Dairy Cow Behavior Recognition Technology Based on Machine Learning Classification

Che-Wei Chou, Chang-Ang Lee, Shu-Wei Guo, Chin-Shiuh Shieh, and Mong-Fong Horng(✉)

Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan
mfhorng@nkust.edu.tw

Abstract. For the livestock industry, the health of cattle is closely related to the quality of their products and the profitability of the enterprise. Timely detection of the health condition of cattle during the farming process is essential to prevent widespread infections or diseases. This study employs inertial sensing devices combined with various machine learning techniques for the classification and comparison of cattle behavior, thus achieving cattle behavior recognition technology. Initially, accelerometer data is collected from different cattle behaviors, and features are extracted from the data. Machine learning algorithms are then applied to classify these features, resulting in the implementation of cattle behavior recognition technology. The computational analysis presents the recognition rates more than 90% for six cattle behaviors and the best case with some behaviors even reaches 95%. With accurate cattle behavior analysis technology in place, livestock operators can gain insights into the activity patterns of cattle, enabling them to identify abnormal health conditions and facilitate early treatment, thus reducing losses.

Keywords: Cattle Behavior Recognition · Machine Learning · Inertial Sensing Technology

1 Introduction

1.1 Background

Currently, there are various diseases affecting cattle, such as anthrax, Bovine Viral Diarrhea (BVD), bovine tuberculosis, Infectious Bovine Keratoconjunctivitis (IBK), Infectious Bovine Rhinotracheitis (IBR), and more. Livestock diseases pose significant challenges to the livestock industry worldwide. Timely detection and resolution of issues are crucial to reduce livestock mortality rates. Early livestock monitoring techniques were time-consuming, had higher error rates, and incurred high equipment costs. Confirming the health status of livestock often required waiting for veterinary observation, delaying necessary treatment and potentially resulting in livestock deaths. This situation significantly impacts the overall economic growth of the livestock industry [1]. Currently in Taiwan, there are less than 40 large animal veterinarians. Estimated data suggests that

J.-S. Pan et al. (Eds.): ICGEC 2023, LNEE 1114, pp. 381–391, 2024.
https://doi.org/10.1007/978-981-99-9412-0_39

on average, one veterinarian is responsible for nearly 5000 head of cattle, which is five times higher than the ratio in Japan [2]. Therefore, if early detection of abnormal cattle behavior can be achieved through cattle behavior recognition technology, there is a potential opportunity to administer relevant treatment before the condition worsens.

1.2 Motivation and Purpose

Benefiting from the advancements in Internet of Things (IoT) devices and cloud-based applications, the adoption of high-tech applications in farm operations worldwide is gradually increasing. The integration of technological innovations has enhanced farmers' production efficiency, ranging from drones capable of automated pesticide spraying and robots for crop harvesting to applications of smart agriculture utilizing artificial intelligence and big data to monitor the conditions of confined livestock. This technological integration holds immense potential for farm management.

This study employs intelligent collars equipped with inertial sensing components on cattle to record the variations in these sensors during different behaviors. It aims to recognize behaviors such as resting, various feeding patterns, rumination, drinking, and salt licking. If certain behavioral data falls below standard values, it can promptly alert farmers for necessary actions. The information farmers desire to obtain while raising cattle is illustrated as follows:

- Injured legs resulting in inability to walk
- Stomach discomfort leading to abnormal eating or rumination behavior
- Insufficient water intake due to short drinking time
- Prolonged absence of salt licking causing nutritional deficiencies

2 Literature Review

2.1 Common Cattle Behaviors

Ruminating is a unique digestive process in cattle, which is crucial for their growth and development. The main differences between resting and ruminating (as shown in Fig. 1) include differences in posture, where cattle slightly raise their heads during rumination; differences in breathing, with a lower respiratory rate during rest and a faster and shallower respiratory rate during rumination; and differences in chewing, as cattle perform chewing motions during rumination to regurgitate and rechew their food.

