Detecting abnormal behavior of cows using accelerometer data

***Abstract*—the paper shows the type of abnormal behavior for cows and two experiments evaluating the performance of recurrent neural networks (RNNs) and a convolutional neural network combined with RNNs on the dataset using different oversampling techniques. The first experiment was comparing 6 different models and it is found that CNN-BI-LSTM has the highest accuracy among the models studied with 90.82%. In the second experiment, the CNN-BI-LSTM model was used with different window sizes trying to detect the abnormal behavior achieving 87.96%**

Overall, the experiments provide useful insights into the effectiveness of RNNs and highlight the potential of CNN-BI-LSTM for scoring the highest average of 90.82% which shows its ability for sequence classification tasks. However, the accuracy of the models may depend on specific application and dataset characteristics, such as the length of input sequences or types of features included.

I. Introduction

Abnormal behavior in cows can be a sign of a serious disease or health problem [1]. These behaviors can have a negative effect on the well-being and productivity of the cow, as well as a risk of other cows being infected. It is essential for farm- ers and livestock managers to be aware of the signs of abnormal behavior and take quick action to avoid the disease from spreading and treat the infected cow. In some cases, cows may also exhibit aggression due to hormonal imbalances or neurological disorders. Other factors that can contribute to abnormal behavior in cows in- include environmental such as poor living conditions, overcrowding, or inadequate nutrition.

The first disease is Bovine spongiform encephalopathy (BSE), also known as mad cow disease [2], which can cause cows to exhibit abnormal behavior such as aggression, and nervousness, which may lead to escaping and attacking behaviors.

The second disease is a viral disease called Rabies [3]. That disease affects the nervous system of cows and can cause them to act aggressively and violently, attacking other animals or humans.

The Pain and discomfort of the cow also may lead to aggressive behavior [4]. Some conditions such as mastitis, lameness, or injury are more likely to show aggressive behavior attacking humans and other cows.

II. Related work

Lameness in dairy cattle is a serious health and welfare issue that requires prompt diagnosis and treatment. According to Haladjian et al., [10], visual observation by farm staff has been the most traditional way of tracking the walking of cows to identify lameness. However, this method has limitations as it may not be able to detect early stages of lameness.

To address this issue, Haladjian et al. propose a method for detecting any alterations in the regular walking behavior of cows through the use of a wearable motion sensor. The process involves creating a reference model of a cow's typical walking pattern within the first few minutes of attaching the sensor, assuming that every cow that has the sensor is healthy at first and walks normally during that time. Any deviations from this baseline pattern can then be identified and further examined by veterinarians.

The team designed a sensor device containing an ARM Cortex-M0 microcontroller, a 6-axis Inertial Measurement Unit (IMU), and a Bluetooth Low Energy (BLE) module for sending the data. The ARM Cortex-M0 microcontroller operates at 16 MHz and has a low power consumption rate. The MPU-6050 was chosen to measure the angle because when the cow starts to lame, it often steps with a different angle to avoid pain. BLE was chosen to make it cost-efficient because it doesn't consume much power.

The designed sensor measures the acceleration and rotation of x, y, and z at 100hz and then divides it into chunks of a 30-second window. The team used step segmentation to know the beginning of the start and the end of each step depending on the linear acceleration that needs only the y-axis. Every step was characterized by two high peaks and one low peak. Before a cow takes a step, there is a period where there is very little variation in the acceleration measured by the accelerometer.

Colin Tobin et al. proposed a system for detecting Bovine Ephemeral Fever (BEF) [11]. BEF is a viral disease that affects cattle, causing fever, lameness, and stiffness, which can lead to reduced milk production and economic losses. Currently, BEF is diagnosed based on clinical signs and confirmed with laboratory testing. However, clinical diagnosis can be challenging, and laboratory testing is expensive and time-consuming.

The system was designed to monitor changes in the accelerometer data, which are indicative of BEF symptoms, such as decreased activity levels. The researchers used data from X, Y, and Z axes to calculate their mean values for every minute. One-minute epochs have 1440 data points, which is equivalent to T at the 25 Hz sampling frequency. The study found that the accelerometer axes (Ax, Ay, Az) did not show any significant effects of model parameters when evaluated separately. However, the statistical model detected differences

Algorithm	Number of behavior	Dataset	Acclometer place	Accuracy	refrence
Decision tree	4	9 dairy cows over a period of 14 days	Head	85%	[5]
C4.5	unknown	15 cows,5 healthy cows, and 10 infected	Neck	92%	[6]
LDA, QDA	5	15 sheep were randomly chosen	Neck	87.1% LDA 89.7%QDA	[7]
Decision-tree	3	6 dairy cows monitored 36 h	Neck	sensitivity 96.45% precision 87.50%	[8]
Multivariate Finite Mixture Model and decision tree	3 classes each have 2 subclass	12 lactating dairy cows for 24 hours.	Back	99%	[9]

TABLE I
GAP ANALYSIS

in both MI and SMA, and the results were the same for both. Therefore, only movement intensity (MI) data were presented and discussed in the study.

