much smaller and lower power-consuming μMUs sensors with wireless data transmission are now available. Thus, by applying artificial intelligence (AI) or machine learning algorithms to process the data from these new class of “Internet of Things” (IoT) sensors, we envision a new paradigm shift for tracking and recognizing small animal motions—the AI-enabled micro motion sensors could supplement or replace vision-based tracking in the near future due to their small size, lightweight, low-cost, and increased functionality in providing real-time motion recognition results without considering ambient lighting conditions.

Our group has recently developed a wireless IoT sensor-based system for laboratory mice motion data collection. By rudimentary data analyses, we were able to classify five behaviors, including resting, walking, rearing, digging, and shaking, of mice using a support vector machine (SVM) with an average recall of 76.23%. In this work, we discuss the extremely important work of selecting the proper AI algorithms in order to enable small motion sensor data to be applicable for real-time tracking and recognition of small animal motions. For example, a mouse’s motions in a cage over several days are random and with unevenly distributed temporal durations; hence, the proper segmentation and imbalanced data set learning algorithms must be utilized. We will show that using a combined segmentation method and an imbalanced learning process, the recognition of common but random mouse behaviors (i.e., resting, walking, rearing, digging, eating, grooming, drinking water, and scratching) in cages is possible with a macro-recall of 94.55%.

The importance of an appropriate data segmentation method has been emphasized before. There are two main segmentation categories: fixed-window segmentation and adaptive-window (dynamic-window) segmentation. In the fixed-window segmentation method (FWS), the window size is constant throughout the segmentation process. This method has been widely used in many previous studies on behavior recognition. Some researchers explored the impact of window size on activity recognition accuracy. Banos et al. concluded that the 1–2 s size would be a good trade-off between calculation speed and classification accuracy. In contrast, the window size in the adaptive-window segmentation varies with input signals by deciding turning points between different behavior signals. Nearly all the existing methods of adaptive-window algorithms have only been applied to deciphering human motions. Most adaptive window segmentations are inspired by detecting change points along with the signal, which is utilized as the boundaries between behaviors. Hong et al. proposed a dynamic sliding window approach in which window boundaries are created when important events, such as sensor-state changings occur. Their team tested that approach on two public datasets, the Kasteren Dataset and Patterson Dataset. The optimal result recall values using the dynamic window approach are 91.38% and 96.76%, which are much higher than those for the static sliding window approach (80.08% and 86.57%).

The systems designed by Sheng et al. and Noor et al. were more relevant. Sheng et al. developed an adaptive time window method for quasi-periodic human activities (walking, running, upstairs, and downstairs). For quasi-periodic activities, it uses the normalized autocorrelation method to determine the period for those behaviors. Noor et al. built an adaptive sliding window segmentation model with a transition behavior detector that could recognize ten behaviors (one dynamic, four static, and...
five transitional). The adaptive-window segmentation method was inspired by the probability density function with a multivariate Gaussian distribution, and an additional transition behavior detector was created to process the transition behaviors. This method achieved an average recall of 95.38% on the internal University of Auckland (UoA) dataset and an overall recall of 96.5% for the public smartphone-based HAR (SBHAR) dataset.[44,45]

In this article, we present our latest development of a combined segmentation method for laboratory mouse common behavior recognition based on our μIMU-based wireless motion sensor. The entire data flow and structure of the tracking system are shown in Figure 2. In the data segmentation process, a combined segmentation method (CSM) consists of both fixed-window segmentation (FWS) and adaptive-window segmentation (AWS). AWS is designed for dynamic behaviors (i.e., walking, rearing, digging, grooming, and scratching), and the window size is adjusted according to the input signals. In contrast, FWS is designed for other static behaviors (i.e., resting, eating, and drinking water). A preliminary status classifier (PSC) is adopted to decide whether an input sequence of signals belongs to a static or dynamic status, which determines the appropriate segmentation method. After segmentation, all the segmented data are input into an SVM model after undergoing feature extraction and principal component analysis (PCA).

