Improved Pig Behavior Analysis Through Strategic Data Preprocessing Framework in Machine Learning

Pranjal Ranjan*, Sanjana Bharadwaj*, Yingqi Pei*, Kenan Burak Aydin†,
Dong Sam Ha*, Gota Morota†, and Sook Shin*
Bradley Department of Electrical and Computer Engineering*
School of Animal Sciences†
Virginia Tech, Blacksburg, Virginia, 24061, USA

{pranjalranjan, sbharadwaj, pyingqi, kenanburak, ha, morota, sook}@vt.edu

2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI) | 979-8-3315-2723-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICTAI62512.2024.00141

***Abstract*—This study presents a novel data preprocessing framework to enhance pig behavior analysis using machine learning techniques. We address the critical issue of data leakage in time series data, which can lead to overfitting and poor generalization in real-world applications. Our approach introduces two key innovations: a non-class-based windowing method and a chronological time sampling technique for train/test splitting. To evaluate these methods, we collected a comprehensive dataset spanning 100 hours of pig behavior over 24 days, using ear-tag sensors and video recordings to capture 12 distinct activities. We evaluate the effectiveness of our preprocessing methods using various machine learning and deep learning models on both time-domain and feature-domain datasets derived from this unique collection. Results demonstrate significant improvements in classification accuracy across all tested models, with increases of up to 15% compared to commonly used data preparation methods. The 2D Residual CNN achieved the highest accuracy of 95.6% in the time domain, while Random Forest performed best in the feature domain with 94.1% accuracy.**

***Index Terms*—Activity Recognition, Behavior Analysis, Pigs, Smart Farm, Machine Learning, Deep Learning, Data Preprocessing, Windowing**

I. Introduction

Activity recognition in pigs is a vital research area within animal behavior and welfare studies [1], [2]. Pigs, like many other animals, exhibit a wide range of behaviors that serve as indicators of their comfort levels and overall well-being [3], [4]. Traditionally, manual, labor-intensive monitoring methods were employed. However, these methods have proven to be inefficient due to challenges in providing continuous monitoring and the potential for human error. To address these limitations, recent advancements have introduced sensor and video camera-based animal monitoring systems in combination with machine learning and deep learning techniques [27], [28].

In sensor-based monitoring systems, mechanical sensors, such as gyroscopes and accelerometers, facilitate the monitoring of these movements and activities, providing a comprehensive understanding of an animal's movements and activities [5], [6]. Gyroscopes, which measure an object's angular velocity, and accelerometers, which measure linear acceleration, are commonly employed in activity recognition systems [7], [8]. In the case of pigs, these systems typically monitor various behaviors, such as lying down, standing, walking, eating, drinking, and interacting with one another [9], [10]. Observing these behaviors helps assess the pig's well-being, identify potential health issues and provide a non-invasive monitoring method compared to traditional manual observation [11], [12]. Previous studies have used traditional machine learning algorithms with varying success, often relying on hand-crafted features extracted from raw sensor data [19]–[22]. This process can be time-consuming and may not effectively capture complex patterns in the data. Furthermore, the potential for leveraging deep learning algorithms for this task on motion sensor data remains largely unexplored [23], [24], [29].

Notably, several studies have overlooked the data leakage problem – that refers to the use of leaked information during model training and validation which would not be available in the prediction stage. This issue mostly occurs when the test data spills over to the training/validation data, thereby making the results invalid. Oftentimes, this can lead to overly optimistic modeling performance (overfitting) on the test data, since the patterns of the data were already revealed to the model during training. Consequently, the same model can show poor performance when it is applied to a new, unlabeled data.

In this paper, we present two different approaches that we use strategically to address the data leakage problem in data preprocessing phase [25]. Firstly, we use non-class-based data segmentation method, to maintain the temporal order of data. Typically, the accelerometer data readings exhibit strong periodic patterns, which are used to segment the data into time windows [26]. For instance, in supervised learning, the data points labeled as the same activity classes (such as lying, eating, or standing, etc.) are grouped together into one data segment. However, such a class-based data segmentation can open a room for data leakage. For instance, data points labelled as a specific activity are grouped together even if they are in the different time segments. This breaks the natural time order in the timeline of the data and assumes that model is aware of the labels in advance, resulting in data leakage. To avoid such an issue, in non-class-based data segmentation, we segmentize the data into fixed size time windows. The size of each time window in this time-based data segmentation approach is heuristically decided, ensuring the temporal order

2375-0197/24/$31.00 ©2024 IEEE
DOI 10.1109/ICTAI62512.2024.00141

of all data points is intact along the original time line.

Secondly, we use a strategic train/test split method that can prevent the data leakage problem. Often, many pig behavior analysis studies employ random or non-chronological data split, a method where data points are arbitrarily selected and assigned to train or test data. For instance, in random split method, future data segments can become a part training set that influences the model's decisions during testing, resulting in overly positive results. However, in real-time data analysis, the model would not have access to the future data since it is yet to occur, leading to a drop in model performance. The strategic split method used in this study addresses this issue by splitting the data time-wisely, which aids to maintain the temporal order of window segments. This approach ensures that future data points are kept exclusively in the test set, reflecting real-world scenarios.

