

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

Yifan Liu¹, Meng Chen¹, Chishing Chan², Ho-yin Chan¹, Jianping Wang¹, Xinge Yu¹, Xinyue Li¹, and Wen Jung Li¹

¹City University of Hong Kong

²CASCUBE Limited

February 22, 2024

Abstract

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 grams) 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%.

Corresponding author(s) Email: hoychan@cityu.edu.hk; xinyueli@cityu.edu.hk; wenjli@cityu.edu.hk

ToC Figure

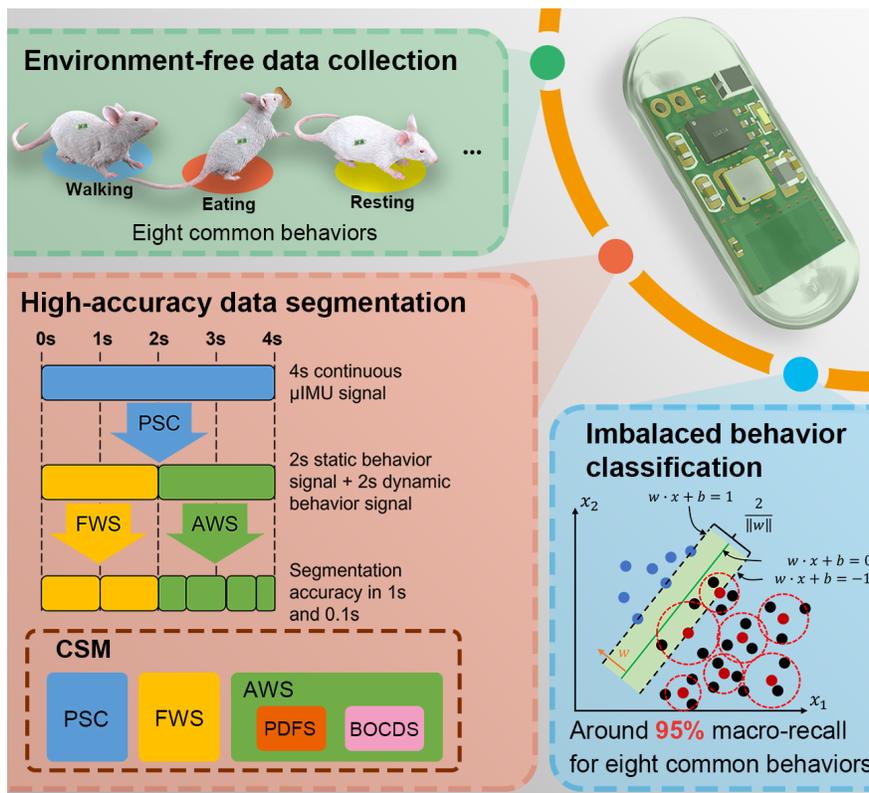


Figure 1: **ToC Figure.** A wireless IoT motion sensor (weighing only 2 grams; 5 grams including a coin-cell battery) is demonstrated for continuously collecting mice motion data without a complex setup. Furthermore, a combined segmentation method and an imbalanced learning process with higher segmentation and classification accuracy are presented. The final macro-recall of eight common behaviors in the cage is 94.55 percent.

Introduction

Because their genome is similar to those of humans (Mural et al., 2002; Dawson, 2011; Yue et al., 2014), mice have been widely used in the preclinical stage of drug/vaccine discovery and mammalian organ injury and recovery mechanism research (Li et al., 2004; Nakamura et al., 2004). Mouse responses could help researchers predict drug effects in humans (Hoehndorf et al., 2013). Moreover, mouse responses could also help understand different genome expressions (Sasamura et al., 1992; Perrin, 2014). Hence, the efficacy and safety of drugs would often require continuous long-term monitoring of mouse behaviors and responses through live and recorded videos (Zhang et al., 2012). 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. 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. 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 (IMUs) sensors. Due to the advancements in microelectromechanical systems (MEMS) technology in the past two decades, the small-size and considerably low-cost IMUs have led to the popularity of applying these sensing devices in sectors such as robotics, sports, and navigation(Ahmad et al., 2013). These compact sensing devices have also been widely utilized in animal studies for drug discovery and monitoring animal health.

Thirteen recent animal behavior research(Venkatraman et al., 2007; Venkatraman et al., 2010; Soltis et al., 2012; Wang et al., 2015; Hammond et al., 2016; Krschel et al., 2017; Barwick et al., 2018; Wang et al., 2018; Kumpulainen et al., 2018; Kadar et al., 2020; Eerdekens et al., 2021; Yang et al., 2021) involving twelve different animals of diverse physical sizes with IMUs are shown in Figure 2. The twelve animals are subdivided into three categories according to their weight. Most of the IMUs have been deployed on relatively large animals (i.e., canine and broiler) and megafauna(STUART, 1991) (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.(Venkatraman et al., 2007) 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.(Venkatraman et al., 2010) 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 IMUs 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 IMU 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, light weight, 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(Chen et al., 2022). 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%.