2. Results

2.1. Behavior Signal Analysis and PSC Results

The images of eight typical behaviors are shown in Figure 3A, and the corresponding acceleration data in the x-, y-, and z-axes...
are presented in Figure S1, Supporting Information. The eight mouse behaviors are divided into two statuses (i.e., static and dynamic) according to the mouse posture or position changes during the behavior. From the figure, it is evident that the acceleration changes are more significant in the five dynamic behaviors, while the acceleration values only slightly change near their average values in static behaviors. We display two or three behavior positions for the five dynamic behaviors to represent the complete behavior sequence.

The data processing ensues after data collection, including segmentation, feature extraction, and classification. After collecting signals of eight behaviors, we took the experiment to determine the relationship between the recognition accuracy and the segmentation window size of the FWS. Considering the conclusion from ref. [37] and the actual behavior time length of mice, our window size test ranges from 0.4 to 2 s without sliding. We use the F-score as the criterion for classification accuracy. The relationship between window size and the F-score for eight behaviors is shown in Figure 3B.

The maximum average F-score of 0.8171 is obtained at 1.6 s; however, not all the behavior recognition accuracies reach their highest at that size. The effects of window size on static and dynamic behavior are quite different. Most of the static behaviors’ F-scores are over 0.9 and only change slightly when the window size increases. The F-score for dynamic behavior varies greatly as the window size increases, and there is no linear or other mathematical relationship between the window size and the accuracy. The five dynamic behaviors reach their maximum at five different window sizes. This result validates our hypothesis that different window sizes affect the precision of behavior classification and that FWS with one window size is NOT appropriate for all behaviors. Therefore, AWS should be applied to dynamic behaviors to find an appropriate segmentation process.

Then, the preliminary status classifier (PSC) is designed to primarily classify the behavior status according to the data changing trend. We calculate the FFT energy and acceleration variance in the 2-second window, consistent with the PSC window, to quantify the changing trends for each behavior type. Figure 4A,B show the distribution of FFT energy and acceleration variance. The FFT energy and acceleration variance of static behaviors (i.e., resting, eating, and drinking water) are much lower than those of the other five dynamic behaviors (i.e., walking, rearing, digging, grooming, and scratching). These obvious value differences make it easier for the PSC to separate the two categories, except that some abnormal values might cause a classification error. We choose the random forest as the preliminary classifier and use four-fold cross-validation to evaluate the performance. The result of PSC is shown in Figure 4C. The total recognition accuracy is over 95%, which

Figure 3. Eight behaviors and SVM classification results based on FWS. A) Images and acceleration signals for eight behavior signals. B) Relationship between F-score and window size (in seconds) for FWS.
means that the preliminary status classifier performs appropriately, and most of the signal is segmented by a suitable segmentation method.

2.2. Final Behavior Classification Results

We constructed five segmentation methods (FWS, BOCDS, PDFS, CSM-1, and CSM-2) and tested them, then compared the performance of the five segmentation methods. The window size in FWS was set as 1.6 s as the optimal for eight behaviors, and the FWS size used in CSM-1 and CMS-2 was 1 s due to its optimal performance for static behaviors. We used the recall value for each behavior to represent its classification performance and the average recall to describe the overall performance of model training and validation. The details of the final behavior classification result are shown in Figure 5.

The CSM-2 method achieved a 92.06% average recall, approximately a 12.02% increase compared with FWS. Most behaviors could be classified with a recall value higher than 88%. Although the CSM-1 method was relatively poor in terms of average recall, it achieved much higher scratching performance. This figure shows that the combined segmentation method performs better than any single segmentation method, and almost all the maximum recall values for each behavior are obtained in the two CSMs.

Figure 4. Preliminary status classifier and combined segmentation result. A) The acceleration variance box plot for eight behaviors. B) The FFT energy box plot for eight behaviors. C) The classification result of the preliminary status classifier.