The healthy control heifers showed comparable MI values on the day of diagnosis and two days before the diagnosis. Conversely, the two heifers that were infected with BEF displayed lower MI values, indicating lower activity levels on the day of diagnosis compared to two days before when they were still healthy. Moreover, the two heifers diagnosed with BEF exhibited lower MI values on the morning before diagnosis in comparison to the healthy control animals.

The study was conducted on a dataset of 8 cows, with 2 of the cows infected with BEF and the other half serving as the control group. The author has proof of the concept of the use of accelerometers to detect abnormal behavior, but more research should be done.

The results of the study show the potential of detecting BEF symptoms in cattle using accelerometer-based measurements. The system has the potential to improve early detection and reduce the spread of the disease, ultimately reducing economic losses.

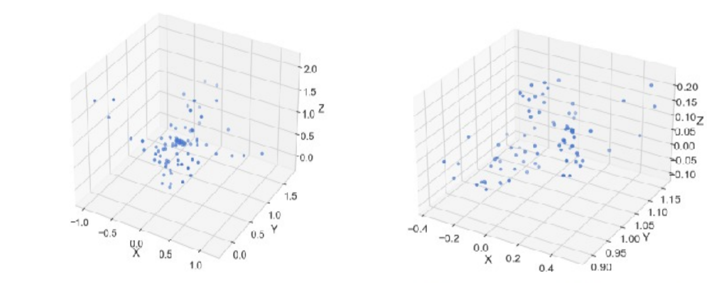
The system proposed in this paper consists of six stages: 4 pre-processing, 2 processing, and output, Data segmentation, filtering, normalization, data augmentation, behavior classification, threshold and output Figure 3. In this section, a comprehensive explanation of each stage is provided.

A. Dataset

This paper has a dataset consisting of tri-axial accelerometer sensor data with labels for thirteen different cow behaviors [12]. The data was collected on June 12th, 2020 at a cow farm in Nagano, Japan, using a Kionix KX122-1037 accelerometer attached to the necks of six different Japanese Black Beef Cows (cow1.csv-cow6.csv). The cows were allowed to roam freely in both a grass field and farm pens and were filmed with Sony FDR-X3000 4K video cameras during data collection.

To ensure the accuracy of the labeling process, human observers, including behavior experts and non-experts, labeled the data from the video footage while matching the timestamps of the video and accelerometer data. The labeling and data-gathering process took a total of 69 person-hours.

The dataset includes 567 minutes of unlabeled data, which were parsed into 197 minutes of high-quality labeled data comprising thirteen behaviors. The labeling was performed by three annotators through majority voting, resulting in accurate behavior classification for each data sample at a sampling frequency of 25Hz.



(a) 3D accelrometer data for Attacking behaviour (b) 3D accelrometer data for Escaping behaviour

Fig. 1. a attacking example

Fig. 2. b escaping example

Fig.2 and Fig.3 show an example of 50 records from the dataset of how the attacking and escaping look like

III. METHODOLOGY

A. Pre-processing

The first step in the proposed system is the segmentation process. Time series segmentation was used which segments are created with a fixed length of 25-time steps and a step size of 25-time steps, where each time step corresponds to a measurement of the accelerometer along the x, y, and z axes without overlapping as it was used before [13] a time-series segmentation approach for analyzing the behavior of chickens in a commercial farm setting. The method involves using a combination of filtering, segmentation, and clustering techniques to identify and categorize different behavioral patterns based on sensor data collected from the chickens. The study demonstrated that the proposed approach was effective in accurately detecting and classifying various behaviors, such as eating, drinking, and walking, and could potentially be used as a tool for monitoring behavior.

The filter that was savgol filter commonly used with z-score normalization [14].The filter window size of 51 to specifies the number of data points used in each local polynomial regression, and the poly order of 3 to specifies the degree of the polynomial used in the regression. this filter was used on every axis separately to smooths out noise in the data.