Cattle licking salt refers to the behavior of cattle satisfying their body's need for sodium and other minerals by licking salt. The deficiency of sodium and other minerals in cattle can lead to nutritional deficiencies and health issues. Insufficient sodium and other minerals can result in reduced appetite, low milk production, delayed growth, and weakened immune function, among other problems.

On farms, cattle feeding behaviors are generally categorized into two types: stall feeding and grazing (as shown in Fig. 2). This differentiation is mainly due to the fact that during stall feeding, cattle consume relatively uniform feed, involving simpler chewing actions primarily using the upper and lower teeth for bite and chewing. On the other hand, during grazing, cattle need to move their heads and mouths to select and chew on grass, involving more complex chewing actions. The advantage of grazing is that cattle can choose and feed according to their preferences, which facilitates better digestion and nutrient absorption for the cattle themselves.

Fig. 1. Illustration of Cattle Resting (RES), Ruminating (RUS), and Licking Salt (SLT) behaviors [3]

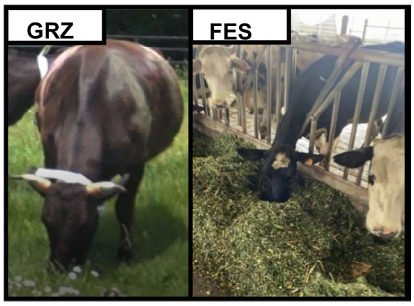

Fig. 2. Illustration of Cattle Grazing Feeding(GRZ) and Bunk Feeding(FES) [3]

2.2 Comparison with Relevant Research

In the study by R. Dutta et al. [4], multiple machine learning classifiers were used to classify different cow behaviors, including eating, rumination, resting, and walking, with random forest showing the best performance. The accelerometers they used were attached near the cow's neck to record their behaviors. J. Kaler et al. [5] used random forest, multi-layer perceptron, SVM, AdaBoost, and KNN to classify lame and non-lame sheep, with random forest yielding the best results.

To recognize the posture of cows using accelerometers, L. Riaboff et al. [6] placed accelerometers on cows in a manner similar to earrings. They applied various machine learning methods including Extreme Gradient Boosting (XGB), random forest, and SVM, with XGB demonstrating the best performance.

3 Cow Behavior Recognition Technology

3.1 Introduction to the Dataset

This study utilized the Japanese Black Cattle behavior dataset collected by Shinshu University in Nagano, Japan [7]. The dataset consists of tri-axial accelerometer data capturing the behavior of six different Japanese Black Cattle individuals. The sampling frequency of the data is 25 Hz. The cattle were allowed to move freely in two open spaces, namely a grassy field and an enclosure within a farm. Simultaneously, video recordings were taken to document the behaviors. The labeling of behaviors was carried out by experienced caretakers based on the video footage.

This study focused on six common behaviors exhibited by the cattle:

- Resting while standing, labeled as RES.
- Ruminating, labeled as RUS.

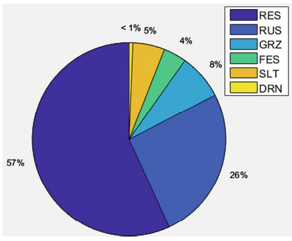

Fig. 3. Proportions of Different Behaviors in the Dataset

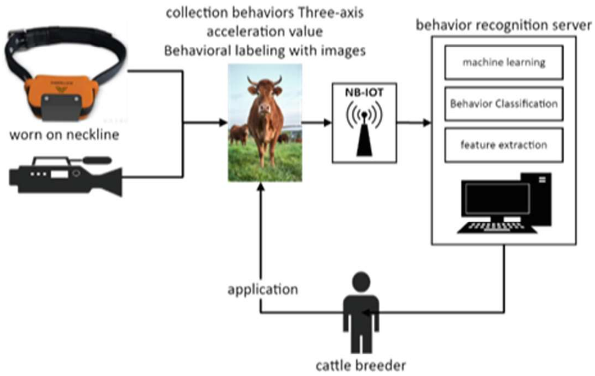

Fig. 4. Application Scenario Diagram

- Feeding at a feeding station, labeled as FES.
- Grazing on the grassy field, labeled as GRZ.
- Drinking water, labeled as DRN.
- Licking salt, labeled as SLT.

