IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 1, March 2024, pp. 524~532
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i1.pp524-532

Deep learning-based classification of cattle behavior using accelerometer sensors

Khalid El moutaouakil, Noureddine Falih
LIMATI Laboratory, Computer Science Department, Polydisciplinary Faculty, University of Sultan Moulay Slimane, Beni Mellal, Morocco

Article Info

Article history:

Received Feb 18, 2023
Revised May 7, 2023
Accepted May 21, 2023

Keywords:

Agriculture 4.0
Deep learning
Livestock farming
Precision livestock
Smart farm

ABSTRACT

The increasing demand for food has led to the adoption of precision livestock, which relies on information and communication technology to promote the best practices in meat production. By automating various aspects of the industry, precision livestock allows for increased productivity, more effective management strategies, and decision-making. The paper proposes a methodology that uses deep learning techniques to automatically classify cattle behavior using accelerometer sensors embedded in collars. The work aims to enhance the efficiency and productivity of the industry by improving the classification of cattle behaviors, which is essential for farmers and barn managers to make informed decisions. We tested three different classification techniques to classify rumination, movement, resting, feeding, salting and other cattle behaviors and we achieved promising results that can contribute to a better understanding and management of cattle behavior in the livestock industry.

Corresponding Author:

Khalid El moutaouakil
LIMATI Laboratory, Polydisciplinary Faculty, Sultan Moulay Slimane University
Mghila, BP 592, Beni Mellal, Morocco
Email: elmoutaouakil.kh@gmail.com

1. INTRODUCTION

The world's urban population is growing rapidly, driven by a combination of overall population growth and the movement of people from rural to urban areas [1]. The United Nations projects that by 2,050, the world's urban population will increase by 2.5 billion people, with nearly 90% of this growth taking place in Asia and Africa [2]. This growth has significant implications for the demand for food, including meat, as urbanization leads to changes in dietary preferences and increases the demand for protein. The shifts in the global economy are affecting the beef cattle industry [3], necessitating a boost in the production and efficiency of high-quality meat. Also, by increasing the production of meat, more profits can be generated from its sale, which is significant given the crucial role that livestock plays in the economy [4].

Precision livestock is the approach to cattle management that relies on information and communication technology to introduce the best practices in meat production [5]. By automating various aspects of the industry, such as optimizing production costs and minimizing environmental effects, this method allows for increased productivity. A significant advantage of precision livestock is that it treats the data of each animal individually [6], enabling decision-making based on their unique potential, including economic objectives and welfare indicators. As a result, more effective management strategies can be implemented [7]. Overall, precision livestock represents a major advancement in the industry [8]. Livestock production activities can be managed either manually or through automation. Manual methods rely on human monitoring of the animals, which can be expensive and lead to inaccuracies in the information recorded [9]. In contrast, automated techniques provide more precise data and can help identify the source of issues and better monitor the animals by enabling quick and accurate tracking of the individual history of each animal [10]. Automation also helps to reduce reading errors, leading to improved quality of production [11].

In order to keep up with the growing demand for food and the expanding population, farmers must enhance their productivity and performance [12]. To achieve this goal, they need to rely on new technologies based on the Agriculture 4.0 standard and adopt innovative techniques to optimize their livestock farms [13]. These technologies can enable the implementation of smart and efficient management strategies through real-time automatic monitoring [14] and the use of advanced techniques such as artificial intelligence.

The primary goal of this study is to create classifying models that use three axial accelerometer sensors data to classify cattle behaviors accurately [15]. The behaviors include Moving, Feeding, Resting, Ruminating, and Salting behaviors that represent the most prominent activities that occupy the animal's time throughout the day [16]. By improving the precision of the classification of these behaviors, the proposed models can contribute to a better understanding of cattle behavior and help in livestock management. The identification and classification of cattle behavior are a very important things for farmers and barn managers to help in decision-making [17]. Machine learning algorithms can classify several behaviors using accelerometers data, as well as video scenes. The use of video scenes and surveillance cameras for monitoring can be very expensive in terms of data processing, storage memory, network bandwidth [18]. In this context, using accelerometer is much more efficient and less expensive [19]. By monitoring cattle behavior, we can detect, among other things: estrus (when too much movement is detected) [20], lameness (short standing times) [21] and signs of diseases (little movements) [22]. This work uses accelerometer data in order to build classifiers that can help improve meat production and livestock management, based on the automatic identification of cattle behavior. It's based on the Japanese black beef cow behavior classification dataset which is among the few datasets available in public access. There are two publications using this dataset to date [23], [24]. We have tested 3 classification models including two models based on decision tree and random forest in addition to a convolutional neural network (CNN) model with our own architecture.