Figure 5. Comparison of six segmentation and classification methods on recall values with four-fold cross-validation.
The influence of a segmentation method difference on static and dynamic behaviors is dissimilar, and the classification accuracy of static behaviors in both CSM-1 and CSM-2 changed slightly compared to FWS. The recall value increase in dynamic behaviors is apparent, and the maximum increase is achieved in walking with CSM-2 at 29.84%. The maximum increasing value of the five behaviors was achieved with CSM-2 at 18.28% on average. As previously demonstrated, we were curious whether combining CSM and imbalanced learning would increase classification accuracy. The ClusterCentroids method was used to reduce the sample size of the large-sample-size behavior, and then the reduced feature samples were taken to train the SVM model and to be validated. After implementing the imbalanced learning steps, the classification result of each behavior class is also shown in Figure 5.

The chosen CSM was CSM-2 (PSC+FWS+BOCDS) due to its optimal performance discussed above, and imbalanced learning was utilized to remove samples from four major behavior classes (i.e., Resting, Eating, Walking, and Rearing). After combining CSM-2 with the imbalanced learning algorithm, the macro-recall of the eight behaviors can be improved to 94.55%, an increase of 14.51% compared with CSM-2, but the performance of each behavior class was dissimilar. The recall value of three static behaviors and walking decreased, while the other four dynamic behaviors largely increased. This situation occurred because the data sample ratio between eight behaviors changed after the under-sampling.

3. Discussion

When attaching our motion sensor to the mouse, to prevent the sensor from direct contact with the mouse body and to serve as a mounting surface for the sensor, a flexible rubber tubing made of elastic rubber was first wrapped around the mouse’s neck. The sensor was wrapped by a piece of parafilm to seal the electronic parts. Then, our sensor was attached to the flexible tubing at the mouse’s back position by tape. The orientation of the sensor was kept the same during the experiments, i.e., the x-axis of the sensor is along with the mouse’s spine and the y-axis is parallel with a virtual line connecting the two ears of the mouse. From our observations, compared with mice without the attached sensor, the frequency of scratching would initially increase when the sensor is attached to a mouse. But the frequency of scratching would decrease to the normal level after two to three hours as the mouse became accustomed to the attached sensor and stopped trying to remove the sensor.

In the final classification, although CSM-1 and CSM-2 have already achieved higher classification accuracy, the reasons for AWS’s better performance in analyzing dynamic behavior signals and for FWS’s suitability in analyzing static behavior signals must be further investigated. Two examples demonstrating the difference in IMU data segmentation and final classification among the FWS, BOCDS, and PDFS approaches are shown in Figure 6. The total acceleration, its corresponding label, the segmentation boundary of the three methods, and the final classification error are displayed in both examples. The first test is based on a static behavior signal, i.e., a 16-s sequence of collected mouse sensor data. According to the F-score in Figure 3, the FWS window size is chosen as 1.6 s, and the entire signal is segmented into 12 slices by FWS. The slice number of BOCDS is smaller (i.e., 8), and PDFS’s number is larger (i.e., 27). Then, the slices are extracted to feature vectors and are classified in the following steps. As the entire static signal is labeled “Eating,” the correctness of the segmentation inside the “Eating” behavior is difficult to evaluate. Therefore, we only consider the final prediction result as the metric for the three segmentation methods. The prediction error is chosen as the assessment metric, which is defined as follows

$$\text{ERR}_{\text{predict}} = \sum_{j=1}^{n} L_{err}(j)$$

where \(\text{ERR}_{\text{predict}}\) is the total prediction error, \(n\) is the number of false predicted slices, and \(L_{err}(j)\) is the length of the \(j\)th false predicted slice.

