SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V12I5P116

Volume 12 Issue 5, 176-189, May 2025

Original Article

Enhanced Livestock Monitoring with Modified Kalman Filter and Decision Tree Algorithm for Noise-Reduction in Sensor Data

Arun. C.A[1], Sudhan. M. B[2], Thripthi P Balakrishnan[3], S Gopinath[4], N. Srividhya[5], V. Saravanan[6], S. Navaneethan[7]

[1]*Department of ECE, VelTech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India.*
[2]*Department of CSE, SRM Institute of Science and Technology, Ramapuram, Chennai, India.*
[3]*Department of CSE, Madanapalle Institute of Technology & Science, Madanapalle, Andhra Pradesh, India.*
[4]*Department of Information Technology, Gnanamani College of Technology, Namakkal, Tamil Nadu, India.*
[5]*Department of ECE , K.Ramakrishnan College of Engineering, Samayapuram Trichy, India.*
[6]*Department of ECE, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, India.*
[7]*Department of ECE, Saveetha Engineering College, Chennai, India.*

[1]*Corresponding Author : arunca.me@gmail.com*

Received: 11 March 2025 Revised: 16 April 2025 Accepted: 29 April 2025 Published: 31 May 2025

Abstract - *Kalman filtering, a robust statistical estimation method, has emerged as a pivotal tool across disciplines, excelling in noise reduction and state estimation with minimal computational demands. Its adaptability has fostered diverse implementations, notably in complex sensory and robotic systems. This technique significantly enhances system reliability and efficiency by effectively filtering signals from noise, thereby catalyzing technological progress across numerous sectors. The method is precious in modern systems that rely on high-sensitivity sensors like accelerometers and gyroscopes, which are crucial for improved performance but vulnerable to noise. By addressing the challenge of noisy data readings, which can significantly impact system performance. By strategically deploying sensors on cattle, real-time data on animal movements are collected, and with this, we can predict valuable insights into their daily activities are possible. This innovative research on animal welfare management has been adopted to optimize farm operations. The suggested revised algorithm of the standard Kalman filter can effectively minimize noise in the livestock management system. Different combinations of values for the Q and R variables are tested and tabulated at Q = 0.01 and R = 100, also Q = 0.01, and R = 1000 we got maximum results. Also, we get better results on the proposed modified Kalman filter with the decision tree algorithm, which effectively predicts the actual data with an accuracy of 88.67%, precision of 87.53%, recall of 87.5%, and F1 score of 87.47%, indicating its strong ability to capture the underlying patterns in the dataset. In contrast, the Linear Regression model may be underfitting due to its inability to model such non-linearity effectively. When comparing the results, the decision tree regression method outperformed linear regression and polynomial regression methods. Also, it is well-suited for capturing sudden shifts and plateaus in the cattle's behavior. This innovative project adopted an advanced system that enhances animal welfare management and optimizes farm operations.*

Keywords - *Kalman filter, Regression algorithms, Livestock management system, MPU6050 and ESP12 Microcontroller.*

1. Introduction

Livestock monitoring plays an important role in ensuring the cattle's health continuously and enhancing productivity. In recent years, the development of miniature sensors and wearable devices has created an opportunity to provide a solution to livestock monitoring. These innovations are transforming traditional livestock monitoring practices into data-driven operations. However, the key research gap is designing low-powered, real-time noise-resilient activity recognition for livestock using a modified standard Kalman filter. Most of the existing data collection approaches do not filter out external disturbances, which degrade classification accuracy and reduce the system's reliability. Therefore, to enhance the system's reliability, this work incorporates a modified Standard Kalman Filter to effectively filter out external disturbances and smooth the sensor data before classification. This research proposes a novel, real-time system for cattle activity prediction and behavior analysis using Machine Learning (ML) algorithms.

Figure 1 shows that the sensors are attached directly to the cattle neck to capture their real-time data, including acceleration patterns and overall movements. The real-time data collected from our proposed device under different positions of cattle standing, walking, and grazing. The original data are affected by external disturbances such as temperature fluctuations and wind. It can cause the sensor to drift in gyroscope values and induce mechanical vibrations in the sensor readings. Also, noise in data readings can be added by outer disturbances such as external pressure and vibrations. These sensors are often exposed to such disturbances, which can result in noisy data readings. Since the actual measured data greatly influences the controller's overall performance, noise in data readings can be very harmful. If noise is included in the measurement data, it won't be possible for the control system to give outputs with higher accuracy, or the system may even fail. A central microcontroller acts as the heart of the system, receiving and processing this raw sensor data in real-time. The microcontroller then transforms the data into a structured format and extracts relevant features that best represent the cattle's movements. This processed data is fed into machine learning algorithms. The algorithm is trained on this meticulously prepared dataset, enabling it to learn and identify patterns within the cattle's movement data. Once trained, the model can predict the cattle's current activity, such as standing, walking, or grazing, in real time and analyze the collected information to identify broader behavioral patterns.