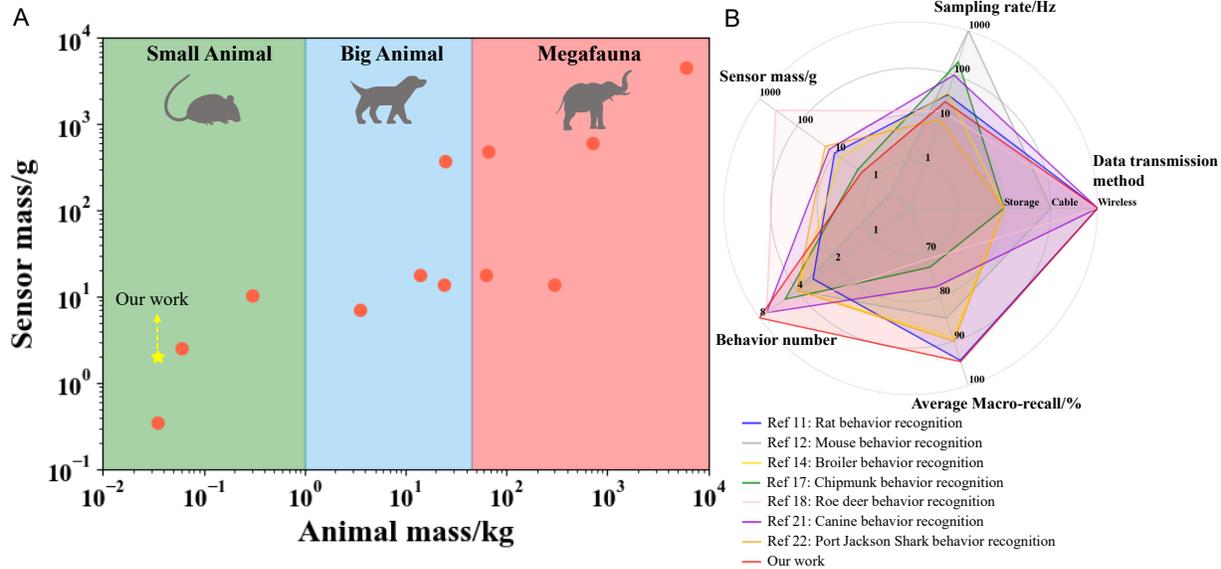


Figure 2: **IMU-based animal behavior recognition work review.** (A) Relationship between the animal mass and sensor mass in previous animal behavior studies. (B) Comparison between eight small animal and big animal behavior studies

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 behavior recognition research (Brugarolas et al., 2013; Winters et al., 2015; Gutierrez-Galan et al., 2018; Schlecht et al., 2004; Ungar et al., 2005; Nakamoto et al., 1999; Lavanya & Mallappa, 2018). Some researchers explored the impact of window size on activity recognition accuracy. Banos et al. (Banos et al., 2014) concluded that the 1-2 s size would be a good trade-off between speed and 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. In (Hong & Nugent, 2009), Hong et al. adapted the location context to determine the breakpoints. In this method, the changes in the location context in sensors indicate activity changes. This method is effective when various activities are related to different locations. In (Laguna et al., 2011), Ortiz 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 (van Kasteren et al., 2008) and Patterson Dataset (Patterson et al., 2005). 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. (Sheng et al., 2015) and Noor et al. (Noor et al., 2017) 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 (Anguita et al., 2013) and an overall recall of 96.5% for the public smartphone-based HAR (SBHAR) dataset (Reyes-Ortiz et al., 2016).