In the final classification result shown in Figure 6C, the corresponding prediction error of FWS is much smaller than that of the two AWS methods, which means that FWS is more appropriate for static signals. The reasons behind false predictions are different in PDFS and BOCDS. The slices of PDFS are shorter than slices of FWS, and some of the slices of BOCDS are much longer. Shorter or longer slices can influence the prediction by affecting the calculated feature vectors. For example, the long slices of “Eating” can be predicted as “Scratching,” and the short slices might be predicted as “Resting.” It could also explain the low classification recall of BOCDS and PDFS for the behavior “Eating” in Figure 5.

The other test is conducted on a 15.6-s dynamic signal sequence that contains six mouse behaviors. The red dotted lines in the figures are the actual behavior segmentation boundary generated by human labeling. The entire signal is segmented into 10 slices by FWS, while the signal is divided into more slices by BOCDS and PDFS (i.e., 25 and 18 respectively). In the segmentation result of FWS, many slices contain sharp signal changes in the middle, which usually indicates the start of new behavior and should be taken as behavior boundaries. Using the BOCDS or PDFS approach would provide more slices and window boundaries that are closer to the actual behavior interval boundaries. We define the segmentation error as the number of reading differences between the segmentation boundary and the nearest human-labeled boundary, which is expressed as follows

$$\text{ERR}_{\text{seg}} = \sum_{i=1}^{n} B_i - H_{\text{nearest}}$$

where \(\text{ERR}_{\text{seg}}\) is the total segmentation error, \(B_i\) is the reading position of the \(i\)th segmentation boundary, and \(H_{\text{nearest}}\) is the human-labeled behavior boundary, which is nearest to \(B_i\). The segmentation error values for BOCDS and PDFS are much less than those in FWS, and the two AWS approaches are more sensitive to these changes among the signals. In addition, the prediction errors of the two AWSs are smaller than the value of the FWS, and the two AWSs both perform better in the dynamic signal test.
Figure 6. Comparison between three different segmentation methods. A) The performance of three segmentation methods (FWS, BOCDS, and PDFS) on static behavior signal sequences. B) The performance of three segmentation methods (FWS, BOCDS, and PDFS) on dynamic behavior signal sequences.
4. Conclusion

A smart and lightweight (i.e., weighs ≈5–6.6% of typical laboratory mice) wireless μIMU sensing device is developed and attached to laboratory mice to collect their motion data over several days continuously. To overcome the shortcomings of using fixed time-windows for data analysis, a new combined segmentation method is introduced, consisting of fixed-window segmentation, adaptive-window segmentations, and one preliminary status classifier for mouse behavior recognition. By applying this method to generate segmented slices of eight behaviors (resting, walking, rearing, digging, eating, grooming, drinking water, and scratching), the classification accuracy with SVM is significantly improved to 92.06% from 80.04%, which is based on the traditional fixed-window segmentation method. After combining the imbalanced learning techniques, the average recall reaches 94.55%. In addition, the average recall is even higher (i.e., 98.09%) for less frequent behaviors, such as rearing, digging, grooming, drinking water, and scratching.

5. Experimental Section

Experiment Setup: Our μIMU sensor in Figure 2 consists of a Bluetooth System on a Chip (i.e., DA14583), an IMU chip (i.e., Bosch BMI160), an antenna, a PCB board, and a button cell. The system consists of a cage, a monitoring camera for recording mouse behaviors, a Wi-Fi router for network connection, an IMU sensor for recording mouse behavior data, and a network-connected storage device for storing video and sensor data. In Figure S2A, Supporting Information, the experimental design is represented. The sensor is taped to the mice’s bodies above a layer of flexible tubing, and its orientation is constant throughout all experiments because variations in orientation would affect the subsequent process’s ability to recognize behavior accurately. The size of the μIMU is only 12 mm × 7 mm × 5 mm (weighing ≈2 g), which is relatively small to ensure that the behavior of the mice would not be affected. The sensor collects the acceleration and angular velocity at 25 Hz; hence, a 25 × 6 matrix of data can represent the behavior of mice in a second. Figure S2B, Supporting Information, shows 5-s collected IMU readings.