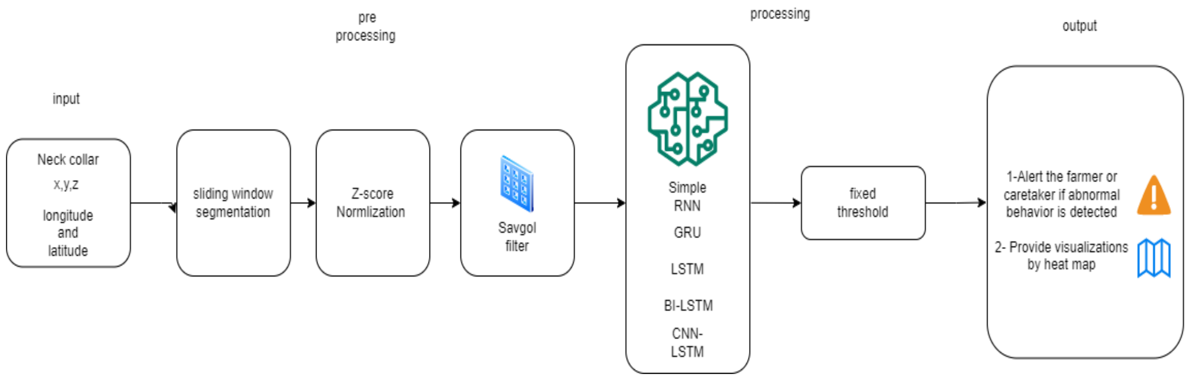


Fig. 3. System Overview

The normalization technique that was used is z-score normalization to Pre-Process accelerometer data after filtring [15]. The equation for calculating the z-score was

$$z = \frac{x - \mu}{\sigma}$$

where:
z is the z-score or standard score of the data point
x is the value of the data point
μ is the mean of the dataset
σ is the standard deviation of the dataset

B. Processing

The first model and Fig.4 represent the simple RNN used [16] the model starts with a Sequential model object, adds a Simple RNN layer with 128 neurons, and applies a dropout of 0.2 to prevent overfitting. Next, it had a dense layer with a softmax activation function for multi-class classification. The model was compiled with the sparse-categorical cross-entropy loss function, the Adam optimizer, and the accuracy metric.

The second model the GRU was used [17]. The model starts sequential model object is called to sequential model object then the GRU layer with 128 units is added as the first layer of the model. A Dropout layer is added after the GRU layer to prevent overfitting. The Dropout layer randomly sets a fraction of the input units to 0.2 at each update during training time, which helps prevent overfitting. A final Dense layer with the number of output units and softmax activation function is added to the model for multi-class classification purposes.

The third model the LSTM was used [18]creates a Sequential model using the Keras. The model includes an LSTM layer with 100 units followed by a dropout layer with a rate of 0.3 to avoid overfitting, followed by one fully connected dense layer with the number of outputs. The activation functions used are soft-max for the output layer.

The fourth model is defined using the Sequential API from Keras [19]. The model starts with a 1D convolutional layer with 64 filters and a kernel size of 3, followed by a max pooling layer with a pool size of 2. This layer is used to extract features from the input data. The output of this layer is then passed to an LSTM layer with 128 units and a dropout rate of

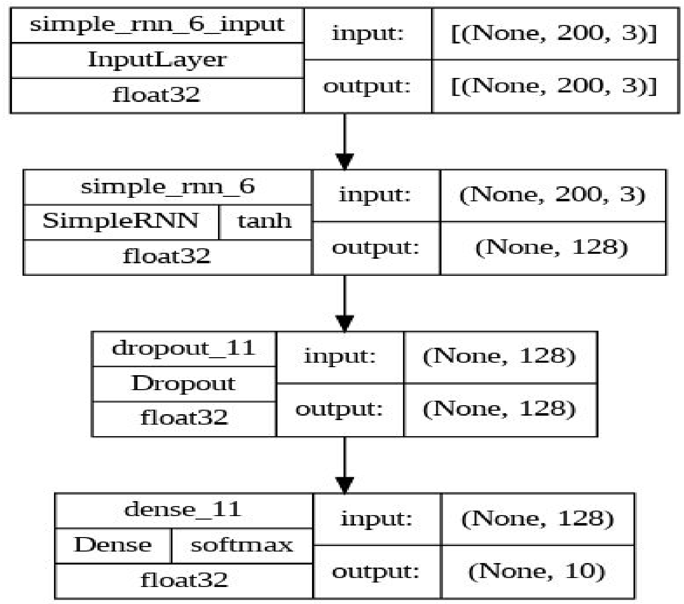


Fig. 4. Simple RNN model

0.2. The LSTM layer is used to capture temporal dependencies in the input data. The output of the LSTM layer is then passed to The final dense layer that has a number of units equal to the number of classes in the dataset and a softmax activation function, which outputs class probabilities. The model is then compiled using the categorical cross-entropy loss function, the Adam optimizer, and the accuracy metric. It is then trained using the fit method of the model object. Finally, the model is evaluated on the test set using the evaluate method, and the accuracy is returned as the output of the function.