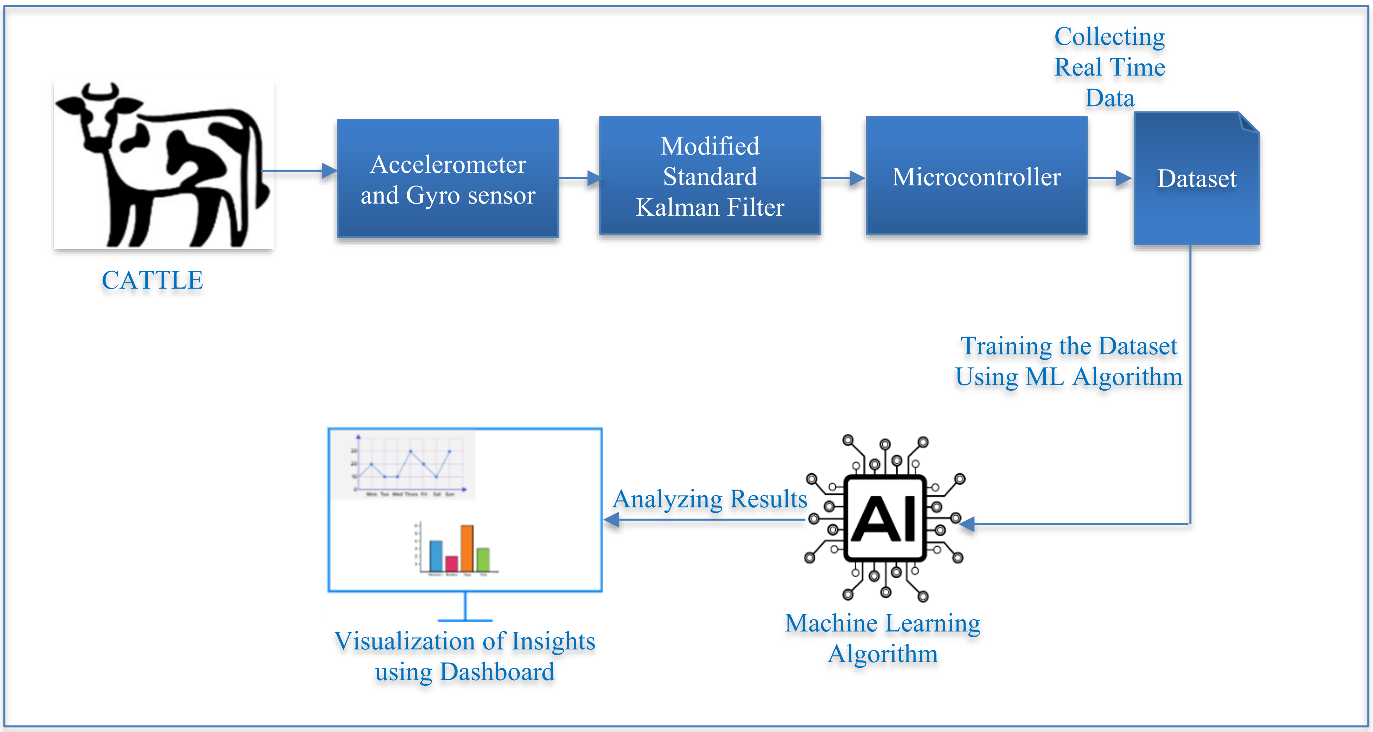

Fig. 1 Block diagram of cattle activity prediction and behavior analysis using ML algorithms

This paper discusses related work in section 2, how real-time data is collected from the proposed livestock monitoring system in section 3, the original equations and modifications done to minimize or filter the noise from external disturbances, and significant divergences in sensor data are addressed in sections 4 and 5, accordingly. The outcomes and conclusions are presented in sections 6 and 7 correspondingly.