Mice (CS7BL6/J) are placed into a mause cage to capture their motions using our tracking device. The entire experimental procedures (Internal Reference No. A-O358) were approved by the City University of Hong Kong’s Animal Research Ethics Sub-Committee and were in accordance with the guidelines and policies in Hong Kong’s Code of Practice for Care and Use of Animals for Experimental Purposes. The food and water are both placed on the top of the cage, and the mouse inside needs to stand up (i.e., rearing) before it can obtain food and water. The cage bottom is deeply lined with bentonite mouse litter to keep the environment clean and allow the mouse to dig into the bedding. Cameras are set on the side of the cages to record the mouse’s behavior day and night. The recorded videos are used to manually label mouse behaviors, including resting, walking, rearing, digging, eating, grooming, drinking water, and scratching. Sample videos of these behavior motions are included in the Supporting Information. The details of the eight behaviors are also shown in Table S1, Supporting Information, of the section. The corresponding start and stop times of each behavior are recorded. This information will be used for later supervised machine learning, as well as for classification model performance validation. The sensor data used in the following data analysis section was collected from four experiments with a total time of 29 h and 4 min.

Combined Segmentation Method (CSM): A key contribution of this work is the combined segmentation method (CSM), which manages data processing, behavior status classification, and signal segmentation. CSM has two major segmentation methods: traditional fixed-window segmentation (FWS) and adaptive-window segmentation (AWS). Probability density function segmentation (PDFS) and Bayesian online change point detection segmentation (BOCDS) are the two techniques used by AWS. The working principle of CSM is shown in Figure S3, Supporting Information. We assume a behavior signal sequence in the figure contains one static and three dynamic behaviors. In CSM, a two-second-window signal is first analyzed using a preliminary status classifier (PSC), which determines the status of the sample. The status can be either static or dynamic. The FWS is assigned to the sample if its status is found to be static. The subsequent two-second-window data from the endpoint of this sample is the next input into PSC after the two-second-window signal has been divided into two one-second slices. Otherwise, the sample status can be dynamic, and AWS can be assigned to it. AWS divides the sample into slices according to several calculated changepoints. Then, the last changepoint inside the sample is taken as the starting point of the next two-second-window signal.

The segmented results using only FWS and our CSM are plotted in Figure S3, Supporting Information. It is apparent that the signal would likely be segmented into more slices in CSM than in FWS. Static behavior 1 and dynamics 1 and 2 are divided into 7.4 slices in the FWS and 9 slices in the CSM. When FWS is applied to the whole signal, segmentation errors occur for slices FW2, FW6, and FW8. These static-dynamic-mixed slices can directly affect the following feature extraction process. The segmentation might directly interfere with the final classification, and the feature vectors. However, for CSM, the AWS would stop at the behavior signal sequence boundary. Thus, when using our CSM, the chance of segmentation error occurring is greatly reduced.

Combined Segmentation Method (CSM): Preliminary Status Classifier (PSC): Our mouse behavior classification system is designed to classify eight mouse behaviors: resting, walking, rearing, digging, eating, grooming, drinking water, and Scratching. These eight types of mouse behaviors are separated into two statuses (static and dynamic) according to mouse body movement. Static status includes resting, eating, and drinking water; the other five behaviors are marked as dynamic. The sensor signal of mice is almost constant for static behaviors; however, the signal oscillates rapidly and largely for dynamic behavior, and any FWS error might cause false recognition. Due to this consideration, the FWS is assigned to static behavior signals, and the AWS deals with dynamic behavior signals. Therefore, an additional preliminary status classifier (PSC) is designed for initial signal sequence classification. This classifier is used to identify whether a signal sequence is suitable for either AWS or FWS. The classifier utilizes two parameters as inputs, one of which is the fast Fourier transform (FFT) energy, and the other parameter is the acceleration variance of the signal sequence.