the fifth model is a Sequential model that includes a Bidirectional LSTM [20] layer with 128 units, a dropout layer with a dropout rate of 0.2, a dense output layer with a softmax activation function. The model is compiled using the categorical cross-entropy loss function, the Adam optimizer, and accuracy as the evaluation metric. Bidirectional LSTMs are useful for capturing long-term dependencies in time series data by processing the input data in both forward and backward directions. Dropout layers are used to prevent overfitting by randomly dropping out units during training. The ReLU activation function is commonly used in neural networks due to its ability to introduce nonlinearity and avoid the vanishing gradient problem. The softmax activation function is used in the output layer to produce a probability distribution over the possible class labels. The categorical cross-entropy loss function is a commonly used loss function for multiclass classification problems. The Adam optimizer is a popular optimization algorithm that is well-suited for deep learning models.

The last model is a Sequential model that consists of several layers. It starts with a Conv1D layer with 64 filters, a kernel size of 3, and the ReLU activation function. This layer is followed by a MaxPooling1D layer with a pool size of 2, which helps reduce the spatial dimensions of the input. Next, a Bidirectional LSTM layer with 128 units is added. The Bidirectional LSTM is a type of recurrent neural network (RNN) layer that processes the input data in both forward and backward directions. This allows the model to capture long-term dependencies in the time series data. To prevent overfitting, a Dropout layer with a dropout rate of 0.2 is applied after the Bidirectional LSTM layer. Dropout randomly drops out units during training, forcing the model to learn more robust and generalizable features. Finally, a Dense layer with a softmax activation function is added as the output layer. The softmax activation function produces a probability distribution over the possible class labels, making it suitable for multiclass classification problems.

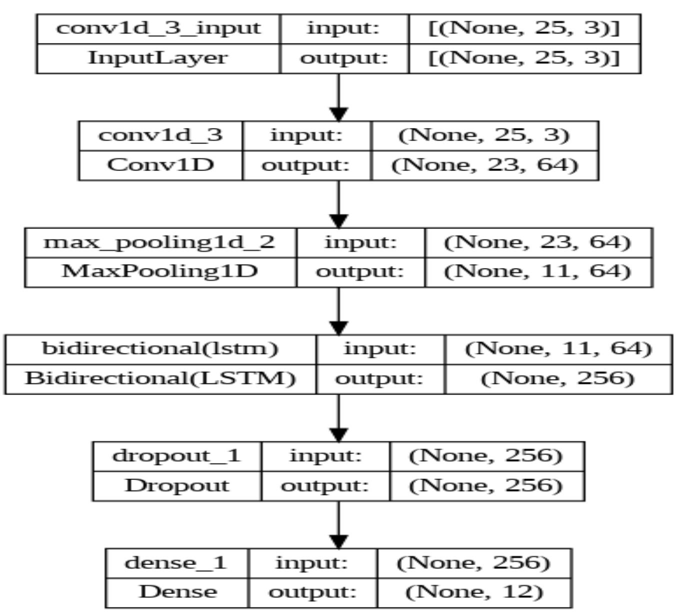


Fig. 5. BI-lstm model

the last thing in processing is threshold [21]. A threshold is a value or a range of values that is used to classify data points or measurements as being part of a certain group or category. In time series data, thresholds are often used to identify anomalous events or patterns that deviate from the expected behavior of the data. By setting a threshold, one can define what constitutes an abnormal behavior or event and trigger an alert or take corrective action when such an event is detected. Thresholds can be static or dynamic, and their selection depends on the nature of the data and the specific application. Static thresholds are fixed and do not change over time.