The paper is organized as follows: in the next section, we will explain the specifics of the dataset that we relied on, the model development process, the model architecture and the evaluation metrics. Then in the following section, we will explain the results of the research and at the same time provide a comprehensive discussion. And at the end, in the conclusion section, we will conclude this work and give reference to the prospect of the development of research results and application outlook of further studies in this regard.

2. METHOD

2.1. Dataset

Our proposed approach used the version number 2.0 of the Japanese black beef cow behavior classification [25] to classify cow behaviors using embedded in collars tri-axial accelerometer sensor data. The data was collected using a commercial accelerometer, specifically the Kionix KX122-1037 model, with a sensitivity of 16 bits and a range of +/- 2g. It has been collected on June 12, 2020, from six Japanese black beef cows at a farm owned by Shinshu University in Nagano, Japan, consists of 13 different labeled cow behaviors. The cows were allowed to roam freely in two areas, a grass field and farm pens, and were recorded using Sony FDR-X3000 4K video cameras for one day.

The data is labeled by human observers, including behavior experts and non-experts, who matched the timestamps of the video and accelerometer data. This resulted in 197 minutes of high-quality labeled data, with an accelerometer sampling rate of 25Hz. This means that 25 data samples are generated every second. The dataset contains 85,0529 labeled samples, with columns representing TimeStamp_UNIX and TimeStamp_JST for GPS timestamps in UNIX and JST, respectively, and AccX, AccY, and AccZ for acceleration along the X, Y, and Z axes and the label column. The dataset is divided into six .csv files, one for each cow. We merged the the behavior classes into six main categories. Table 1 provides the number of samples per label category for each cow and Table 2 show the distribution of data per class.

Table 1. The number of samples available for each cow per behavior class

Behavior/Number of samples	Cow 1	Cow 2	Cow 3	Cow 4	Cow 5	Cow 6
Resting (RES)	35,814	47,419	20,501	16,139	11,025	19,996
Ruminating (RUS)	1,620	25,930	11,805	14,820	0	356
Moving (MOV)	6,672	8,541	7,915	17,438	4,846	5,956
Salting (SLT)	204	0	10,654	0	0	0
Feeding (FES)	10,401	2,199	1,300	2,707	3,567	7,849
Other behaviors (ETC)	105,917	103,084	129,297	62,064	53,922	100,571

Table 2. Distribution of data by class

Behaviors	Samples
Resting (RES)	150,894
Ruminating (RUS)	54,531
Moving (MOV)	51,368
Salting (SLT)	10,858
Feeding (FES)	28,023
Other behaviors (ETC)	554,855
Sum	850,529

We have a total of 28,023 samples for Feeding behavior (FES), 51,368 for moving (MOV), 150,894 for Resting (RES), 54,531 for Ruminating (RUS), 10,858 for Salting (SLT) and 554,855 for other behaviors (ETC). Therefore, the total number of samples is 850,529. Also, a sample of the dataset is shown in Figure 1.

AccX	AccY	AccZ	label
0.334	0.424	0.863	ETC
0.329	0.424	0.844	ETC
0.290	0.392	0.873	ETC
0.271	0.360	0.884	ETC
0.277	0.346	0.882	ETC
...	...	...	...
-0.129	1.005	0.191	RES
-0.090	1.027	0.168	RES
-0.040	1.042	0.157	RES
-0.009	1.046	0.154	RES
-0.015	1.032	0.157	RES

Figure 1. Samples of the dataset

2.2. Model development process

Figure 2 shows a schematic of the complete model development process, starting with the input dataset of tri-axis accelerometer data. The pre-processing stage involves filters to eliminate the noise due to sensors malfunction [26] and data normalization to remove differences in the magnitude of characteristic values and facilitate the learning proces. In the feature extraction stage, we segment the raw data and split the dataset into a training set (80%), a validation set (10%), and a test set (10%) then we apply three classification models: Random Forest, decision tree, and a deep learning CNN model with our own architecture. Finally, we perform behavior analysis by classifying the six cattle behaviors using the three classifiers.