The proportions of these six behaviors in the dataset are depicted in Fig. 3. Figure 4 illustrates the application scenario of this study.

3.2 Behavior Recognition Server System Architecture

This study utilized an Intel i5–9500 CPU running on the Windows 10 operating system for cattle behavior recognition technology. The Python and MATLAB functions used during development are listed in Fig. 5.

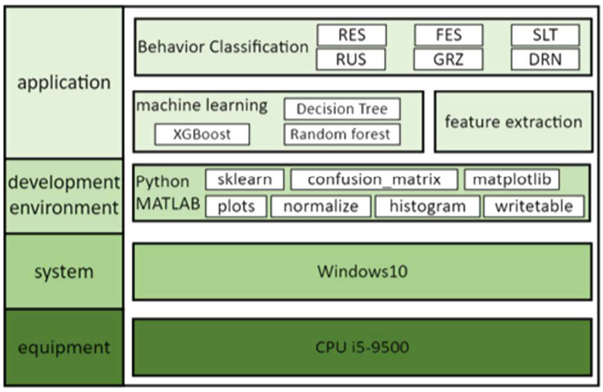

Fig. 5. System Architecture Diagram of Behavior Recognition Server

3.3 Calculation Process

The calculation process of this study is illustrated in Fig. 6. Firstly, the collected three-axis acceleration values undergo data preprocessing such as normalization and moving average, followed by feature extraction. Next, the dynamic vector sum is calculated to quantify the intensity of the object's movement, enhancing recognition accuracy. Subsequently, the accumulated distance values between the dynamic vector sum and the predicted value are computed over a 4-s time interval to distinguish whether the cattle are in a moving state. Finally, the data that is not in a moving state is subjected to classification using the decision tree and random forest methods for the classification of the remaining four behaviors.

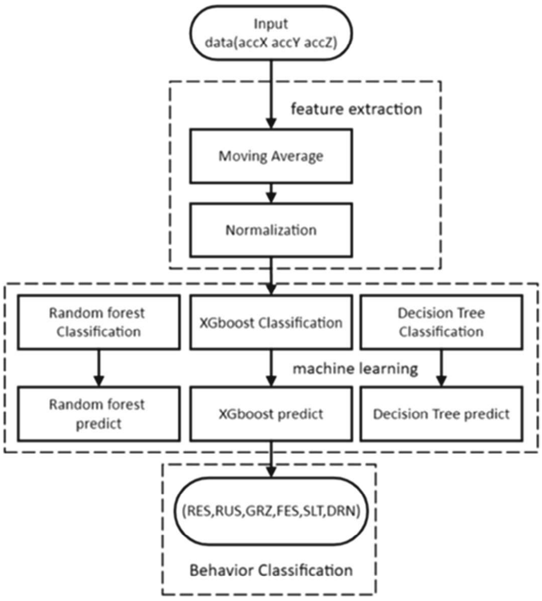

Fig. 6. Flowchart of the Calculation Process

3.4 Data Preprocessing

In order to mitigate the impact of noise caused by the environment when collecting cattle behavior data, such as distinguishing between resting and rumination, where the difference lies only in slight head raising and lower jaw movement, noise can easily lead to inaccurate recognition. Therefore, prior to analyzing the data, data preprocessing is necessary to enhance subsequent behavior recognition. This study employed moving average filtering and normalization as preprocessing techniques on the raw data. Moving Average (MA) helps to eliminate noise or fluctuations in the signal, aiding in the removal of momentary interferences or noise. This is represented by Eq. (1).

$$y[x_i] = \frac{1}{N}\sum_{k=0}^{N-1} x_i[n-k], i \in \{x, y, z\} \tag{1}$$

where $y[x_i]$ represents the filtered result, $x_i[n]$ denotes the values of the original input sequence, and N indicates the size of the moving average window.