C. Output

The output consists of one output. The output is the behavior detected by the model from the 13 behavior

IV. Experiments and Results

A. Experiment 1

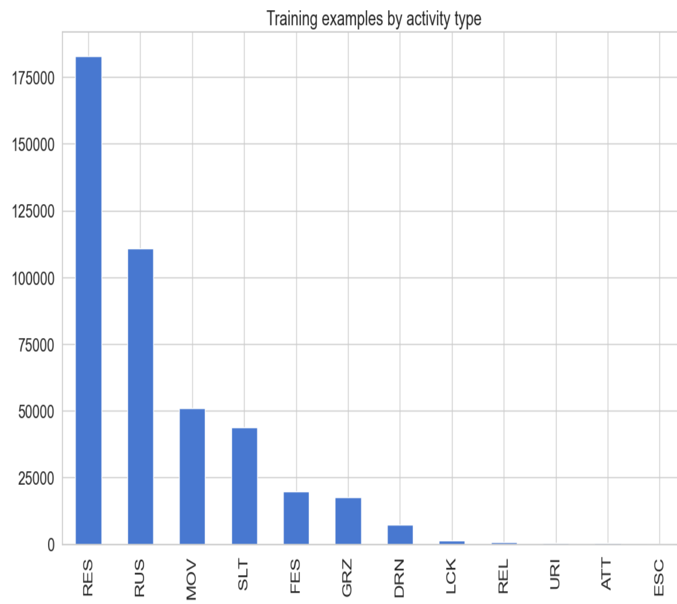


Fig. 6. The imbalance in the Data set

In the two experiments, we dropped being mounted(BMN) for its poor sample data. The first experiment, The Synthetic Minority Over-sampling Technique (SMOTE) on the training dataset to address the class imbalance shown in Fig.6 SMOTE is a popular method for generating synthetic samples of the minority class by interpolating between existing samples. After applying SMOTE, also the Adaptive Synthetic (ADASYN) was used algorithm on the training dataset to further address the class imbalance. ADASYN focuses on generating synthetic samples for the minority class based on the density distribution of different classes.The two techniques used k neighbors =2 due to the low number of samples for minority classes.we evaluated the model using the following metrics:

Accuracy =$TP + TN/TP + TN + FP + FN$

Loss=Sparse Categorical Cross-Entropy

precision =$TP/TP + FP$

Recall= $TP + FN/TP$

Model	Accuracy	Loss	Precision Score	Recall Score	F1 Score
RNN	80.54%	0.5627	0.8292	0.8054	0.8110
GRU	83.59%	0.5409	0.8413	0.8353	0.8366
LSTM	85.43%	0.5426	0.8627	0.8543	0.8565
BI-LSTM	85.61%	0.5472	0.8654	0.8561	0.8589
CNN-LSTM	88.39%	0.4455	0.8920	0.8839	0.8852
CNN-BI-LSTM	90.82%	0.4053	0.9087	0.9082	0.9081

TABLE II
ACCURACY OF THE MODELS USING SMOTE

Model	Accuracy	Loss	Precision Score	Recall Score	F1 Score
RNN	78.23%	0.6224	0.8230	0.7899	0.7988
GRU	83.70%	0.5353	0.8539	0.8370	0.8418
LSTM	80.72%	0.6701	0.8278	0.8072	0.8105
BI-LSTM	86.22%	0.5052	0.8714	0.8622	0.8644
CNN-LSTM	88.86%	0.4040	0.8954	0.8886	0.8906
CNN-BI-LSTM	89.95%	0.4556	0.9060	0.8995	0.9009

TABLE III
ACCURACY OF THE MODELS USING ADYSN

F1-Score= $2 * TP/2 * TP + FP + FN$

The two tables represent the accuracy, loss, precision score, recall score, and F1 score of different models evaluated in your research paper. The first table (Table II) shows the accuracy of the models using SMOTE, a technique for handling imbalanced datasets. The second table (Table III) shows the accuracy of the models using ADYSN, another technique for dealing with imbalanced datasets.

From the results in Table II, it can be observed that the models achieved varying levels of accuracy. The GRU model had the highest accuracy of 83.70%, followed by the CNN-BI-LSTM model with 89.95% accuracy. These results indicate that applying SMOTE helped improve the model performance compared to the results without using any data augmentation technique.

Similarly, in Table III, the models trained using ADYSN showed improved accuracy compared to the models without any data augmentation. The CNN-BI-LSTM model achieved the highest accuracy of 89.95%, followed by the CNN-LSTM model with 88.86% accuracy.in this experiment attacking and escaping were not detected even after applying the over-sampling techniques

B. Experiment 2

In the second experiment, we focused on utilizing the CNN-BI-LSTM model with different window sizes in combination with the SMOTE and ADYSN techniques. The objective was to improve the detection of minority abnormal classes, specifically attacking and escaping behaviors.