2. Related Works

Kalman filtering, a method for state estimation introduced by Rudolf E. Kalman [1] in the year 1960, has gained popularity across a variety of industries because of its low processing resource requirements and capacity to eliminate undesired noise from data containing noisy measurements. Over the past two decades, numerous variations and applications of the Kalman filter have been developed. Its applications include satellite navigation, robotics, signal processing, wireless sensor networks, automobiles, military, biomedical, and economic systems. The Kalman filter plays a vital role in robotics, contributing to localization, position estimation, trajectory tracking, parameter identification, and the control of mobile robots. High-sensitivity sensors are critical components of contemporary systems in various industries like the electronics sector, health and medical sector, automobiles, automation and control systems, and environmental monitoring. By perceiving minute changes and reacting more accurately to changing circumstances, they improve accuracy and performance. Accurate monitoring of vital signs is made possible in the healthcare industry; obstacle and pedestrian detection is improved in automotive systems; manufacturing efficiency is increased in industrial settings; accurate pollutant measurements are provided in environmental monitoring; and user experiences are improved in consumer electronics. When reading data from the environment, these highly sensitive sensors are very susceptible to noise, which may compromise their accuracy and dependability. Kalman filtering is a widely recognized algorithm that delivers approximations of certain unknown future variables by utilizing the measurements perceived over a specific time. It operates recursively, and it is particularly useful for systems that consist of noise and uncertainty.

To evaluate the condition of a linear system from a sequence of observations affected by noise, a recursive process is employed in typical Kalman filters. The standard Kalman filter is tailored for such environments that involve Gaussian noise. It is used if the system needs a simple, efficient, and reliable solution. Additional Kalman filter types include the Ensemble, Unscented, and Extended Kalman Filter.

Kalman filter was chosen for noise reductions based on a thorough evaluation of several filtering methods, which is compared in Table 1. These are designed to handle different complexities in systems, such as nonlinearity or non-Gaussian noise. Some of them are used in large-scale systems like weather forecasting and industrial automation. Kalman filtering is essential for improving the reliability and effectiveness of numerous technological applications.

Urrea, Claudio, and Rayko Agramonte et al. [2] compare the different types of Kalman filters used in robotics. With equivalent computational complexity for general state-space problems, the Unscented Kalman filter exceeds the performance of the Extended Kalman filter in terms of prediction and error estimation. S.A. Quadri et al. [3] collected 1000 samples from a gyroscope and accelerometer to ascertain the measurement and process noise from the SparkFun Inertial Measurement Unit (IMU). After this, the estimation errors due to measurement and process noise were calculated and compared. The proposed measurement noise is inversely proportional to the gain of the Kalman filter and it may be the reason for the increment of estimation error.

Alfian Ma'arif et al. [17, 19] proposed a modified Standard Kalman filter for noise reduction in sensor data under a noisy environment. Measurement variance and process variance constant values are obtained by using MATLAB and tests with IMU. The ratio of constants is tested on different values to reduce the damping value. Winursito, Anggun, et al. [18] introduced a technique to minimize the error in sensor readings, especially in outdoor environments such as agriculture applications. For ignoring negative values root mean squared error value is obtained and compared with and without Kalman filter sensor values. Q. -Q. Qian et al. [20] outlined a method to enhance the accuracy of the frequency detection method based Kalman filter with disturbance estimation. Using disturbance estimation, they isolated the frequency disturbances caused by power fluctuations from the noise. Jwo, Dah-Jing, and Amita Biswal et al. [21] explained how the Kalman filter works without requiring basic knowledge regarding comprehensive theoretical knowledge of probability and stochastic processes.

Wang S et al. [26] explore machine-learning-based human motion recognition using a wearable plastic-fiber sensing system. The research investigates the potential of wearable sensors in capturing human motion data and employs machine learning techniques for motion recognition. The findings highlight the efficiency of wearable sensor systems coupled with machine learning algorithms in accurately identifying various human movements, with implications for applications such as healthcare, fitness tracking, and gesture control interfaces.

Table 1. Comparison of Kalman filter with other filters in a noisy environment

Technique	Description	Limitations in Noisy Environments
Moving Average Filter [16-19]	Smooth's data by averaging over a window of previous samples.	Tends to lag behind rapid signal changes; not adaptive to dynamic noise characteristics.
Low-Pass Filter (LPF) [16-19]	Attenuates high-frequency noise while preserving low-frequency signals.	It can distort fast-changing signals; fixed cut-off frequency may not handle variable noise well.
Median Filter [16-19]	Replaces each point with the median of neighboring values; good for removing outliers.	Effective only for impulsive noise (spikes), not for Gaussian or continuous noise; computationally heavier.
Savitzky-Golay Filter [20-21]	Smooth's data while preserving features like peaks by fitting local polynomials.	Assumes consistent noise pattern; not optimal for real-time, highly dynamic systems.
Standard Kalman Filter [20-21]	Recursive estimator predicting current state based on prior estimates and measurements.	Requires a good model of system dynamics and noise characteristics; slightly complex to implement.