To illustrate the benefits of using the proposed methods, we compare the classification performance of various machine learning and deep learning models using:

1) The proposed train/test split vs non-chronological train/test split (using proposed non-class-based data segmentation)
2) The proposed data segmentation method vs class-based data segmentation method (using proposed train/test split)
3) The proposed data segmentation and train/test split vs class-based data segmentation and chronological train/test split.

Our results showed a marked improvement of 15% on using the methods over the common approaches. [1]

II. Data Acquisition

A. Sensors

The sensor used in this study to record the linear acceleration and angular velocity across three dimensions was the MetaMotionC (MMC) sensor produced by MBIENTAB, as seen in Figure 1. MMC is a wearable device that offers real-time and continuous monitoring of motion and environmental sensor data, with a sampling rate of 50 Hz. The inertial measurement unit of MMC measures 3-axis acceleration and angular velocity.

The incorporation of ultra-low-power features, high-value sensors, and a coin cell battery architecture into a compact device renders MMC ideal for deployment in pig ear tag sensors. Additionally, MMC houses an ARM Cortex M4F processor, along with an onboard 9-DoF IMU (Inertial Measurement Unit), a high-precision altimeter, and wireless communication capabilities at 33 kb/s on the 2.4GHz frequency, making it suitable for activity classification applications.

The sensor measurement is 7/8 inch in diameter and 1/4 inch in thickness, while the pink case is 1.12 inch in the length as seen in Figure 2. Each data point contained acceleration and angular velocity in three axes, a timestamp, and a counter value. The timestamp and the counter value are used to check the validity of the data.

B. Sensor Placement and Camera Setup

To complement the sensor data, we strategically placed a camera on the barn's roof. This camera provided a visual record of the pigs' activities and was instrumental in our process of manually annotating the motion data captured by the ear-tag sensor. The camera used, is an RGB sensor camera to capture videos, at a speed of thirty frames per second. The video recordings provided ground truth for data annotations.

C. Data Collection

One of the highlights of our study is the dataset which we collected and annotated. Our data collection efforts involved observation of two pigs over a period of twenty four days during the fall of 2022, accumulating a comprehensive dataset spanning a total of one hundred hours. We used both the MetaMotionC sensor and the RGB camera to collect the data. The pigs were of different sizes and were kept together in a pigpen. The pigpen was large enough for the pigs to move around freely during the data collection. Meanwhile, sufficient food and water were provided in the pigpen so that they could eat and drink at any time. A few toys were provided for the pigs as well.

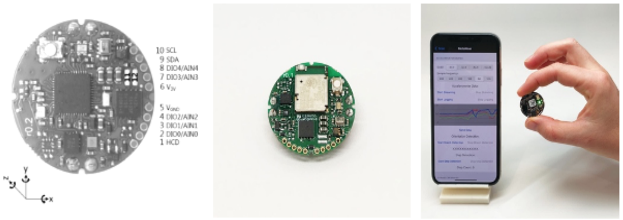

Fig. 1. The MMC circuit board (left and center), a hand holding a coin next to a cell phone (right). The pictures are courtesy of MBIENTAB (https://mbientlab.com)

One evaluator monitored the pigs' behaviors using a closed-circuit television in a nearby room and checked in the event a pig attempted to damage the sensor node. The data were collected at two different periods for a day, which helped diversify the collected data. Each period was adjusted accordingly to the availability of the farm manager.

D. Data Labeling

After the phase of raw data collection, using the sensor and the camera, we reviewed one hundred hours of data to manually annotate the acquired data using SensiML Data Capture Lab. This tool proved invaluable for labeling various events within the sensor data, offering user-friendly graphing tools and a media player for synchronizing video and audio files with the sensor data. Our annotated dataset spanned a time duration of forty hours and encompassed data from twelve distinct pig activities. Table I provides the distribution of different activities observed in the collected data.

[1] Common approaches here refers to class-based data segmentation and chronological train/test split methods

However, it is important to note that this meticulously curated dataset, while rich in information, was not without its challenges. Noise and outliers were present in the data due to a variety of factors, including sensor failures, transmission errors, and intermittent battery issues. The need for data preprocessing became evident to address these issues and enhance prediction performance, a topic we delve into in the following section.

III. Experimental Setup

A. Dataset Preprocessing

To ensure the quality and integrity of the collected sensor data, we performed several essential preprocessing steps: outlier detection, data cleaning, and standardization. This comprehensive approach formed a solid foundation for the reliable and accurate classification of pig behaviors in our study.