First, we applied the SMOTE technique to the training dataset, which involved oversampling the minority classes to achieve a more balanced representation. This approach helps address the issue of class imbalance and allows the model to learn from a more diverse set of samples. The CNN-BI-LSTM model was then trained on the augmented dataset.

Next, we employed the ADYSN technique on the training dataset. ADYSN is a specialized data augmentation technique designed to enhance the representation of rare events or abnormal behaviors. By leveraging ADYSN, we aimed to further enhance the detection capabilities of the CNN-BI-LSTM model for the attacking and escaping behaviors.

For both the SMOTE and ADYSN experiments, we evaluated the performance of the CNN-BI-LSTM model using various window sizes. The window size refers to the number of consecutive accelerometer readings considered as a single input sequence. By exploring different window sizes, we aimed to find the optimal configuration that maximizes the detection accuracy and minimizes false positives/negatives for the minority abnormal classes. the two algorithms used k neighbors =5

In this experiment, we aimed to detect the minority classes of ”attacking” and ”escaping” using a window size of 5. We achieved reasonable accuracy in detecting these classes; however, we noticed that the model's performance had an impact on the other classes as shown in Fig.7. When we

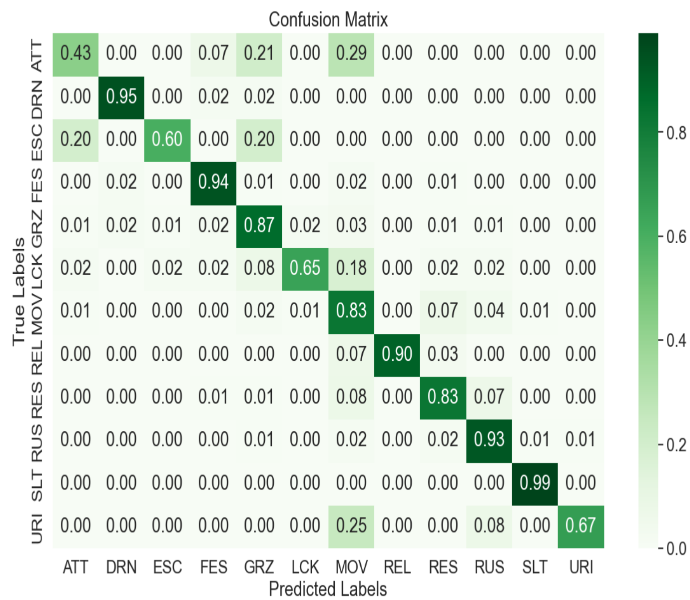


Fig. 7. The confusion matrix using SMOTE

increased the window size to 10, the model was not able to effectively detect the ”attacking” and ”escaping” classes. We hypothesized that the larger window size may have led to the dilution of the distinctive patterns and features related to these specific classes.

To further investigate the impact of window size, we conducted a range analysis. By varying the window size within a specific range, we observed changes in the model's ability to accurately detect the ”attacking” and ”escaping” classes, as well as any potential trade-offs with the performance of other classes.

C. Experiment3

In the third experiment, a mobile application was developed using Android Studio Fig.8 to measure the acceleration values

Window Size 5	Accuracy	Loss	Precision Score	Recall Score	F1 Score
SMOTE	87.96%	0.4030	0.8950	0.8796	[illegible]
ADASYN	87.36%	0.4413	0.8932	0.8736	[illegible]

TABLE IV
ACCURACY OF THE CNN-BI-LSTM USING WINDOW SIZE 5

Window Size 10	Accuracy	Loss	Precision Score	Recall Score
SMOTE	88.33%	0.4306	0.8938	0.8833
ADASYN	87.11%	0.4313	0.8807	0.8711

TABLE V
ACCURACY, LOSS, PRECISION SCORE, RECALL SCORE, AND F1 SCORE FOR WINDOW SIZE 10 WITH SMOTE AND ADASYN

(Acc X, Acc Y, and Acc Z) of a cow. The application was configured with SENSOR DELAY NORMAL, which sampled the acceleration data at a rate of 5Hz. To ensure accurate measurements, the mobile device was attached to the cow using a strap and double-face tape.