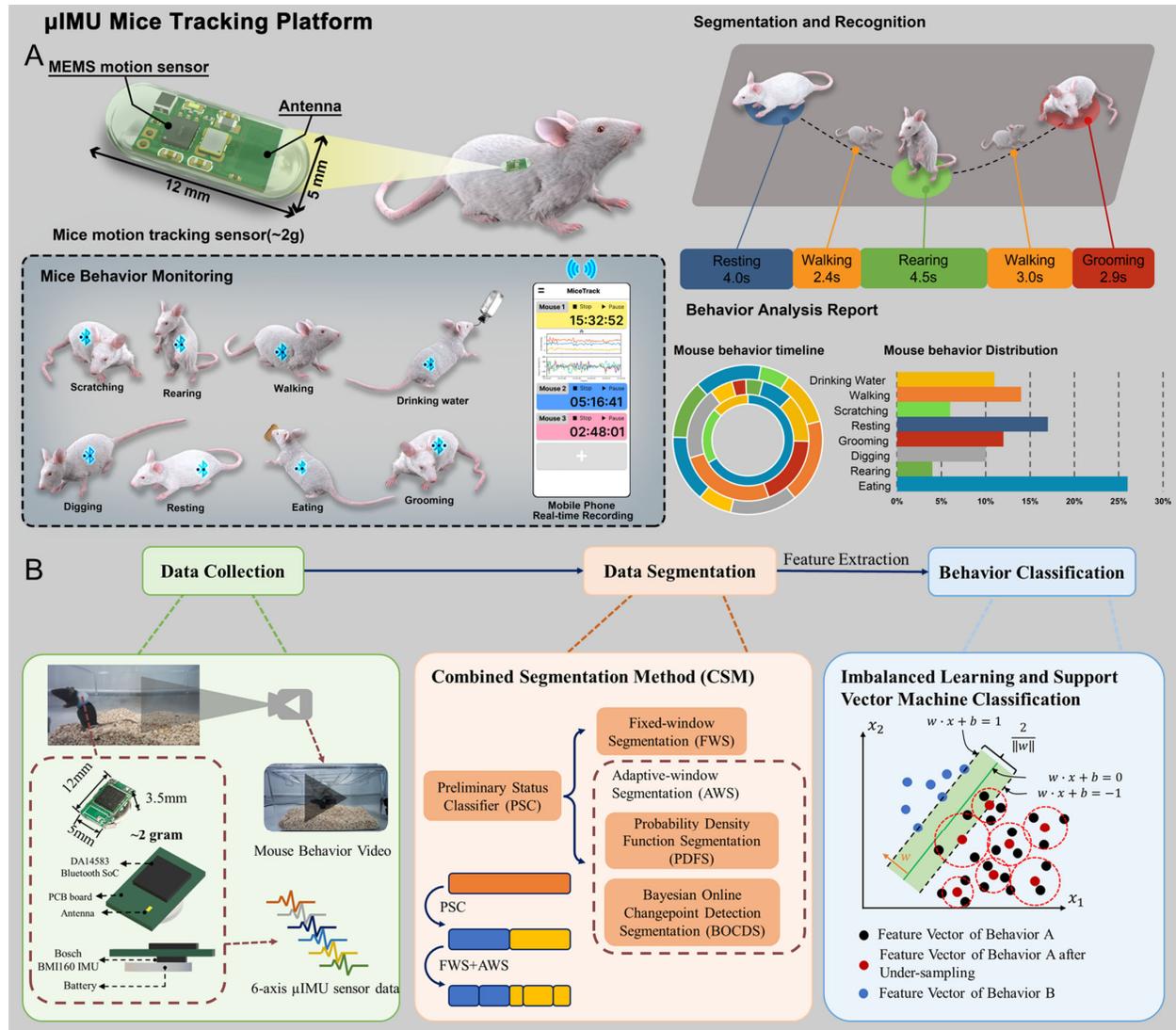


Figure 3: **Conceptual illustration of an attachable IMU-based mouse tracking system.** (A) Conceptual illustration (B) Data flow of the mouse tracking system

In this paper, we present our latest development of a combined segmentation method for laboratory mouse common behavior recognition based on our IMU-based wireless motion sensor. The entire data flow and structure of the tracking system are shown in Figure 3. 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).

Experimental Section/Methods

Experiment setup

Our IMU sensor in Figure 3 consists of a Bluetooth System on a Chip (i.e., Dialog Semiconductor DA14583), an IMU chip (i.e., Bosch BMI160), an antenna, a PCB board, and a button cell. Basically, the system contains one cage, one monitoring camera that is used to record the mouse behaviors, one Wi-Fi router providing the network connection, one IMU sensor recording the mouse behavior data, and one network-connected storage storing the video and sensor data. The experiment setup is shown in Figure S1A. The sensor is attached to the mice by tape, and its orientation is the same during all the experiments since the different orientations would affect the recognition accuracy in the following process. The size of the IMU is only 12 mm x 7 mm x 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 25x6 matrix of data can represent the behavior of mice in a second. Figure S1B shows 5-second collected IMU readings.

Mice (C57BL/65) are placed into a mouse cage to capture their motions using our tracking device. The entire experimental procedures (Internal Reference No. A-0358) 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 Section. The details of the eight behaviors are also shown in Table S1 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 following data analysis section was collected from 4 experiments with a total time of 29 hours and 4 minutes.

Combined Segmentation Method (CSM)

The combined segmentation method (CSM), a core contribution of this work, handles data processing, including behavior status classification and signal segmentation. CSM has two major segmentation methods: traditional fixed-window segmentation (FWS) and adaptive-window segmentation (AWS). AWS consists of two methods: probability density function segmentation (PDFS) and Bayesian online change point detection segmentation (BOCDS). The working principle of CSM is shown in Figure S2. 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. If the sample status is determined to be static, the FWS is assigned to it. The two-second-window signal can be segmented into two one-second slices, and then the subsequent two-second-window data from the endpoint of this sample is the next input into PSC. 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 S2. 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 indirectly interfere with the final classification through 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.

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 seconds.