We initiated the data preprocessing by detecting and addressing erroneous data points through a robust outlier detection process. This step is essential as outliers can introduce noise and inaccuracies into the dataset, potentially leading to skewed model performance and unreliable predictions. To identify these outliers, we employed the robust statistical interquartile range (IQR) method, chosen for its effectiveness in handling non-normally distributed data and its robustness against extreme values. The IQR method works by calculating the difference between the first quartile (Q1) and third quartile (Q3) of the data. Any data point falling below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR is considered an outlier. This method allows for the detection of outliers while taking into account the spread of the middle 50% of the data, making it less sensitive to extreme values compared to methods based on standard deviation.

Once the outliers were detected, we employed a data cleaning process to address these anomalous points. Instead of simply removing the outliers, which could lead to loss of potentially important information, we used linear interpolation to replace them with estimated values. This approach was chosen to maintain the continuity and temporal structure of the time series data. Mathematically, for an outlier point at time t between two known points (t_0, y_0) and (t_1, y_1), the interpolated value y is calculated as:

$$y = y_0 + (y_1 - y_0)\frac{t - t_0}{t_1 - t_0} \tag{1}$$

This method preserves the overall trend of the data while removing potentially erroneous extreme values, ensuring that the temporal consistency of the dataset is maintained.

Following outlier detection and data cleaning, we performed standardization on the values for each dimension of the accelerometer and gyroscope sensors. The standardization process involves scaling the data to have a mean of zero and a standard deviation of one. Mathematically, for each feature x, we applied the following transformation:

$$x_{standardized} = \frac{x - \mu}{\sigma} \tag{2}$$

Where μ is the mean of the feature and σ is its standard deviation. This standardization ensures that each feature contributes equally to the model, improves the numerical stability of many machine learning algorithms, and helps in faster convergence during the training of neural networks.

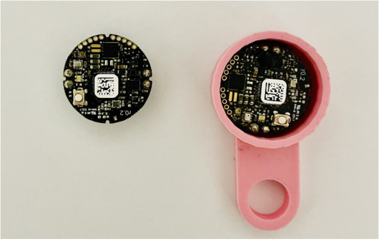

Fig. 2. The coin sensor and the sensor case used in the study

B. Proposed Time Window Segmentation Method

Windowing is a crucial preprocessing step in sensor-based activity recognition, involving the division of continuous sensor data into smaller time intervals, or windows, for classification purposes. Each window typically represents a few seconds of data, labeled with the activity performed during that period [16]. The choice of windowing method significantly impacts the performance and real-world applicability of the resulting classification model.

The traditional approach, known as ”class-based windowing,” involves sorting data by activity labels before creating windows. This method follows these steps:

1) Sort all data points by their associated activity labels.
2) Create fixed-size windows from this sorted data.
3) Assign each window the label of the majority activity within that window.

While intuitive, this approach assumes prior knowledge of activity labels and system access to this information during both training and testing. This assumption introduces significant bias, as the system gains access to activity labels during training that would be unavailable during real-world testing. Consequently, models trained on data windowed in this manner may show satisfactory performance on training data but exhibit poor generalization to new, out-of-distribution data.

To mitigate this bias, we propose a novel ”non-class-based windowing” technique. Our approach involves:

1) Maintaining the chronological order of the recorded dataset.
2) Creating fixed-size windows from this temporally ordered data, without reference to activity labels.
3) Assigning each window the label of the majority activity within that window, but only for model training and evaluation purposes.

This non-class-based windowing approach offers several significant advantages that enhance the robustness and real-world applicability of our pig behavior classification system. By preserving the temporal structure of the data, it allows our models to learn from realistic patterns and sequences of pig behaviors, capturing the natural flow and transitions between activities. The approach effectively eliminates label leakage, accurately reflecting real-world scenarios where activity labels are unavailable during testing, thus preventing the model from relying on information it wouldn't have access to in practice. This leads to improved generalization to new, unseen data, particularly in out-of-distribution scenarios where the distribution of activities may differ from the training set. Furthermore, it provides a more realistic evaluation framework, closely mimicking real-world application conditions and giving a truer picture of how the model would perform when deployed in actual farm environments. Collectively, these advantages contribute to a more robust, reliable, and practically applicable pig behavior classification system.

Our approach ensures robustness and accurate recognition of activities, even in scenarios with differently interleaved activities compared to the training data. This is crucial for the practical application of our pig behavior classification system, where activity sequences may vary significantly across different environments or time periods.

Following the application of our non-class-based windowing technique, we refined our dataset by eliminating low-frequency classes. We then obtained a dataset with six primary classes of pig behaviors, each containing a comparable number of samples. This balanced dataset, which we refer to as the 'time domain dataset', forms the foundation of our subsequent analysis and modeling efforts.

TABLE I
PERCENTAGE OF SIX MAJOR BEHAVIORS

Class	***Proportion (%)***
Drinking	3.53
Eating	44.6
Interacting With Each Other	3.73
Laying	24.85
Standing	8.68
Walking	14.58

C. Feature Extraction

In machine learning, feature extraction transforms raw data into numerical features that can be effectively processed by algorithms while preserving the underlying data distribution [13], [14]. For our pig behavior analysis, we employed a comprehensive approach, deriving features from both time and frequency domains to capture a wide range of characteristics in the pig movement data.