After applying moving average processing, this study normalized the data to the range of [0, 1] to reduce the differences between different cattle, aiming to enhance the performance and stability of machine learning algorithms. This is shown in Eq. (2).

$$Nr[n] = \frac{(y[n] - \min(n))}{(\max(\mathrm{n}) - \min(\mathrm{n}))} \tag{2}$$

where $y[n]$ represents the data after moving average, and $\min(n)$ and $\max(\mathrm{n})$ are the minimum and maximum values, respectively.

3.5 Machine Learning Methods

Machine learning is well-suited for training effective classifiers from known behavioral samples to distinguish patterns in cattle behavior. These patterns can then be used to classify the behaviors of unknown cattle. The machine learning classification methods used in this study are introduced as follows.

Decision Tree. In this study, the sklearn.tree package was used for implementing the decision tree algorithm. The imported dataset was split into an 80% training set and a 20% test set for classification. Through experimentation, it was determined that setting the maximum depth of the tree (max depth) to 12 achieved the highest accuracy.

The minimum sample split (min samples split) is one of the conditions controlling the stopping criterion for tree splitting. When the sample count at a node is less than the specified value, the decision tree stops splitting at that node to prevent overfitting. This parameter is typically set between 2% and 10% of the total dataset. Given the unequal proportions of behavior sample labels in this study, the behavior with the fewest samples was chosen and set to 2% for this parameter value.

Another parameter is the maximum number of features (max features), used to ensure that each node considers a certain number of features to avoid having too few features, which could result in an insufficient sample count.

Random Forest. There are two common voting mechanisms in Random Forest: majority voting and weighted voting. Majority voting selects the class that appears most frequently as the prediction, while weighted voting assigns weights based on the accuracy of each decision tree and selects the final prediction after summing the weighted results.

To implement the multi-decision tree voting of Random Forest, this study fine-tuned the parameters of the decision trees. Firstly, the data underwent preprocessing, and then the dataset was split into a 20% testing set and an 80% training set, with different behaviors labeled accordingly. The model was constructed using the RandomForestClassifier package from the sklearn library. When adjusting the Random Forest model, the following three parameters were involved: **Number of Trees** (n estimators): A higher number of trees leads to higher accuracy but increases computation time. This study set the number of decision trees to 150 for voting. **Tree Depth** (max depth): In comparison with individual decision trees, the maximum depth of the trees in Random Forest was also set to 12. **Minimum Samples for Split** (min samples split): Since Random Forest involves multi-decision tree voting, it tolerates a relatively higher minimum sample requirement. Setting it too high would reduce the diversity among different decision trees and undermine the advantages of Random Forest. This study set the minimum samples for split to 40.

Additionally, concerning the maximum number of features (max features), setting it too low would result in each decision tree considering only a few features, affecting the model's performance. Setting it too high might lead to overfitting. Considering that some behavior labels had a smaller proportion in the dataset, the calculation of the maximum number of features was based on the average of these behavior labels, followed by taking the square root. Finally, this study used a confusion matrix to present the prediction results, comparing the performance of Random Forest and decision trees in classification. Through the multi-decision tree voting mechanism, Random Forest embodies the concept of collective intelligence.

Extreme Gradient Boosting (XGBoost). In this study, after employing the machine learning methods of decision trees and random forests for classification, XGBoost was introduced for comparison. The distinction between XGBoost and random forests can be explained as the difference between Bagging and Boosting. Bagging generates individual trees through random sampling, without interconnections, while random forests are an implementation of Bagging. Boosting, on the other hand, generates trees successively, with each subsequent tree built upon the previous one. XGBoost is a realization of the Boosting method. Each tree in a Boosting model addresses the shortcomings of the preceding tree, making it generally more precise than Bagging.