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 Section.

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(Lima et al., 2019).

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 x 38 vector and represent the segmented sample, are extracted from the six axes slice. The details of the 38 features are shown in Table S2 in the 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)(Abdi & Williams, 2010) 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 4-fold 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 Section.

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(Chen et al., 2022), 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.

Results

Behavior signal analysis and PSC results

The images of eight typical behaviors and the corresponding acceleration data in the x-, y-, and z-axes are presented in Figure 4A. 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.

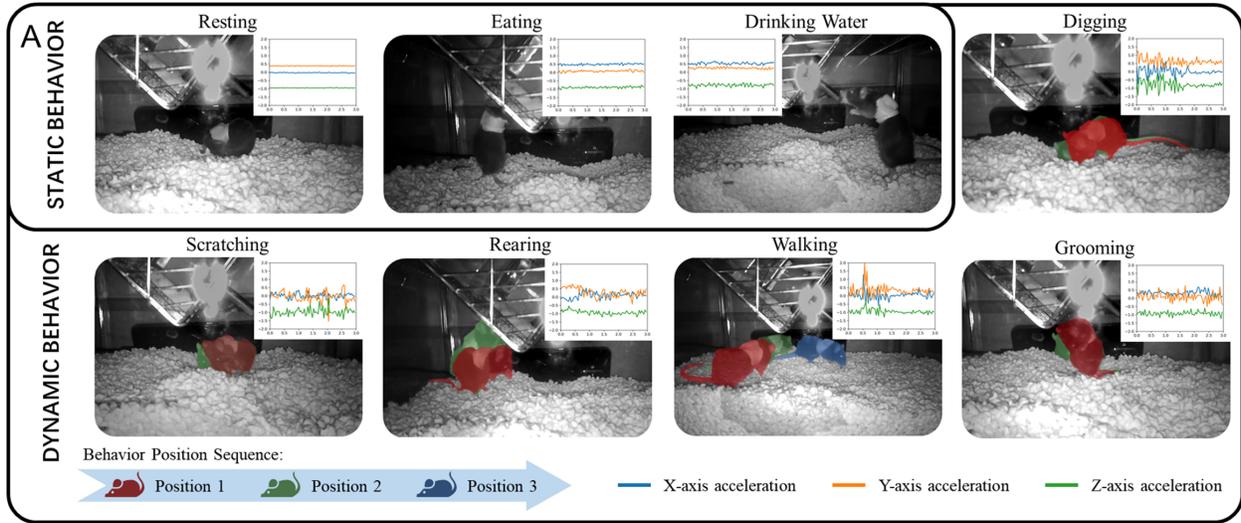


Figure 4: **Typical acceleration data for eight behaviors and SVM classification result based on FWS.** (A) Images and acceleration signals for eight behavior signals. (B) Relationship between F-score and window size (in seconds) for FWS.

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 from32 and the actual behavior time length of mice, our window size test ranges from 0.4 s 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 4B.

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 window size influences on static and dynamic behaviors 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 proves our estimation that different window sizes influence the behavior classification accuracy and that FWS with one window size is NOT suitable 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 5A and Figure 5B 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 4-fold cross-validation to evaluate the performance. The result of PSC is shown in Figure 5C. The total recognition accuracy is over 95%, which means that the preliminary status classifier performs appropriately, and most of the signal is segmented by a suitable segmentation method.

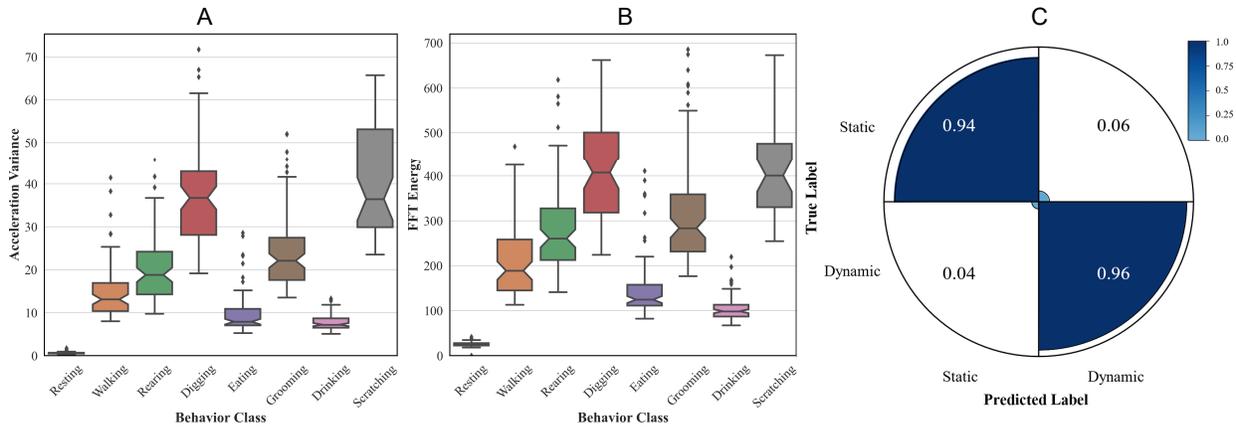


Figure 5: **Preliminary status classifier and combined segmentation result.** (A) The acceleration variance box plot for eight behaviors. (B) The FFT energy box plot for eight behaviors. (C) The classification result of the preliminary status classifier

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 seconds as the optimal for eight behaviors, and the FWS size used in CSM-1 and CMS-2 was 1 second 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 6.