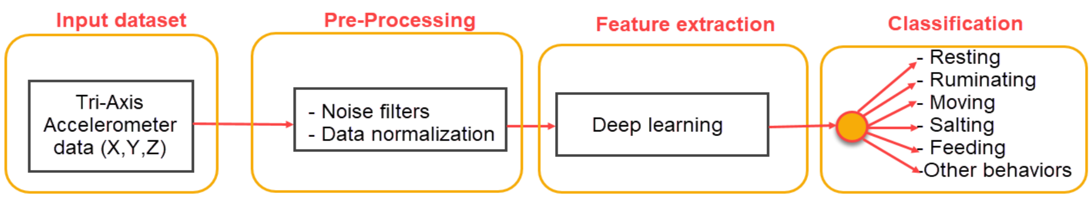

Figure 2. Model development process

We conducted the training and validation of the models using Python 3.11.0 in a Jupyter Notebook development environment. Subsequently, we tested the models on an HP laptop with the following specifications: a 10th generation Core i7 processor and 16GB of DDR4 RAM. In addition, the laptop is running the Windows 11 Pro operating system.

2.3. Model architecture

The pre-processing data stage involves normalizing the input features, reshaping the data, converting the labels to categorical variables, and balancing the classes in the training dataset using bootstrap resampling. The data normalization technique used in our model is the Z-score normalization. In Z-score normalization, the mean of each variable is subtracted from each value in the variable, and then the result is divided by the standard deviation of the variable. This rescales the values to have a mean of 0 and a standard deviation of 1.

Our CNN model proposed architecture comprises of 8 layers, consisting of 3 convolutional layers, 3 max pooling layers, 1 flatten layer, and 2 dense layers. Rectified linear unit (ReLU) is used as an activation function, and we normalized the probability of our classes using the Softmax function. The architecture comprises 126,598 trainable parameters in total. Figure 3 displays the details of each layer of the model.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 3, 1, 32)          2624

 max_pooling2d (MaxPooling2D  (None, 2, 1, 32)         0
 )

 conv2d_1 (Conv2D)           (None, 2, 1, 96)          27744

 max_pooling2d_1 (MaxPooling  (None, 1, 1, 96)         0
 2D)

 conv2d_2 (Conv2D)           (None, 1, 1, 96)          83040

 max_pooling2d_2 (MaxPooling  (None, 1, 1, 96)         0
 2D)