We began by extracting statistical features from the time domain representations of our sensor data. These features were computed for each axis (x, y, and z) of both the accelerometer and gyroscope readings, resulting in 36 time domain features. The features include mean, standard deviation, minimum, maximum, median, and interquartile range (IQR) of the sensor readings. The mean provides a measure of central tendency, standard deviation captures data variability, minimum and maximum define the range of motion, median offers a robust measure of central tendency, and IQR provides a measure of statistical dispersion robust to outliers.

These time domain features are commonly chosen in activity recognition studies due to their low complexity and low computational power consumption. They provide valuable insights into the pigs' movements, such as range of motion, speed, and stability [15]. Their clear physical interpretations also facilitate understanding their relationship to pig movements.

To complement the time domain features and capture periodic patterns, we also extracted features from the frequency domain. We converted the time domain dataset into a frequency domain representation using the Fast Fourier Transform (FFT) method. From this, we extracted several features: sample frequencies, phase, maximum amplitude frequency, power spectral density (PSD), power spectral entropy, and weighted frequencies.

These frequency domain features were chosen for their ability to detect and quantify patterns that may not be apparent in the time domain. They are particularly useful for capturing repetitive movements or cyclic patterns in pig behavior. For example, the maximum amplitude frequency can help identify the primary rhythm of an activity, such as the step frequency during walking. The PSD can distinguish between activities with similar time domain characteristics but different frequency profiles. Power spectral entropy can help differentiate between more regular activities and more chaotic ones.

After extracting both time and frequency domain features, we concatenated all features to create a comprehensive 'feature domain dataset'. This dataset combines the strengths of both domains: time domain features capture instantaneous characteristics and overall statistical properties of the movements, while frequency domain features capture rhythmic and periodic aspects. By utilizing both time and frequency domain features, we ensure a comprehensive representation of the pigs' movement patterns. This multi-domain approach allows our models to leverage a rich set of information, capturing both the immediate physical characteristics of the movements and the underlying rhythmic patterns.

D. Proposed Train/Test Split Method

The process of dividing a dataset into training and testing subsets is a critical step in the development and evaluation of machine learning models. Traditionally, random sampling has been a commonly employed technique for creating these subsets, as it aims to provide an unbiased representation of the population. This method involves randomly selecting data points from the entire dataset to form the training and testing sets, operating under the assumption that each data point is independent and identically distributed. While this approach is effective for many types of data, it presents significant challenges when applied to time series data, such as the pig behavior data in our study. The fundamental issue lies in the inherent temporal structure and dependencies present in time series data, which random sampling fails to preserve.

Time series data, by its very nature, exhibits strong sequential dependence. This means that the value of a data point

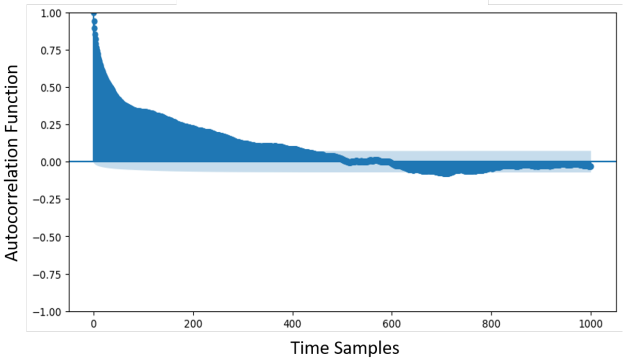

Fig. 3. This ACF diagram provides insights into the dependency among different activities of pigs along the time lags. The higher autocorrelation values corresponding to the time samples between 0 and 200 demonstrate that pig activities during those time lags are highly correlated and those between 600 and 1000 are not correlated.

at a specific time is often heavily influenced by the values of preceding data points. In the context of our pig behavior study, the activity a pig is engaged in at any given moment is not independent of its recent past activities. For instance, if a pig is currently eating, it's more likely that it was approaching the feeding area in the immediate past, and less likely that it was sleeping. This sequential nature leads to significant correlations between adjacent data points in the time series, a characteristic that violates the fundamental assumption of independence in random sampling.

To illustrate the highly correlated nature of our time series dataset, we employed an autocorrelation function (ACF) analysis. The ACF measures the correlation between a time series and a lagged version of itself, providing insights into the temporal dependencies within the data. Mathematically, the ACF is defined as:

$$\rho_k = \frac{\sum_{t=1}^{n-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2} \tag{3}$$

Where ρ_k is the autocorrelation at lag k, x_t is the value at time t, $\bar{x}$ is the mean of the series, and n is the total number of observations. The resulting ACF plot (Figure 3) for our collected dataset clearly highlights the strong temporal correlations present in the data. The plot shows high autocorrelation values for small lag values, indicating strong short-term dependencies, with the correlation gradually decreasing for larger lags. This pattern is typical of time series data where recent past values have a strong influence on current values.