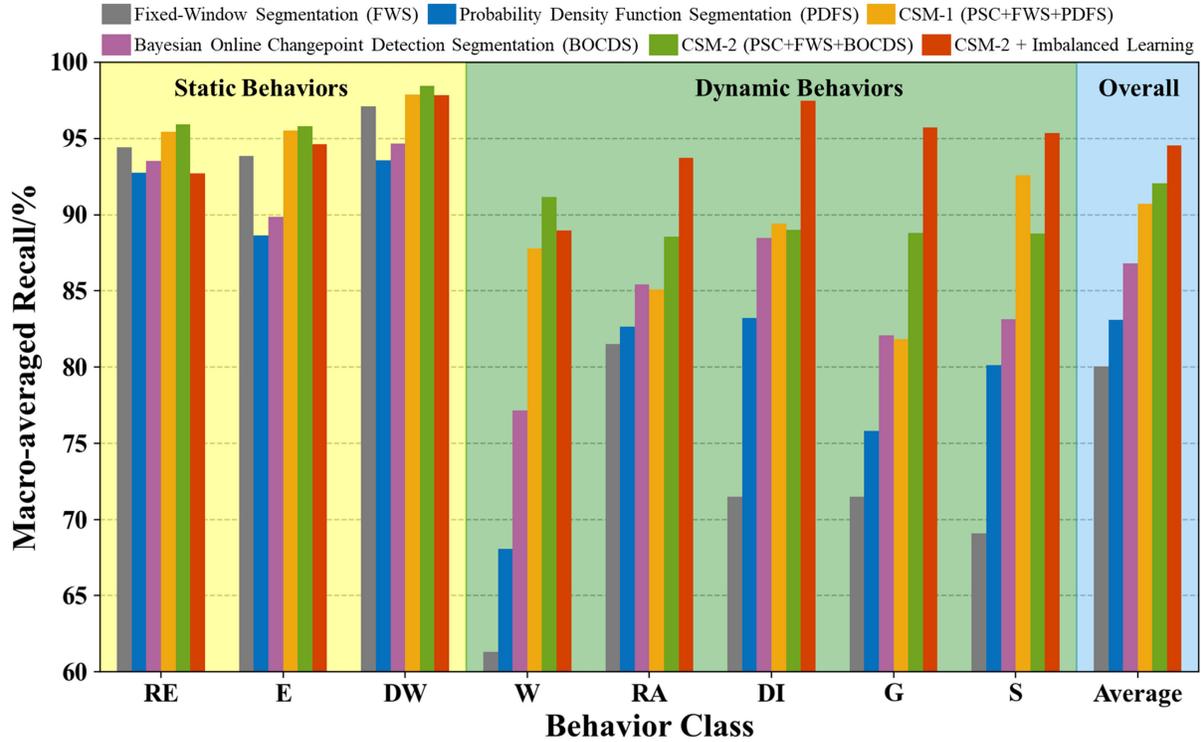


Figure 6: Comparison of six segmentation and classification methods on recall values with 4-fold cross validation

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 the 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.

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 6.

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 the FWS. The macro-recall value also increased by 2.49% 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.

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 the 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.

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 6.

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 the FWS. The macro-recall value also increased by 2.49% 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.

Discussion

Although CSM-1 and CSM-2 have already achieved higher classification accuracy, the reasons why AWS performed better in dynamic behavior signals and why FWS is more suitable for static behavior signals must be further analyzed. Two examples demonstrating the difference in IMU readings segmentation and final classification among the FWS, BOCDS, and PDFS approaches are shown in Figure 7. The total acceleration, its corresponding label, the segmentation boundary of the three methods, and the final classification error area are displayed in both examples. The first test is based on a static behavior signal, a 16-second sequence of collected mouse sensor data. According to the F-score in Figure 4, the FWS window size is chosen as 1.6 seconds, 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 challenging 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:

$$ERR_{predict} = \sum_{j=1}^m L_{err}(j) \quad (1)$$

where $ERR_{predict}$ is the total prediction error, m 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 7C, 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 prediction 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 6.

The other test is conducted on a 15.6-second 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:

$$ERR_{seg} = \sum_{i=1}^n (B_i - H_{nearest}) \quad (2)$$

where ERR_{seg} is the total segmentation error, B_i is the reading position of the i th segmentation boundary, and $H_{nearest}$ 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.