H. Myung et al. [27], the conventional Kalman filtering method that fuses the odometry with the gyroscope, may produce infeasible solutions because the robot parameters are estimated regardless of their physical constraints. The authors discussed the constrained Kalman filtering method that applies a general constrained optimization technique to the estimation of the robot parameters. The additional state variables improve the state observability, and the accuracy is also improved by the non-approximated Kalman filter design. Experimental results show the proposed method effectively compensates for the odometry error and yields feasible parameter estimation at the same time.

3. Design and Implementation of Proposed Livestock Monitoring System

Livestock monitoring has undergone a transformative shift with the integration of IoT and ML technologies. This innovative idea presents an advanced system that enhances animal welfare management and optimizes farm operations. By strategically deploying accelerometer and gyro sensors on cattle, real-time data on animal movements and rotations are collected, capturingvaluable insights into their daily activities [4-6]. The proposed livestock monitoring project utilized the MPU6050 [9], accelerometer, and gyroscope sensor, integrating a 3-axis accelerometer and a 3-axis gyroscope, offering precise motion sensing capabilities [7-10]. This data is essential for monitoring cattle health and detecting any irregularities or signs of distress [11-14]. Additionally, the 3-axis gyroscope measures angular velocity, enabling the identification of rotational movements, including turning and head movements. The following figure 2 shows the Schematic diagram of the data acquisition system of the proposed model.