We choose the random forest as the preliminary classifier and use 75% of the dataset as the training dataset to train the random forest model and the remaining 25% as the validation dataset. It can classify static and dynamic behaviors accurately with high speed. Considering that the data length would highly affect the two input features, we fixed the size of the input signal sequence as 2 s.

Combined Segmentation Method (CSM): Segmentation Method: As discussed above, several segmentation methods were applied to analyze the mice’s behavioral data, including FWS, PDFS, BOCDS, and two other combined segmentation methods (i.e., CSM-1 and CSM-2). The mathematical representations of these segmentation methods are discussed in the Supporting Information.

Feature Extraction and Model Training and Validation: The following process after sample segmentation is feature extraction, which refers to the data transformation technique performed on the slices. The raw data slices are unsuitable for direct use by conventional machine learning algorithms, such as naive Bayes classifiers, SVMs, and decision trees. Therefore, feature extraction is necessary, and it can reduce the segment data dimension without losing information and increase the computing speed of the final classification.

For each slice after the CSM, six time-domain features are calculated for each axis of the slice, and five frequency-domain features are extracted for total acceleration and angular velocity. Thus, 38 features, which form a 1 × 38 vector and represent the segmented sample, are extracted from the six axes slice. The details and equations of the 38 features are shown.
in Table S2, Supporting Information. The features are used for the following model training and validation.

Before putting a feature vector into the training model, we applied principal component analysis (PCA)\(^{[47]}\) to reduce its dimensionality. It would rearrange its data representation to increase its interpretability without losing information. We selected the support vector model (SVM) for the final supervised learning model due to its satisfactory performance in higher-dimensional space. This work used fourfold cross-validation to evaluate the classification model performance. The collected behavior samples were randomly split into four groups of the same size. Then, three groups of datasets were used for model training, and the other was used for model validation. This step was repeated until four groups were included in the validation set. Then, a confusion matrix was calculated to evaluate the performance of our trained model, along with another three metrics. The definitions of these metrics are provided in the Supporting Information.\(^{[48–50]}\)

Due to the sample size imbalance between different behaviors, training an effective classification model that can recognize both large-sample-size mouse behavior (i.e., Resting, Eating) and small-sample-size behavior (i.e., Drinking, Scratching) is difficult. In our previous work,\(^{[29]}\) Chen et al. proposed and proved the effectiveness of an imbalanced learning method to improve classification accuracy. This paper also applied the imbalanced learning method during the classification process. The ClusterCentroids algorithm was adapted to perform the under-sampling function. It would remove some data samples from the large-sample-size behaviors to maintain a balance between each behavior class.

Supporting Information
Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements
The work described in this article was partially supported by grants from the Hong Kong Research Grants Council (project nos. T42-717/20-R, C7174-20C, and 11207222) and the Science and Technology Innovation Commission of Shenzhen Municipality (SZSTI) (project no. SGDX201908162312725). The Interactive Supporting Information of this article can be found at https://doi.org/10.22541/au.166005178.87001477/v1. Corrections added on April 21st, 2023 after online publication: Ho-yin Chan’s name and the copyright year were corrected.

Conflict of Interest
The authors declare no conflict of interest.

Author Contributions
Y.L. and M.C. equally contributed to this work. W.J.L., H.Y.C., and C.C. conceived the project; W.J.L. supervised the research with input from X.Y. on sensor and power unit design; M.C. designed and built the pMU motion tracking system and conducted the experiments; Y.L. analyzed and interpreted the data; J.W., X.Y., and X.L. provided advise on applying the machine learning algorithms. Y.L. wrote the manuscript with significant input from M.C.; all authors critically revised the manuscript.

Data Availability Statement
The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords
adaptive window segmentation, animal behavior, behavior recognition, machine learning, micro sensor

Received: July 20, 2022
Revised: September 28, 2022
Published online: January 31, 2023