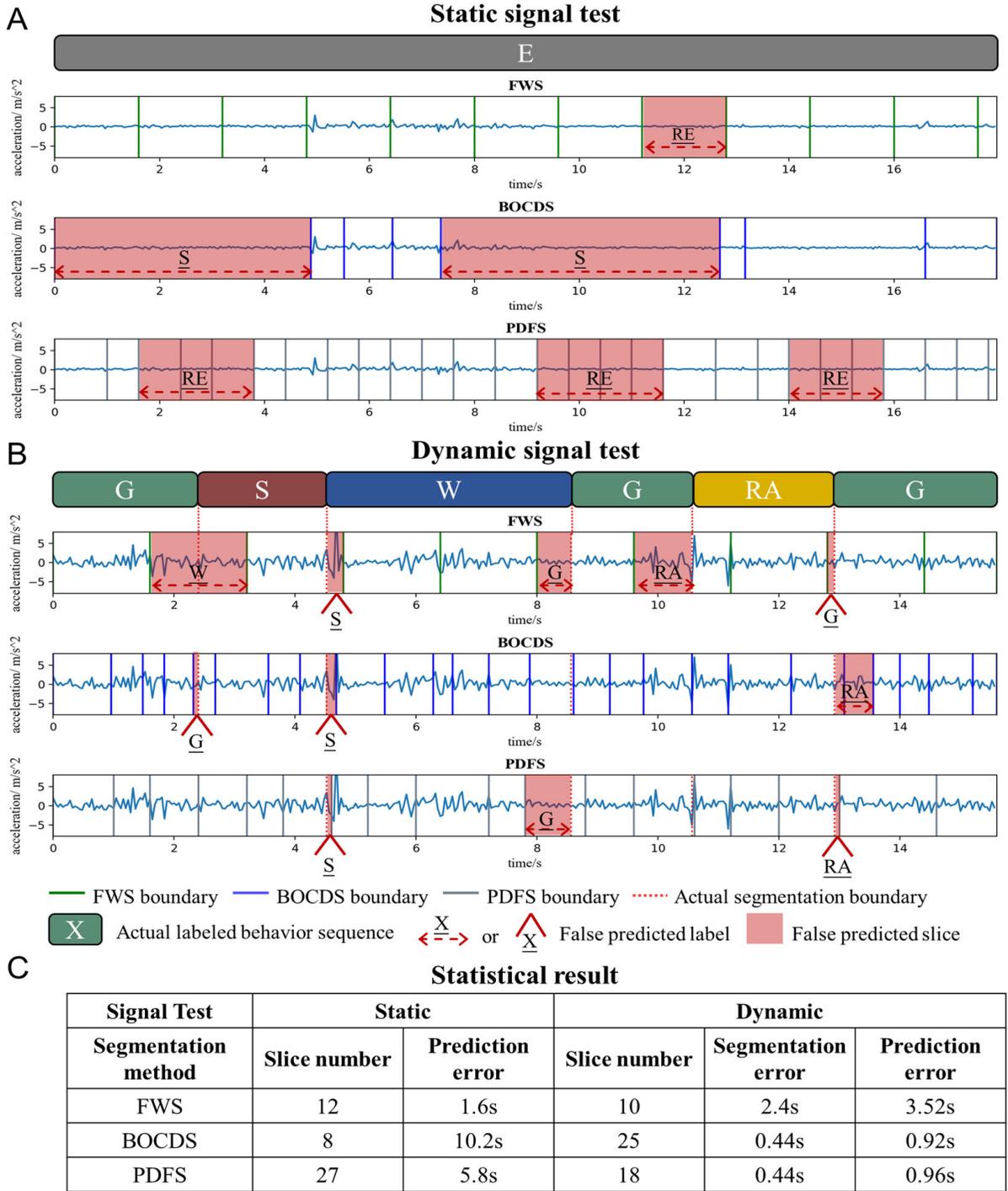


Figure 7: **Comparison between three different segmentation methods.** (A) The performance of three segmentation methods (FWS, BOCDS, and PDFS) on static behavior signal sequences. (B) The performance of three segmentation methods (FWS, BOCDS, and PDFS) on dynamic behavior signal sequences

Conclusion

A smart and lightweight (i.e., weighs 5 to 6.6% of typical laboratory mice) wireless IMU sensing device is developed and attached to laboratory mice to collect their motion data over several days continuously. Machine learning algorithms are then used to classify and recognize the daily activities of the mice. A new combined segmentation method is introduced, consisting of fixed-window segmentation, adaptive-window segmentations, and one preliminary status classifier for mouse behavior recognition. The raw signal of the mouse behavior collected by the IMU sensor is first analyzed by the preliminary classifier and then segmented by the combined segmentation method. The segmented slices are used in the feature extraction process. The final SVM model returns the recognition of the behavior using the feature vectors. Compared to the low performance of the traditional fixed-window segmentation method at only 80.04%, the combined segmentation method, especially CSM-2 (PSC+FWS+BOCDS), could increase the accuracy in the segmentation process, which can also result in a significant improvement in the final recognition result. The average recall of the combined segmentation method for eight behaviors (resting, walking, rearing, digging, eating, grooming, drinking water, and scratching) increases to 92.06% in the SVM model training process. After combining the imbalanced learning techniques, the average recall reaches 94.55%. In addition, the average recall is even higher for less frequent behaviors, such as rearing, digging, grooming, drinking water, and scratching, which achieves 98.09%.