Given these temporal dependencies, applying random sampling to our time series data would likely result in a phenomenon known as data leakage between the training and validation sets. Data leakage occurs when the model gains access during training to information that would not be available in a real-world prediction scenario. In the context of time series data, this often manifests as the model having access to future data points during training, which would not be possible in a real-time prediction task. This leakage can lead to overfitting, where the model performs exceptionally well on the training data but fails to generalize to new, unseen data.

The consequences of overfitting in pig behavior analysis can be severe. It can result in unreliable predictions when the model is applied to new data, significantly impacting the performance and practical utility of the system. For instance, a model that has overfit to the training data might fail to accurately classify pig behaviors in slightly different environmental conditions or with pigs that exhibit subtly different behavior patterns.

To address these challenges and mitigate the risk of data leakage, we propose and implement a method we term ”chronological time sampling” for dividing our dataset into training and test sets. This approach respects and preserves the temporal order of the recorded data, thereby preventing data leakage between these sets. The process of chronological time sampling involves partitioning the dataset sequentially according to the recording time, maintaining the natural order of events.

In our analysis, we implemented this method by designating the last quarter (25%) of the chronologically ordered dataset as the test set, while the first three-quarters (75%) were used as the training set. This division strategy ensures that the model is always trained on past data and tested on future data, mimicking real-world scenarios where predictions must be made based solely on historical information.

This chronological splitting approach offers several key advantages that enhance the robustness and real-world applicability of our pig behavior classification model. It preserves the natural temporal structure and transitions between behaviors, allows the model to learn from realistic activity sequences, and prevents future data leakage during training. This method provides a more accurate representation of real-world performance, captures temporal dynamics and evolving patterns in pig behavior, and encourages the model to learn generalizable patterns rather than overfitting to specific instances. By addressing the unique challenges of time series data in pig behavior analysis, our approach establishes a solid foundation for developing reliable and practically applicable classification models.

IV. Classification Models

The primary objective of this study was to conduct a comparative analysis between commonly used data preparation strategies with the one proposed in this work. To evaluate the performance of these strategies, we employed the widely recognized metric of classification accuracy (%). The selection of machine learning and deep learning models for benchmarking was made with careful consideration of their suitability for the task.

For traditional machine learning, we used decision trees, random forest, and K-Nearest Neighbors for benchmarking. Among deep learning methods, variants of convolutional neural networks and recurrent neural networks were utilized.

The Decision Trees (DT) model is an algorithm that recursively partitions input data into subsets. The algorithm selects a feature at each node that optimally separates the data into different classes and continues until reaching a leaf node, where it outputs the class label. We chose the Gini Impurity (2) to measure the quality of a split in Decision Trees.

$$Gini(p) = 1 - \sum_{i=1}^{C} p_i^2 \tag{4}$$

Random Forest is an ensemble learning method that creates multiple decision trees and combines their results. It employs bootstrapping to sample the training data randomly and trains a decision tree on each subset [18]. The final output is determined by averaging the predictions of each decision tree.

The K-Nearest Neighbors model compares the distance between samples and selects the K-nearest samples to the test sample. The test sample's class is determined by the majority class of the K-nearest neighbors.

In the realm of deep learning models, we leveraged the 1-D convolutional neural network (CNN) is designed to process one-dimensional data, such as time series data. The model comprises of convolutional layers (3) that learn features from the input data and pooling layers that reduce output dimensionality [17]. The 1-D Residual CNN model is a 1-D CNN variant that utilizes residual connections to enhance information flow through the network, allowing the model to learn more complex features by reusing earlier features. The 2-D CNN model is a convolutional neural network designed to process two-dimensional data, such as images. The 2-D Residual CNN model is based on the popular ResNet-34 architecture, known for its superior performance in classification tasks, because of the use of skip connections.

$$Y_i = \sum_{k=-\infty}^{\infty} X_k W_{i-k} \tag{5}$$

The long short-term model (LSTM) model is a type of recurrent neural network (RNN) designed to process sequential data, such as time series data. The model consists of a chain of LSTM units that learn long-term dependencies in the input data. The hypothesis behind using LSTM is that these models are capable of learning temporal dependency patterns which is useful when dealing with time-series data such as ours. Moreover, LSTM can be integrated with other models such as CNN, to capture both long temporal dependencies and local trend features. To examine this feature, we also implement a hybrid CNN+LSTM model. The model first extracts features from the input data using a convolutional neural network and then passes the features through an LSTM layer to learn long-term dependencies.

V. Experimental Results

In this section, we present the evaluation results of various machine learning and deep learning models using different data preparation methods. The analysis is divided into three parts, each corresponding to a specific comparison. It is important to note here that 'Others' essentially refers to class-based segmentation and random train/test split.