In the model building process, preprocessed acceleration data was used. The dataset was split into an 80% training set and a 20% testing set, with behaviors labeled accordingly. The model was constructed using the ensemble package from the sklearn library. When selecting the XGBoost model, the softmax function from Eq. (3) was utilized. This function, also known as the normalization function, maps vectors to the [0, 1] interval, representing the probability distribution of each classification. The sum of the vector elements equals 1.

$$Softmax\left(z_j\right) = \frac{e^{z_j}}{\sum_{R=1}^{K} e^{Z_R}}, for\ i = 1, \ldots, K \tag{3}$$

Next, important model parameters need to be determined, including the maximum tree depth, total number of iterations, learning rate, and maximum number of features. After hyperparameter tuning, a model with 14 layers and 400 iterations was chosen. The learning rate reflects the similarity of features between generating subsequent sets of trees and the previous set of trees. A higher learning rate indicates a higher degree of similarity. In this study, a learning rate of 0.1 was adopted, meaning that 10% of the features from the previous tree are retained in each iteration. Lastly, since the dataset size remains unchanged, the choice of maximum number of features aligns with the previous decision for the decision tree and random forest models.

3.6 Model Optimization and Performance Tuning

GridSearchCV provides a straightforward and effective method for selecting the best combination of hyperparameters. It operates by conducting an exhaustive search within the specified ranges of hyperparameters, performing cross-validation for each combination, and calculating corresponding model performance metrics. Ultimately, it selects the optimal set of hyperparameters based on the results of cross-validation.

Table 1 and Table 2 respectively present the tuned hyperparameters in the Random Forest and XGBoost models of this study. Among these, "max_depth" refers to the depth of decision trees, where a higher value indicates a more complex model. "n_estimators" represents the number of decision trees within the model. "learning_rate" denotes the learning rate, with higher values implying a greater proportion of the previous tree's contribution in the subsequent tree. "subsample" signifies the proportion of sub-samples used during training; for instance. Lastly, "colsample_bytree" refers to the proportion of sub-samples used when building each decision tree.

Table 1. Random Forest Hyperparameter Tuning Metrics.

Parameter	Range
max_depth	7、8、9、10、11、12、13、14
n_estimators	100、150、200、250、300

Table 2. XGBoost Hyperparameter Tuning Metrics.

Parameter	Range
max_depth	7、8、9、10、11、12、13、14
learning_rate	0.01、0.02、0.05、0.1、0.2、0.5
subsample	0.5、0.6、0.7、0.8、0.9
colsample_bytree	0.8、0.9、1.0
n_estimators	100、150、200、250、300

4 Performance Evaluation and Analysis

4.1 Restated Validation Objectives

In this study, three different machine learning methods were employed for cattle behavior classification: Decision Tree, Random Forest, and XGBoost.

4.2 Machine Learning Prediction Results

Figures 7 and 8 present the confusion matrix plots obtained from using the Decision Tree and Random Forest machine learning methods for classification. From Fig. 7, it can be observed that while the Decision Tree exhibits a certain level of recognition capability for each behavior, its performance is relatively poorer in identifying Resting (RES) and Rumination (RUS), as well as Stationary Feeding (FES) and Grazing Feeding (GRZ). This is due to the higher similarity between these behaviors.

In Fig. 8, the Random Forest benefits from its internal voting mechanism, effectively mitigating the risk of overfitting compared to a single Decision Tree, as it leverages random sampling and feature selection. However, the voting mechanism's influence is also notable in classifying similar behaviors such as Resting (RES) and Rumination (RUS), as well as Stationary Feeding (FES) and Grazing Feeding (GRZ), leading to improved results in these cases.

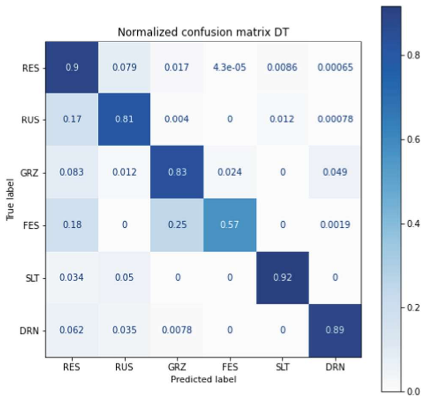

Fig. 7. Confusion Matrix of Decision Tree Classification Results.