Acknowledgements

Funding

The work described in this paper was partially supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. T42-717/20-R, No. JLFS/E104/18, and No. 11210819), and also in part by the Hong Kong Innovation Technology Commission (Project No. UIM/382).

Author contributions

W.J.L., H.Y.C., and C.C. conceived the project; W.J.L. supervised the research; M.C. designed the IMU motion tracking system and conducted the experiments; Y.L. analyzed and interpreted the data; J. W. 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 and material availability

All data needed to evaluate the conclusions are present in the paper and/or in the Supporting information. The data used for the plots in this paper and other findings of this study are available from the corresponding author upon reasonable request.

Conflict of interest

The authors declare that they have no other competing interests.

Supporting Information

[10.22541/au.166005178.87001477/v1](https://doi.org/10.22541/au.166005178.87001477/v1)

References

- A Comparison of Whole-Genome Shotgun-Derived Mouse Chromosome 16 and the Human Genome. (2002). *Science*, 296(5573), 16611671. <https://doi.org/10.1126/science.1069193>
- A Comparative Assessment of the Pig Mouse and Human Genomes. (2011). In *The Minipig in Biomedical Research* (pp. 323342). CRC Press. <https://doi.org/10.1201/b11356-28>
- A comparative encyclopedia of DNA elements in the mouse genome. (2014). *Nature*, 515(7527), 355364. <https://doi.org/10.1038/nature13992>
- Chronic behavioral testing after focal ischemia in the mouse: functional recovery and the effects of gender. (2004). *Experimental Neurology*, 187(1), 94104. <https://doi.org/10.1016/j.expneurol.2004.01.004>
- Intracerebral Hemorrhage in Mice: Model Characterization and Application for Genetically Modified Mice. (2004). *Journal of Cerebral Blood Flow & Metabolism*, 24(5), 487494. <https://doi.org/10.1097/00004647-200405000-00002>
- Mouse model phenotypes provide information about human drug targets. (2013). *Bioinformatics*, 30(5), 719725. <https://doi.org/10.1093/bioinformatics/btt613>
- Cloning characterization, and expression of two angiotensin receptor (AT-1) isoforms from the mouse genome. (1992). *Biochemical and Biophysical Research Communications*, 185(1), 253259. [https://doi.org/10.1016/s0006-291x\(05\)80983-0](https://doi.org/10.1016/s0006-291x(05)80983-0)
- Preclinical research: Make mouse studies work. (2014). *Nature*, 507(7493), 423425. <https://doi.org/10.1038/507423a>
- Novel Object Recognition as a Facile Behavior Test for Evaluating Drug Effects in APP/PS1 Alzheimer's Disease Mouse Model. (2012). *Journal of Alzheimers Disease*, 31(4), 801812. <https://doi.org/10.3233/jad-2012-120151>
- Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications. (2013). *International Journal of Signal Processing Systems*, 256262. <https://doi.org/10.12720/ijsp.1.2.256-262>
- Wireless Inertial Sensors for Monitoring Animal Behavior. (2007, August). *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. <https://doi.org/10.1109/iembs.2007.4352303>
- Investigating Neural Correlates of Behavior in Freely Behaving Rodents Using Inertial Sensors. (2010). *Journal of Neurophysiology*, 104(1), 569575. <https://doi.org/10.1152/jn.00121.2010>
- Accelerometers in collars identify behavioral states in captive African elephants *Loxodonta africana*. (2012). *Endangered Species Research*, 18(3), 255263. <https://doi.org/10.3354/esr00452>
- Movement resting, and attack behaviors of wild pumas are revealed by tri-axial accelerometer measurements. (2015). *Movement Ecology*, 3(1). <https://doi.org/10.1186/s40462-015-0030-0>
- Using accelerometers to remotely and automatically characterize behavior in small animals. (2016). *Journal of Experimental Biology*. <https://doi.org/10.1242/jeb.136135>
- Remote monitoring of vigilance behavior in large herbivores using acceleration data. (2017). *Animal Biotelemetry*, 5(1). <https://doi.org/10.1186/s40317-017-0125-z>
- Categorising sheep activity using a tri-axial accelerometer. (2018). *Computers and Electronics in Agriculture*, 145, 289297. <https://doi.org/10.1016/j.compag.2018.01.007>
- Loor, J. J. (Ed.). (2018). Development and validation of an ensemble classifier for real-time recognition of cow behavior patterns from accelerometer data and location data. *PLOS ONE*, 13(9), e0203546. <https://doi.org/10.1371/journal.pone.0203546>