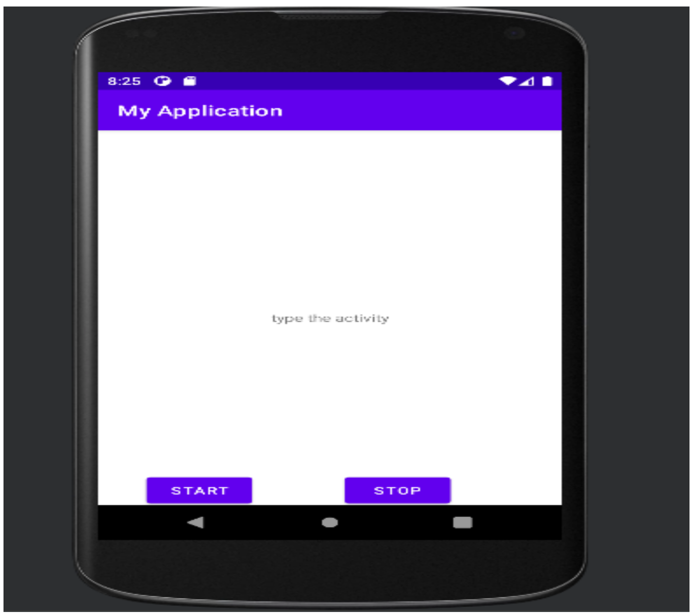


Fig. 8. The application interface

The experiment involved capturing samples of the cow's movement as shown in Fig.9 and rest periods as shown in Fig.10. The data collected from the accelerometer sensor in the phone was processed and analyzed. The experiment utilized video recording alongside the running of the mobile application to synchronize the sensor data with the corresponding cow's activities. The recorded videos were later trimmed to extract the necessary sensor data.

However, there were some limitations in this experiment. The first limitation was related to the attachment of the mobile device to the cow. Due to difficulties in securely attaching the device, the attachment method had to be changed, which may have affected the accuracy and consistency of the collected data. Additionally, the positioning of the mobile device on the cow might not have been optimal, leading to potential variations in the recorded acceleration values.

Fig. 9. The rest behavior

Fig. 10. The movement behavior

As a result of these limitations, the overall accuracy of the experiment was reported to be 47.3%. It is important to consider these limitations when interpreting the results and understanding the reliability of the collected data. Future improvements in the attachment method and positioning of the mobile device could potentially enhance the accuracy and reliability of similar experiments.

V. CONCLUSION AND FUTURE WORK

Based on the experiments discussed, the system proposes the bi-LSTM model is effective at classifying different types of animal behavior based on input sequences, with accuracy scores ranging from 89.5% to 97.6%. In addition, the comparison of different types of neural networks for detecting cow behavior suggests that the BI-LSTM and CNN-LSTM networks perform the best overall, while the Simple RNN network performs the worst.

Future work in this area could involve further exploration of the capabilities and limitations of different neural network architectures for analyzing animal behavior. This could include testing the networks on different types of input data and investigating the impact of factors such as the length and complexity of the input sequences.

In addition, future work could involve developing more advanced machine-learning models for behavior monitoring and prediction, with the goal of improving animal welfare and productivity. This could involve incorporating additional data sources, such as environmental or physiological data, into the analysis, and using advanced techniques such as transfer learning to leverage pre-existing models and data.

References

[1] V. Lorenzutti, A. S. McLean, and D. Kelton, "Abnormal behaviors in dairy cows: potential impact on health, production and welfare," *Journal of Dairy Science*, vol. 103, no. 12, pp. 10997–11014, 2020.

[2] M. H. Groschup and M. Beekes, "Clinical signs and diagnosis of bovine spongiform encephalopathy," *Journal of veterinary medicine. Series B*, vol. 50, no. 8, pp. 361–366, 2003.

[3] N. Hailat, O. Al-Rawashdeh, and M. Ababneh, "Rabies outbreak in cattle caused by ingestion of contaminated water," *Veterinary Medicine and Science*, vol. 2, no. 3, pp. 234–237, 2016.

[4] M. A. von Keyserlingk, D. Olenick, and D. M. Weary, "Pain and aggression in dairy cows," *Applied Animal Behaviour Science*, vol. 105, no. 1-3, pp. 87–94, 2008.

[5] J. Kaler, L. E. Green, R. Grogono-Thomas, and G. F. Medley, "Activity classification using a single accelerometer placed at the head of dairy cows: Effects of classifier type, window size, and sample frequency," *Computers and Electronics in Agriculture*, vol. 116, pp. 1–8, 2015.

[6] M. S. Kamal, M. J. Uddin, M. A. Hossain, M. S. Rahman, M. K. Uddin, and M. A. Ullah, "Sensor-based disease detection: A case study using accelerometers to recognize symptoms of bovine ephemeral fever," *Computers and Electronics in Agriculture*, vol. 185, p. 106079, 2021.