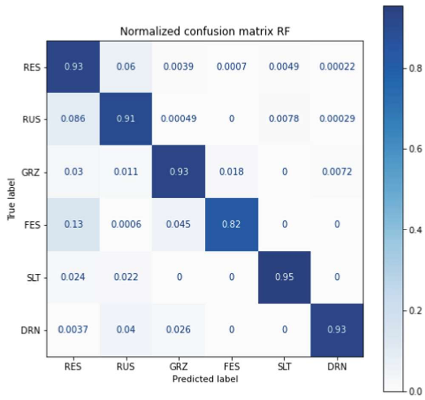

Fig. 8. Confusion Matrix of Random Forest Classification Results.

Within the XGBoost model (as shown in Fig. 9), owing to the boosting nature, each set of decision trees generated retains certain features from the preceding set, enhancing the classification performance. Consequently, the XGBoost model improves classification accuracy, particularly for similar behaviors like Stationary Feeding (FES) and Grazing Feeding (GRZ), where the model's ability to enhance classification.

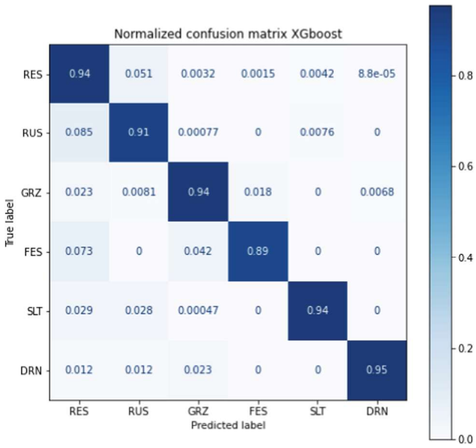

Fig. 9. Confusion Matrix of XGBoost Classification Results.

5 Conclusion

The main objective of this study is to explore how accelerometer and machine learning techniques can be employed to identify cattle behavior. The results of the study demonstrate that the combination of accelerometers and machine learning techniques can accurately monitor various behaviors of cattle. This research introduces a novel approach to achieve cattle behavior recognition, aiding farmers in gaining a more precise understanding of cattle behavior. This methodology contributes to the timely detection of potential health issues and the improvement of cattle husbandry practices.

Acknowledgement. This research was supported by the National Science and Technology Council with grant numbers: MOST 109–2221-E-992 -073 -MY3, NSTC 112–2622-8–992-009 -TD1 and NSTC 112–2221-E-992 -057 -MY3.

References

1. Rony, M., Riad, D.B., Hasan, Z.: Cattle External Disease Classification Using Deep Learning Techniques. In: 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, pp. 1–7
2. Bettermilk. The talent dilemma in Taiwan's large animal veterinary field: Reasons for the severe imbalance in human-cattle ratio (13 September 2022). https://www.bettermilk.com.tw/blogs/veterinary/108510
3. Williams, L.R., Bishop-Hurley, G.J., Anderson, A.E., Swain, D.L.: Application of accelerometers to record drinking behaviour of beef cattle. Animal Product. Sci. **59**(1), 122–132 (2017)
4. Dutta, R., et al.: Dynamic cattle behavioural classification using supervised ensemble classifiers. Comput. Electr. Agricult. **111**, 18–28 (2015)

5. Kaler, J., Mitsch, J., Vázquez-Diosdado, J.A., Bollard, N., Dottorini, T., Ellis, K. A.: Automated detection of lameness in sheep using machine learning approaches: novel insights into behavioural differences among lame and non-lame sheep. Royal Soc. Open Sci. **7**(1), 190824 (2020)
6. Riaboff, L., Poggi, S., Madouasse, A., Couvreur, S.: Development of a methodological framework for a robust prediction of the main behaviours of dairy cows using a combination of machine learning algorithms on accelerometer data, vol. 169 (2020)
7. Ito, H., et al.: Japanese Black Beef Cow Behavior Classification Dataset (v2.0.0) (2022). Zenodo. https://doi.org/10.5281/zenodo.5849025