 flatten (Flatten)           (None, 96)                0

 dense (Dense)               (None, 128)               12416

 dense_1 (Dense)             (None, 6)                 774

=================================================================
Total params: 126,598
Trainable params: 126,598
Non-trainable params: 0
_________________________________________________________________
```

Figure 3. CNN model architecture

The input shape of the model is (3, 1, 1), which corresponds to the time-series dataset with 3 features and a single time step. The model consists of three convolutional layers with ReLU activation functions and max pooling layers in between. The output from the last convolutional layer is flattened into a vector and passed through two fully connected layers with ReLU and softmax activation functions, respectively. The model is compiled with the Adam optimizer, categorical cross-entropy loss function, and evaluation metrics of accuracy and F1-score using the macro average, computed for each of the 6 possible classes. The CNN has 32 filters in the first convolutional layer and 96 filters in the second and third convolutional layers. The kernel sizes for the convolutional layers are (9, 9) and (3, 3) for the first and subsequent layers, respectively. The CNN is trained for 100 epochs with a batch size of 64 and the Glorot uniform initializer is used to initialize the kernel weights with a random seed. The output classes are mapped to integer values using a dictionary called labels_map.

2.4. Evaluation metrics

To assess how effectively our classification models are performing, we used several evaluation metrics. It allows us to measure the accuracy and effectiveness of the models. These evaluation metrics provide us with a set of quantitative measures that enable us to compare the performance of the different models and determine which one is the most effective for our specific use case.

2.4.1 Precision

Precision is an an indispensable evaluation metric. It measures the ability of a model to correctly identify positive instances, minimizing false positives. It is the ratio of true positives to the sum of true positives and false positives.

2.4.2. Accuracy

Accuracy is a fundamental evaluation metric. It measures the overall correctness of a model's predictions. It is the ratio of the number of correct predictions to the total number of predictions.

2.4.3. Recall

Recall is a critical evaluation metric. It measures the ability of a model to identify all relevant instances of a class. It is the ratio of true positives to the sum of true positives and false negatives.

2.4.4. F1-score

The F1-score serves as a helpful metric to evaluate model performance. It’s a metric indicating test accuracy, throughout the training and validation of the model for each successive epoch. It measures the accuracy of the models and takes into account Precision and Recall of the test to classify examples as positive or negative. The F1-score of the classification model is calculated as follows:

$$F1 - score = \frac{2(P*R)}{(P+R)} \quad (1)$$

where: P is the precision and R is the recall of the classification model.

2.4.5. Support

Support is the number of instances of a class in the dataset. It is used to calculate the weighted average of different metrics. It is a critical component in calculating various evaluation metrics.

2.4.6. Micro avg

Micro avg is a way of aggregating the metrics across all classes by treating all instances equally. It is the ratio of the sum of true positives across all classes to the sum of true positives, false positives, and false negatives across all classes. Micro avg gives equal weight to each instance and is useful when the dataset is imbalanced.

2.4.7. Weighted avg