The first part of the analysis compares the proposed chronological time split method against the random train/test split method (referred as 'Others' in the table), while using the common class-based windowing approach (Table II). The results show that the proposed splitting method significantly improves the performance of all models in both time and feature domains. For example, the 1-D CNN model accuracy increased from 77.5% with the common method to 89.2% using the proposed method in the time domain. Similarly, in the feature domain, the accuracy increased from 78.8% to 86.7%. This trend is consistent across all the models tested, indicating the advantages of the proposed splitting method.

In the second part of the analysis, we evaluate the impact of the proposed non-class-based windowing method compared to the class-based windowing method (referred as 'Others' in the table), using the common random splitting method (Table III). The results demonstrate that the proposed windowing method leads to better performance in both the time and feature domains for most of the models. For example, the 2-D Residual CNN model achieved 88.1% accuracy in the time domain with the proposed method, compared to 80.0% using class-based windowing. Similar improvements were observed for other models as well, highlighting the benefits of the non-class-based windowing method.

TABLE II
Comparison of classification performances on datasets with proposed train/test split vs other train/test split

Model	Time Domain		Feature Domain	
	Others	Proposed	Others	Proposed
1-D CNN	77.5	89.2	78.8	86.7
1-D Res-CNN	79.2	90.1	79.5	87.5
2-D CNN	78.6	88.4	79.1	87.1
2-D Res-CNN	**81.2**	**90.5**	80.0	88.5
LSTM	78.5	87.4	78.2	88.1
CNN+LSTM	77.6	88.1	78	86.9
Random Forest	70.1	82.1	**80.5**	**91.1**
Decision Tree	56.2	70.8	64.5	87.2
KNN	52.1	61.2	60.3	82.2
Average	72.3	83.1	75.4	87.3

TABLE III
Comparison of Classification Performances on datasets segmented with proposed windowing method vs Segmented with other Windowing method

Model	Time Domain		Feature Domain	
	Others	Proposed	Others	Proposed
1-D CNN	77.5	87.1	78.8	86.2
1-D Res-CNN	79.2	87.5	79.5	87.1
2-D CNN	78.6	86.8	79.1	86.4
2-D Res-CNN	**80.0**	**88.1**	80.0	87.5
LSTM	78.5	86.8	78.2	85.7
CNN+LSTM	77.6	87.1	78.0	85.9
Random Forest	63.2	80.4	**81.1**	**90.2**
Decision Tree	56.1	67.2	64.5	86.8
KNN	52.1	59.2	60.3	78.0
Average	71.4	81.1	75.5	85.9

The third part of the analysis focused on the combined efficiency, comparing the performance of the models using

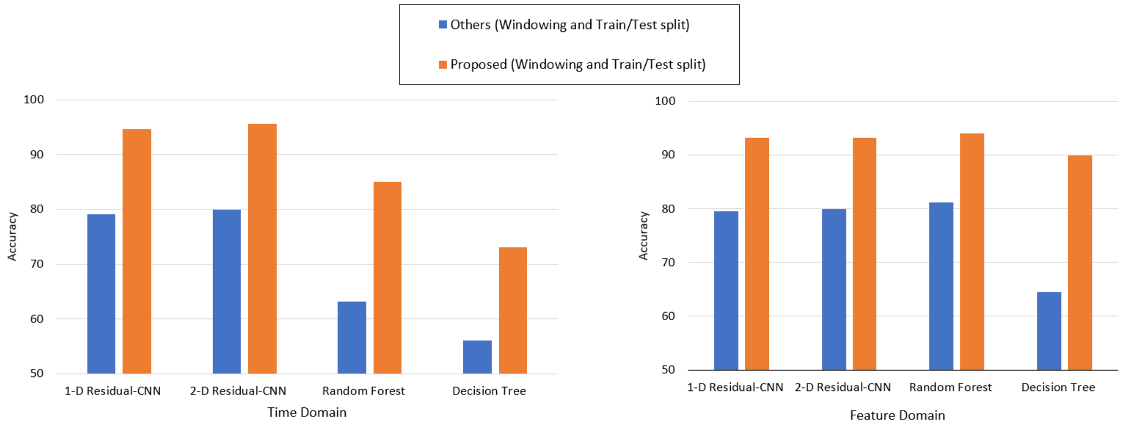

Fig. 4. ML-based pig activity classification results with different data preparation methods on (a) time domain data and (b) feature domain data

TABLE IV
COMPARISON OF CLASSIFICATION PERFORMANCES ON DATASET WITH PROPOSED SPLIT AND WINDOWING VS OTHER SPLIT AND WINDOWING