[7] J. Marais, R. Wolhuter, T. Niesler, and S. Le Roux, "Automatic classification of sheep behaviour using 3-axis accelerometer data," 11 2014.

[8] J. A. Vázquez Diosdado, Z. E. Barker, H. R. Hodges, J. R. Amory, D. P. Croft, N. J. Bell, and E. A. Codling, "Classification of behaviour in housed dairy cows using an accelerometer-based activity monitoring system," *Animal Biotelemetry*, vol. 3, p. 15, Jun 2015.

[9] B. Achour, M. Belkadi, R. Aoudjit, and M. Laghrouche, "Unsupervised automated monitoring of dairy cows' behavior based on inertial measurement unit attached to their back," *Computers and Electronics in Agriculture*, vol. 167, p. 105068, 2019.

[10] J. Haladjian, B. Brügge, and S. Nüske, "An approach for early lameness detection in dairy cattle," in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, UbiComp '17, (New York, NY, USA), p. 53–56, Association for Computing Machinery, 2017.

[11] C. Tobin, D. W. Bailey, M. G. Trotter, and L. O'Connor, "Sensor based disease detection: A case study using accelerometers to recognize symptoms of bovine ephemeral fever," *Computers and Electronics in Agriculture*, vol. 175, p. 105605, 2020.

[12] H. Ito, K. ichi Takeda, K. K. Tokgoz, L. Minati, M. Fukawa, L. Chao, J. Bartels, I. Rachi, and S. A, "Japanese Black Beef Cow Behavior Classification Dataset." Zenodo, Jan. 2022.

[13] D. Lee, Y. Kim, K. S. Cho, T.-K. Kim, Y.-J. Choi, H.-S. Kim, and S. Kim, "Automated time-series segmentation for monitoring animal behavior," *Computers and Electronics in Agriculture*, vol. 161, pp. 162–171, 2019.

[14] P. Llonch, H. Komen, W. Saeys, S. Viazzi, P. van der Tol, I. Halachmi, C. Bahr, and K. Lokhorst, "Monitoring social behaviour of dairy cows with an accelerometer using different classifier models," *Applied Animal Behaviour Science*, vol. 219, p. 104824, 2019.

[15] W. Ouweltjes, E. A. M. Bokkers, I. J. M. de Boer, A. T. M. van Knegsel, S. S. van der Beek, J. C. M. Vernooij, M. J. W. Heetkamp, and C. G. van Reenen, "Monitoring cow behaviour and gastrointestinal motility around calving using a tri-axial accelerometer," *Applied Animal Behaviour Science*, vol. 219, p. 104824, 2019.

[16] L. Gao, W. Li, B. Li, Y. Ding, Y. Huang, and H. Liu, "Activity recognition of dairy cows using multiclass classification with a low-complexity recurrent neural network," *Computers and Electronics in Agriculture*, vol. 157, pp. 567–575, 2019.

[17] Y. Liu, X. Guo, Y. Xie, G. Wang, X. Huang, and L. He, "Deep learning based behavioral identification and classification for dairy cows using accelerometer data," *Computers and Electronics in Agriculture*, vol. 178, p. 105731, 2020.

[18] S. Schulz, A.-R. Abdel-Rahman, K. Reiners, M. Schick, B. Wirthgen, A. Hoeflich, and N. Kemper, "A long short-term memory network for daily behaviour analysis in dairy cows," *Sensors*, vol. 19, no. 15, p. 3307, 2019.

[19] D. Wu, Y. Wang, M. Han, L. Song, Y. Shang, X. Zhang, and H. Song, "Using a cnn-lstm for basic behaviors detection of a single dairy cow in a complex environment," *Computers and Electronics in Agriculture*, vol. 182, p. 106016, 2021.

[20] Y. Wu, M. Liu, Z. Peng, M. Liu, M. Wang, and Y. Peng, "Recognising cattle behaviour with deep residual bidirectional lstm model using a wearable movement monitoring collar," *Agriculture*, vol. 12, no. 8, 2022.

[21] O. R. Bidder, H. A. Campbell, A. Gómez-Laich, P. Urgé, J. Walker, Y. Cai, L. Gao, F. Quintana, and R. P. Wilson, "Love thy neighbour: automatic animal behavioural classification of acceleration data using the k-nearest neighbour algorithm," *PloS one*, vol. 9, no. 2, p. e88609, 2014.