Weighted avg is a way of aggregating the metrics across all classes by taking into account the support of each class. It is the weighted average of the metrics for each class, where the weight is the support of the class. Weighted avg gives more weight to the classes with more instances and is useful when the dataset is balanced.

3. RESULTS AND DISCUSSION

3.1. Training results with the CNN model

The CNN architecture is implemented using the Python programming language. It incorporates several libraries, such as pandas, numpy, tensorflow, and scikit-learn. The model underwent training for 100 epochs and attained an accuracy of 99.65% with a loss of 0.98%. Figure 4 displays line plots illustrating a steady rise in the F1 score. It’s a metric indicating test accuracy, throughout the training and validation of the model for each successive epoch. These plots also depict the loss observed during both training and validation phases.

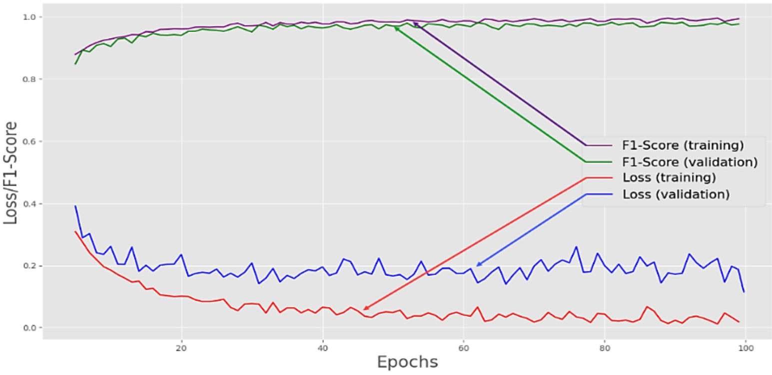

Figure 4. The evolution of F1-score and loss during both training and validation

The confusion matrix serves as a concise summary of the classifier's performance. The rows correspond to the actual class instances, and the columns correspond to the predicted class instances. Figure 5 represent the CNN model confusion matrix.

Table 3 presents a summary of the metrics' values that were obtained during the testing phase of our CNN model. These metrics provide information about how well the model performed in terms precision, recall, and other evaluation measures. By presenting this information in a table, we can easily compare the performance of our model across different metrics and make decisions about its effectiveness.

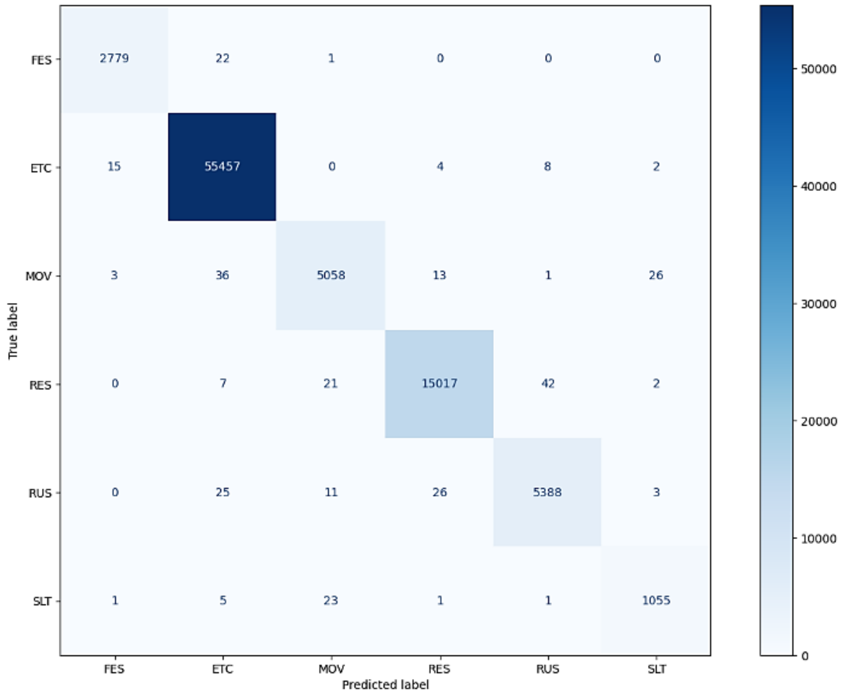

Figure 5. CNN model confusion matrix

Table 3. Classification metrics of the CNN model

Behaviors	Precision	Recall	F1-score	Support
FES	0.9932	0.9918	0.9925	2,802
ETC	0.9983	0.9995	0.9989	55,486
MOV	0.9890	0.9846	0.9868	5,137
RES	0.9971	0.9952	0.9962	15,089
RUS	0.9904	0.9881	0.9893	5,453
SLT	0.9697	0.9715	0.9706	1,086
Accuracy	0.9965	0.9965	0.9965	85,053
Micro avg	0.9896	0.9884	0.9890	85,053
Weighted avg	0.9965	0.9965	0.9965	85,053

The table shows the classification metrics of a CNN model that has been trained on a dataset with 85,053 samples and 6 possible output classes. The precision, recall, and F1-score are computed for each of the classes, as well as the support, which is the number of samples in each class. The micro-average and weighted-average metrics are also provided. The results show that the model has achieved high accuracy, with an overall accuracy of 0.9965. The F1-scores for most of the classes are also high, ranging from 0.9868 to 0.9989. The precision and recall metrics are generally high across all classes, with some classes achieving near-perfect scores. Overall, the results suggest that the CNN model is performing well on the classification task.

3.2. Training results with the random forest-based model

A Random Forest-based model was used to test the database and achieved an accuracy of 72.45%. Table 4 gives an overview of metrics for evaluating the Random Forest model performance. It contains values for the different evaluation metrics. The model's performance is analyzed through these metrics.

Table 4. Classification metrics of the Random Forest-based model

Behaviors	Precision	Recall	F1-score	Support
FES	0.6417	0.2441	0.3537	2,808
ETC	0.7570	0.8874	0.8170	55,486
MOV	0.2517	0.0286	0.0514	5,137
RES	0.6921	0.5861	0.6347	15,089
RUS	0.4830	0.4167	0.4474	5,453
SLT	0.4921	0.4006	0.4416	1,086
Accuracy	0.7245	07245	0.7245	85,053
Micro avg	0.5529	0.4272	0.4576	85,053
Weighted avg	0.6902	0.7245	0.6947	85,053

The results show that the Random Forest-based model has achieved lower accuracy compared to the CNN model, with an overall accuracy of 0.7245. The F1-scores for most of the classes are also lower, ranging from 0.0514 to 0.8170. The class with the lowest F1-score is MOV, with a score of 0.0514. The precision and recall metrics are generally lower across all classes, with some classes achieving relatively low scores.