Model	**Time Domain**		**Feature Domain**	
	Others	**Proposed**	**Others**	**Proposed**
1-D CNN	77.5	94.4	78.8	93.2
1-D Res-CNN	79.2	94.75	79.5	93.2
2-D CNN	78.6	90.5	79.1	91.8
2-D Res-CNN	**80.0**	**95.6**	80	93.3
LSTM	78.5	93.0	78.2	93.0
CNN+LSTM	77.6	93.2	78	91.5
Random Forest	63.2	85.0	**81.2**	**94.1**
Decision Tree	56.1	73.1	64.5	90.0
KNN	52.1	60	60.3	85.1
Average	71.4	86.6	75.4	91.7

both the commonly used class-based windowing and random split methods (referred as 'Others' in the table) against the proposed windowing and splitting methods (Table IV). The results reveal that the combination of the proposed methods further enhances the model's performance in both time and feature domains. Figure 4 represents the obtained results. In Figure 4a, the 2-D residual network stands out with the highest classification accuracy among the presented methods in the time domain. We conduct a comparative analysis, examining the performance of both data preprocessing methods individually and when utilized in combination. Similarly, in Figure 4b, showcases results in the feature domain, with the random forest method demonstrating the highest accuracy. Notably, we observe an improvement in performance when applying our proposed methods even in the feature domain.

For example, the 1-D Residual CNN model achieved an accuracy of 79.2% using the commonly used methods in the time domain, which increased to 94.75% with the proposed methods. A similar trend was observed in the feature domain as well, with accuracy increasing from 79.5% to 93.2%.

This improvement is consistent across all tested models, indicating that the proposed methods are effective in improving pig behavior analysis.

The results demonstrate the effectiveness of proposed methods in addressing the data leakage issue observed in other commonly used methods. The non-class-based windowing approach divides the sensor data into windows according to the natural time order, without prior sorting by activity labels. This ensures that the system does not utilize activity labels during dataset creation, reflecting real-world scenarios where activity labels are unavailable during testing. The chronological time split method, on the other hand, preserves the temporal order of the data, preventing the model from learning patterns that are specific to the training set. The combination of these methods offers a promising framework for improving the accuracy and effectiveness of pig activity recognition, thereby contributing to advancements in precision livestock farming practices.

VI. CONCLUSION

In conclusion, this study presents a novel data preprocessing framework that significantly enhances the accuracy and reliability of pig behavior analysis using machine learning techniques. Our approach, which introduces a non-class-based windowing method and a chronological time sampling technique for train/test splitting, effectively addresses the critical issue of data leakage in time series data. The experimental results demonstrate substantial improvements in classification accuracy across various machine learning and deep learning models, with increases of up to 15% compared to commonly used data preparation methods. The 2D Residual CNN achieved the highest accuracy of 95.6% in the time domain, while Random Forest performed best in the feature domain with 94.1% accuracy.

These findings underscore the critical importance of appropriate data preparation in pig behavior analysis and offer a robust framework for enhancing the reliability and real-world applicability of activity recognition systems in precision livestock farming. By preserving the temporal structure of the data and preventing data leakage between training and testing sets, our approach enables more accurate and generalizable models for pig behavior classification. Future work could explore the application of these preprocessing techniques to larger and more diverse datasets, as well as investigate their effectiveness in real-time behavior monitoring systems for practical implementation in farm settings.

Acknowledgement

This research was supported in part by the National Science Foundation grant with the award number 2106987.

References

[1] Marchant-Forde, J. N. (2009). Welfare of dry sows. In The Welfare of Pigs (pp. 95-139). Springer, Dordrecht.

[2] Rault, J. L., Lay, D. C., & Marchant-Forde, J. N. (2011). Castration induced pain in pigs and other livestock. Applied Animal Behaviour Science, 135(3), 214-225.

[3] Herskin, Mette & Jensen, Karin (2000). Effects of Different Degrees of Social Isolation on the Behaviour of Weaned Piglets Kept for Experimental Purposes. Animal Welfare, 9(3), 237-249.

[4] Kashiha, M. A., Bahr, C., Ott, S., Moons, C. P., Niewold, T. A., Tuyttens, F., & Berckmans, D. (2014). Automatic monitoring of pig locomotion using image analysis. Livestock Science, 159, 141-148.

[5] Noda, Takuji, Kawabata, Yuuki, Arai, Nobuaki, Mitamura, Hiromichi, & Watanabe, Shun (2014). Animal-mounted gyroscope/accelerometer/magnetometer: In situ measurement of the movement performance of fast-start behaviour in fish. Journal of Experimental Marine Biology and Ecology, 451, 55-68. doi:10.1016/j.jembe.2013.10.031

[6] Tian, Fuyang, Wang, Jun, Xiong, Benhai, Jiang, Linshu, Song, Zhanhua, & Li, Fa-De (2021). Real-Time Behavioral Recognition in Dairy Cows Based on Geomagnetism and Acceleration Information. IEEE Access, 9, 109497-109509.

[7] Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013). A public domain dataset for human activity recognition using smartphones. ESANN, 3, 3.

[8] Iqbal, M.W., Draganova, I., Morel, P.C.H., Morris, S.T. (2021). Validation of an Accelerometer Sensor-Based Collar for Monitoring Grazing and Rumination Behaviours in Grazing Dairy Cows. Animals, 11, 2724.