Dog activity classification with movement sensor placed on the collar. (2018, December). *Proceedings of the Fifth International Conference on Animal-Computer Interaction*. <https://doi.org/10.1145/3295598.3295602>

Assessment of Machine Learning Models to Identify Port Jackson Shark Behaviours Using Tri-Axial Accelerometers. (2020). *Sensors*, 20(24), 7096. <https://doi.org/10.3390/s20247096>

A framework for energy-efficient equine activity recognition with leg accelerometers. (2021). *Computers and Electronics in Agriculture*, 183, 106020. <https://doi.org/10.1016/j.compag.2021.106020>

Classification of broiler behaviours using triaxial accelerometer and machine learning. (2021). *Animal*, 15(7), 100269. <https://doi.org/10.1016/j.animal.2021.100269>

MAMMALIAN EXTINCTIONS IN THE LATE PLEISTOCENE OF NORTHERN EURASIA AND NORTH AMERICA. (1991). *Biological Reviews*, 66(4), 453562. <https://doi.org/10.1111/j.1469-185x.1991.tb01149.x>

Wireless AI-Powered IoT Sensors for Laboratory Mice Behavior Recognition. (2022). *IEEE Internet of Things Journal*, 9(3), 18991912. <https://doi.org/10.1109/jiot.2021.3090583>

Behavior recognition based on machine learning algorithms for a wireless canine machine interface. (2013, May). *2013 IEEE International Conference on Body Sensor Networks*. <https://doi.org/10.1109/bsn.2013.6575505>

Knowledge engineering for unsupervised canine posture detection from IMU data. (2015, November). *Proceedings of the 12th International Conference on Advances in Computer Entertainment Technology*. <https://doi.org/10.1145/2832932.2837015>

Embedded neural network for real-time animal behavior classification. (2018). *Neurocomputing*, 272, 1726. <https://doi.org/10.1016/j.neucom.2017.03.090>

The use of differentially corrected global positioning system to monitor activities of cattle at pasture. (2004). *Applied Animal Behaviour Science*, 85(3-4), 185202. <https://doi.org/10.1016/j.applanim.2003.11.003>

Inference of Animal Activity From GPS Collar Data on Free-Ranging Cattle. (2005). *Rangeland Ecology & Management*, 58(3). https://doi.org/10.2458/azu_rangelands_v58i3_gutman

Simultaneous optimization of a structure and control for mechanical systems using the Nash bargaining game. (1999). *1999 Information Decision and Control. Data and Information Fusion Symposium, Signal Processing and Communications Symposium and Decision and Control Symposium. Proceedings (Cat. No.99EX251)*. <https://doi.org/10.1109/idc.1999.754171>

Activity Recognition from Accelerometer Data Using Symbolic Data Approach. (2018). In *Data Analytics and Learning* (pp. 317329). Springer Singapore. https://doi.org/10.1007/978-981-13-2514-4_27

Window Size Impact in Human Activity Recognition. (2014). *Sensors*, 14(4), 64746499. <https://doi.org/10.3390/s140406474>

Partitioning time series sensor data for activity recognition. (2009, November). *2009 9th International Conference on Information Technology and Applications in Biomedicine*. <https://doi.org/10.1109/itab.2009.5394306>

A Dynamic Sliding Window Approach for Activity Recognition. (2011). In *User Modeling Adaption and Personalization* (pp. 219230). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-22362-4_19

Accurate activity recognition in a home setting. (2008). *Proceedings of the 10th International Conference on Ubiquitous Computing - UbiComp 08*. <https://doi.org/10.1145/1409635.1409637>

Fine-Grained Activity Recognition by Aggregating Abstract Object Usage. (2005). *Ninth IEEE International Symposium on Wearable Computers (ISWC05)*. <https://doi.org/10.1109/iswc.2005.22>

An adaptive time window method for human activity recognition. (2015, May). *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*. <https://doi.org/10.1109/ccece.2015.7129445>

Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer. (2017). *Pervasive and Mobile Computing*, 38, 4159. <https://doi.org/10.1016/j.pmcj.2016.09.009>

A public domain dataset for human activity recognition using smartphones. (2013). *Proceedings of the 21th International European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 437442.

Transition-Aware Human Activity Recognition Using Smartphones. (2016). *Neurocomputing*, 171, 754767. <https://doi.org/10.1016/j.neucom.2015.07.085>

Human Activity Recognition Using Inertial Sensors in a Smartphone: An Overview. (2019). *Sensors*, 19(14), 3213. <https://doi.org/10.3390/s19143213>

Principal component analysis. (2010). *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4), 433459. <https://doi.org/10.1002/wics.101>