Overall, the results suggest that the Random Forest-based model is performing less effectively on the classification task compared to the CNN model. This is due to the fact that Random Forests are less suited for modeling sequential data such as time-series, compared to CNNs. The inferior performance of the Random Forest-based model suggests that it is not capable of capturing the complex patterns and dependencies present in the sequential data

3.3. Training results with the decision tree based model

We tested the database using a decision tree based model and achieved an accuracy rate of 63.39%. Table 5 summarizes the metrics that were used to evaluate the performance of the decision tree-based model in classifying cattle behavior. By showing this information, it is easier to compare the performance of the model across different metrics and identify areas where the model can be improved. Overall, the table provides a clear and concise summary of the evaluation metrics used to assess the decision tree model's effectiveness.

Table 5. Classification metrics of the decision Tree-based model

Behaviors	Precision	Recall	F1-score	Support
FES	0.3223	0.3298	0.3260	2,802
ETC	0.7485	0.7401	0.7443	55,486
MOV	0.1492	0.1633	0.1559	5,137
RES	0.5649	0.5694	0.5671	15,089
RUS	0. 3851	0.3809	0.3830	5,453
SLT	0.3801	0.3840	0.3820	1,086
Accuracy	0.6339	0.6339	0.6339	85,053
Micro avg	0.4250	0.4279	0.4264	85,053
Weighted avg	0.6377	0.6339	0.6358	85,053

According to the findings, the decision Tree model performed poorly in comparison to both the CNN and Random Forest models, with an accuracy of 0.6339 and lower F1-scores ranging from 0.1559 to 0.7443, with the MOV class having the lowest score. The precision and recall metrics were also lower across all classes. These results suggest that decision Trees are not as effective as CNNs and Random Forests for modeling sequential data like time-series. However, it's important to keep in mind that the model's performance may vary based on the dataset, and additional testing may be necessary to determine its generalizability.

3.4. Discussion of the results

The decision tree-based model achieved an accuracy rate of 63.39%, which is lower than the accuracy rates of both the random forest and CNN models. This indicates that the decision tree model struggled to capture the underlying patterns in the cattle behavior dataset, and was not able to make accurate predictions. This also highlights the limitations of the decision tree model and underscores the need for alternative approaches in handling this type of data.

On the other hand, the random forest model achieved an accuracy rate of 72.45%, which is higher than the decision tree model but still significantly lower than the accuracy rate achieved by the CNN model. The random forest model is a more complex and advanced version of the decision tree model, which uses multiple decision trees to make predictions. This allows it to capture more complex relationships and interactions between the input variables, resulting in improved accuracy compared to the decision tree model.

However, the CNN model achieved the best performance, achieving an accuracy rate of 99.65%. This suggests that the CNN model was able to learn highly discriminative features from the cattle behavior dataset, which allowed it to make highly accurate predictions. The CNN model is the most suitable option for this particular task, as it was able to provide the highest accuracy rate and best overall performance compared to the other models.

4. CONCLUSION