[9] Ding, Qi-an, Chen, Jia, Shen, Ming-xia, & Liu, Long-shen (2022). Activity detection of suckling piglets based on motion area analysis using frame differences in combination with convolution neural network. Computers and Electronics in Agriculture, 194, 106741. doi:10.1016/j.compag.2022.106741

[10] Zhang, K., Li, D., Huang, J., & Chen, Y. (2020). Automated Video Behavior Recognition of Pigs Using Two-Stream Convolutional Networks. Sensors, 20(4), 1085. doi:10.3390/s20041085

[11] Matthews, S.G., Miller, A.L., & PlÖtz, T. (2017). Automated tracking to measure behavioural changes in pigs for health and welfare monitoring. Sci Rep, 7, 17582.

[12] Matthews, S.G., Miller, A.L., Clapp, J., Plötz, T., & Kyriazakis, I. (2016). Early detection of health and welfare compromises through automated detection of behavioural changes in pigs. The Veterinary Journal, 217, 43-51.

[13] Huynh, T. & Schiele, B. (2005). Analyzing features for activity recognition. Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages and Technologies, 159-163. Springer.

[14] Stachowicz, Joanna, Nasser, Roland, Adrion, Felix, & Umstätter, Christina (2022). Can we detect patterns in behavioral time series of cows using cluster analysis? Journal of Dairy Science, 105(12), 9971-9981.

[15] Domun, Yuvraj, Pedersen, Lene Juul, White, David, Adeyemi, Olutobi, & Norton, Tomas (2019). Learning patterns from time-series data to discriminate predictions of tail-biting, fouling and diarrhoea in pigs. Computers and Electronics in Agriculture, 163, 104878.

[16] Alghamdi, Saleh, Zhao, Zhuqing, Ha, Dong S., Morota, Gota, & Ha, Sook S. (2022). Improved pig behavior analysis by optimizing window sizes for individual behaviors on acceleration and angular velocity data. Journal of Animal Science, 100(11), skac293. November.

[17] Cheng, Man, Yuan, Hongbo, Wang, Qifan, Cai, Zhenjiang, Liu, Yueqin, & Zhang, Yingjie (2022). Application of deep learning in sheep behaviors recognition and influence analysis of training data characteristics on the recognition effect. Computers and Electronics in Agriculture, 198, 107010.

[18] Valletta, John Joseph, Torney, Colin, Kings, Michael, Thornton, Alex, & Madden, Joah (2017). Applications of machine learning in animal behaviour studies. Animal Behaviour, 124, 203-220.

[19] Fan, S., Jia, Y., & Jia, C. (2019). A Feature Selection and Classification Method for Activity Recognition Based on an Inertial Sensing Unit. Information, 10(10), 290.

[20] Bagnall, A., Lines, J., Hills, J., & Bostrom, A. (2015). Time-series classification with cote: the collective of transformation-based ensembles. IEEE Transactions on Knowledge and Data Engineering, 27(9), 2522-2535.

[21] Bostrom, A. & Bagnall, A. (2015). Binary shapelet transform for multiclass time series classification. International conference on big data analytics and knowledge discovery, 9263, 257-269. Springer.

[22] Baydogan, M. G., Runger, G., & Tuv, E. (2013). A bag-of-features framework to classify time series. IEEE transactions on pattern analysis and machine intelligence, 35(11), 2796-2802.

[23] Wang, Z., Yan, W., & Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. 2017 International joint conference on neural networks (IJCNN), 1578-1585. IEEE.

[24] Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review. Data Mining and Knowledge Discovery, 33(4), 917-963.

[25] Zhang, L., Huang, D., & Chen, X. (2020). An Intelligent Pig Behavior Recognition System Based on Deep Learning. Journal of Sensors, 2020, 8880775.

[26] Escalante, Hugo Jair, Rodriguez, Sara V., Cordero, Jorge, Kristensen, Anders Ringgaard, & Cornou, Cécile (2013). Sow-activity classification from acceleration patterns: A machine learning approach. Computers and Electronics in Agriculture, 93, 17-26.

[27] Aguilar-Lazcano, C.A., Espinosa-Curiel, I.E., Ríos-Martínez, J.A., Madera-Ramírez, F.A., & Pérez-Espinosa, H. (2023). Machine Learning-Based Sensor Data Fusion for Animal Monitoring: Scoping Review. Sensors, 23(1), 5732.

[28] Zhang, L., Guo, W., Lv, C., Guo, M., Yang, M., Fu, Q., & Liu, X. (2023). Advancements in artificial intelligence technology for improving animal welfare: Current applications and research progress. Animal Research and One Health, 1-17.

[29] M. Oszust and D. Warchoł (2022). Time Series Augmentation with Time-Scale Modifications and Piecewise Aggregate Approximation for Human Action Recognition. 2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI), Macao, China, 2022, pp. 700-704