In conclusion, this article highlights the importance of precision livestock management in the beef cattle industry, which is crucial for meeting the increasing demand for food production. For that, the article proposes a methodology that uses accelerometer sensors embedded in collars to automatically classify cattle behaviors, which can help farmers and barn managers in decision-making. The study used the Japanese black beef cow behavior classification dataset to classify cow behaviors using deep learning techniques, achieving promising results. The use of automated techniques, such as precision livestock, can help in monitoring and managing the livestock industry, leading to increased productivity, efficiency, and improved quality of production. The article concludes that future studies can build on the proposed methodology to enhance the development and application of precision livestock management in the industry

REFERENCES

[1] E. Crist, C. Mora, and R. Engelman, "The interaction of human population, food production, and biodiversity protection," *Science*, vol. 356, no. 6335, pp. 260–264, Apr. 2017, doi: 10.1126/science.aal2011.
[2] A. Mottet, C. de Haan, A. Falcucci, G. Tempio, C. Opio, and P. Gerber, "Livestock: On our plates or eating at our table? A new analysis of the feed/food debate," *Global Food Security*, vol. 14, pp. 1–8, Sep. 2017, doi: 10.1016/j.gfs.2017.01.001.
[3] C. Navarrete-Molina, C. A. Meza-Herrera, M. A. Herrera-Machuca, N. Lopez-Villalobos, A. Lopez-Santos, and F. G. Veliz-Deras, "To beef or not to beef: Unveiling the economic environmental impact generated by the intensive beef cattle industry in an arid region," *Journal of Cleaner Production*, vol. 231, pp. 1027–1035, Sep. 2019, doi: 10.1016/j.jclepro.2019.05.267.
[4] D. Lovarelli, J. Bacenetti, and M. Guarino, "A review on dairy cattle farming: Is precision livestock farming the compromise for an environmental, economic and social sustainable production?," *Journal of Cleaner Production*, vol. 262, p. 121409, Jul. 2020, doi: 10.1016/j.jclepro.2020.121409.
[5] S. Morrone, C. Dimauro, F. Gambella, and M. G. Cappai, "Industry 4.0 and Precision Livestock Farming (PLF): An up to Date Overview across Animal Productions," *Sensors*, vol. 22, no. 12, p. 4319, Jun. 2022, doi: 10.3390/s22124319.
[6] E. Tullo, A. Finzi, and M. Guarino, "Review: Environmental impact of livestock farming and Precision Livestock Farming as a mitigation strategy," *Science of the Total Environment*, vol. 650, pp. 2751–2760, Feb. 2019, doi: 10.1016/j.scitotenv.2018.10.018.
[7] J. L. Kleen and R. Guatteo, "Precision Livestock Farming: What Does It Contain and What Are the Perspectives?," *Animals*, vol. 13, no. 5, p. 779, Feb. 2023, doi: 10.3390/ani13050779.
[8] C. Bahlo, P. Dahlhaus, H. Thompson, and M. Trotter, "The role of interoperable data standards in precision livestock farming in extensive livestock systems: A review," *Computers and Electronics in Agriculture*, vol. 156, pp. 459–466, Jan. 2019, doi: 10.1016/j.compag.2018.12.007.
[9] B. Laberge and A. N. Rousseau, "Rethinking environment control strategy of confined animal housing systems through precision livestock farming," *Biosystems Engineering*, vol. 155, pp. 96–123, Mar. 2017, doi: 10.1016/j.biosystemseng.2016.12.005.
[10] A. Gomez Villa, A. Salazar, and F. Vargas, "Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks," *Ecological Informatics*, vol. 41, pp. 24–32, Sep. 2017, doi: 10.1016/j.ecoinf.2017.07.004.
[11] R. García, J. Aguilar, M. Toro, A. Pinto, and P. Rodríguez, "A systematic literature review on the use of machine learning in precision livestock farming," *Computers and Electronics in Agriculture*, vol. 179, p. 105826, Dec. 2020, doi: 10.1016/j.compag.2020.105826.
[12] O. Calicioglu, A. Flammini, S. Bracco, L. Bellù, and R. Sims, "The future challenges of food and agriculture: An integrated analysis of trends and solutions," *Sustainability (Switzerland)*, vol. 11, no. 1, p. 222, Jan. 2019, doi: 10.3390/su11010222.
[13] N. Yahya, "Agricultural 4.0: Its implementation toward future sustainability," in *Green Energy and Technology*, vol. 0, no. 9789811075773, Springer Singapore, 2018, pp. 125–145. doi: 10.1007/978-981-10-7578-0_5.
[14] K. El Moutaouakil, B. Jabir, and N. Falih, "Agriculture 4.0: Literature Review and Application Challenges in the 'Beni Mellal-Khenifra' region," Oct. 2022. doi: 10.1109/ICOA55659.2022.9934114.
[15] S. Hosseininoorbin *et al.*, "Deep learning-based cattle behaviour classification using joint time-frequency data representation," *Computers and Electronics in Agriculture*, vol. 187, p. 106241, Aug. 2021, doi: 10.1016/j.compag.2021.106241.
[16] B. Sharma and D. Koundal, "Cattle health monitoring system using wireless sensor network: A survey from innovation perspective," *IET Wireless Sensor Systems*, vol. 8, no. 4, pp. 143–151, Aug. 2018, doi: 10.1049/iet-wss.2017.0060.
[17] A. da Silva Santos, V. W. C. de Medeiros, and G. E. Gonçalves, "Monitoring and classification of cattle behavior: a survey," *Smart Agricultural Technology*, vol. 3, p. 100091, Feb. 2023, doi: 10.1016/j.atech.2022.100091.
[18] Y. Guo, D. He, and L. Chai, "A machine vision-based method for monitoring scene-interactive behaviors of dairy calf," *Animals*, vol. 10, no. 2, p. 190, Jan. 2020, doi: 10.3390/ani10020190.
[19] J. Cabezas *et al.*, "Analysis of Accelerometer and GPS Data for Cattle Behaviour Identification and Anomalous Events Detection," *Entropy*, vol. 24, no. 3, p. 336, Feb. 2022, doi: 10.3390/e24030336.
[20] S. Reith and S. Hoy, "Review: Behavioral signs of estrus and the potential of fully automated systems for detection of estrus in dairy cattle," *Animal*, vol. 12, no. 2, pp. 398–407, 2018, doi: 10.1017/S1751731117001975.
[21] M. Alsaaod, M. Fadul, and A. Steiner, "Automatic lameness detection in cattle," *Veterinary Journal*, vol. 246, pp. 35–44, Apr. 2019, doi: 10.1016/j.tvjl.2019.01.005.
[22] 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, Aug. 2020, doi:

10.1016/j.compag.2020.105605.
[23] C. Li *et al.*, "Data Augmentation for Inertial Sensor Data in CNNs for Cattle Behavior Classification," *IEEE Sensors Letters*, vol. 5, no. 11, pp. 1–4, Nov. 2021, doi: 10.1109/LSENS.2021.3119056.
[24] J. Bartels *et al.*, "A 216 µW, 87% accurate cow behavior classifying decision tree on FPGA with interpolated ARCTAN2," in *Proceedings - IEEE International Symposium on Circuits and Systems*, May 2021, vol. 2021-May. doi: 10.1109/ISCAS51556.2021.9401342.
[25] H. Ito *et al.*, "Japanese Black Beef Cow Behavior Classification Dataset." Zenodo, Jan. 2022. doi: 10.5281/zenodo.5849025.
[26] L. Riaboff, L. Shalloo, A. F. Smeaton, S. Couvreur, A. Madouasse, and M. T. Keane, "Predicting livestock behaviour using accelerometers: A systematic review of processing techniques for ruminant behaviour prediction from raw accelerometer data," *Computers and Electronics in Agriculture*, vol. 192, p. 106610, Jan. 2022, doi: 10.1016/j.compag.2021.106610.

BIOGRAPHIES OF AUTHORS

Khalid El moutaouakil In 2017, he earned his Master's degree in computer engineering and systems from the Polydisciplinary Faculty of Sultan Moulay Slimane University in Beni Mellal, Morocco. Currently, he is pursuing his Ph.D. studies in the same faculty and works as a computer science teacher in a high school in Marrakech, Morocco. His research interests lie in Digital Agriculture, Deep Learning, and Information Systems. To reach him, you can contact him via email at elmoutaouakil.kh@gmail.com.

Noureddine Falih In 2013, he obtained a Doctor of Computer Science degree from the Faculty of Sciences and Technologies of Mohammedia, Morocco. Since 2014, he has been working as an associate professor at the Polydisciplinary Faculty of Sultan Moulay Slimane University in Beni Mellal, Morocco. With 18 years of professional experience in several renowned companies, his research interests revolve around Information System Governance, Business Intelligence, Big Data Analytics, and Digital Agriculture. For further communication, he can be reached via email at nourfald@yahoo.fr.