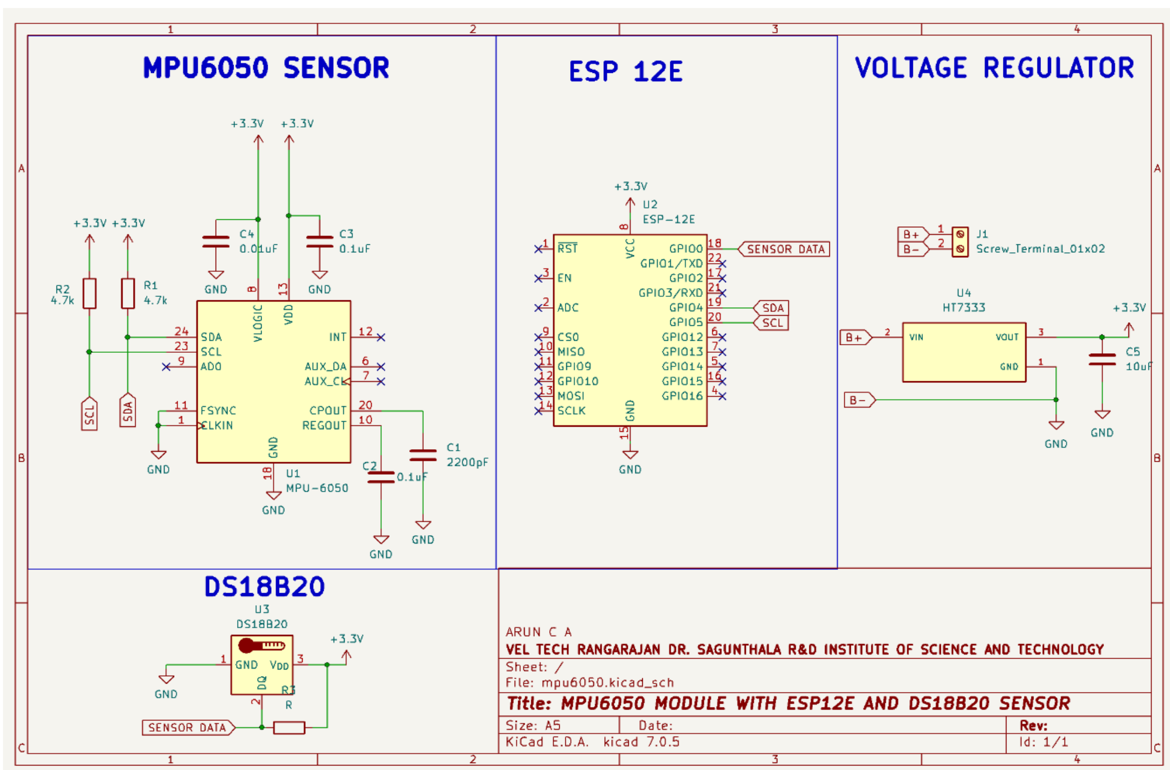

Fig. 2 Schematic diagram of Data acquisition system

The MPU6050 sensor typically communicates over the Inter-Integrated Circuit (I2C) bus, so ensure that the Serial Data (SDA) and Serial Clock (SCL) pins of the MPU6050 are connected to the corresponding pins on the microcontroller. Initialize communication with the MPU6050 sensor and configure the sensor's settings, including the gyroscope range (e.g., ±250, ±500, ±1000, or ±2000 degrees per second) and the accelerometer range (e.g., ±2g, ±4g, ±8g, or ±16g). This can be done by writing specific configuration values to the sensor's registers [10]. Reading raw sensor measurements from specific registers on the MPU6050 using I2C communication. Convert the raw sensor data into degrees per second for gyroscope data based on the sensor's configuration settings. Store the acquired sensor data in memory or transmit it to an external device (e.g., a computer or microcontroller) for further processing and analysis. This small and battery-powered device is fixed on the neck of the cattle to collect the data. The combination of MPU6050 and ESP12E microcontroller [15] facilitates seamless data transmission and processing, empowering farmers and researchers with actionable data to optimize cattle management practices, enhance productivity, and promote animal welfare.

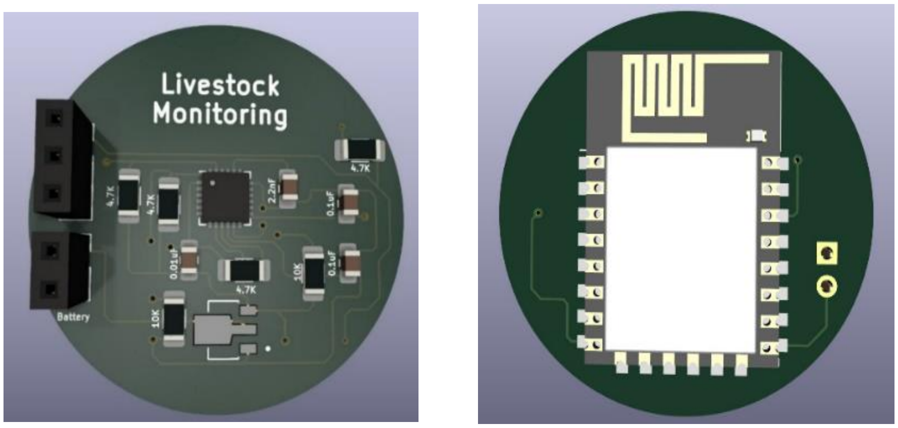

Fig. 3 Front and Rear view of Data Acquisition System (DAS)

Fig. 4 Real-time Data Acquisition System (DAS)

The subsequent Figures 3 and 4 depict the front and rear 3D layout and Real-Time visualization of the Data Acquisition System (DAS). A variety of tactics must be used in tandem to manage and reduce noise in sensor data. Radiofrequency interference and electromagnetic interference can be minimized by grounding and insulating the sensor system properly. Data can be smoothed using software filters (Kalman filters, moving average) and hardware filters (LPF, HPF, BPF, and BRF) [15-16]. Common-mode noise can possibly be reduced by putting sensors far from noisy areas and by utilizing differential sensors. Effective ways to reduce mechanical noise in sensors include using dampers to isolate them from noise, applying machine learning models to differentiate between signals and noise, and using sophisticated signal processing techniques like Fourier transforms and wavelet transforms. Additional noise reduction can be achieved by utilizing statistical approaches for outlier detection, numerous sensors, and sensor fusion techniques [17-18].

4. Standard Kalman Filter Noise Reducer

It is an iterative mathematical procedure that lowers the mean squared error by using a set of equations to forecast the state of the system and revising predictions derived from differences between expected and observed values. There are two primary processes involved: as prediction process and the update process. A detailed explanation of these processes is given below.

4.1. Prediction Process

The prediction process is a fundamental process in estimating the upcoming state of a dynamic system drawn from its current condition and known dynamics. This is achieved in two steps: State Prediction (1) and Error Covariance Prediction (2).

4.1.1. State Prediction

$$\hat{x}_{k/k-1} = F\hat{x}_{k-1/k-1} + Bu_k \qquad (1)$$

$\hat{x}_{k/k-1}$ = predicted state estimate at time k using the state at k-1
F=state transition model

$\hat{x}_{k-1/k-1}$ = state estimate at time k-1
B=control input model which applies to control vector
u_k= control vector

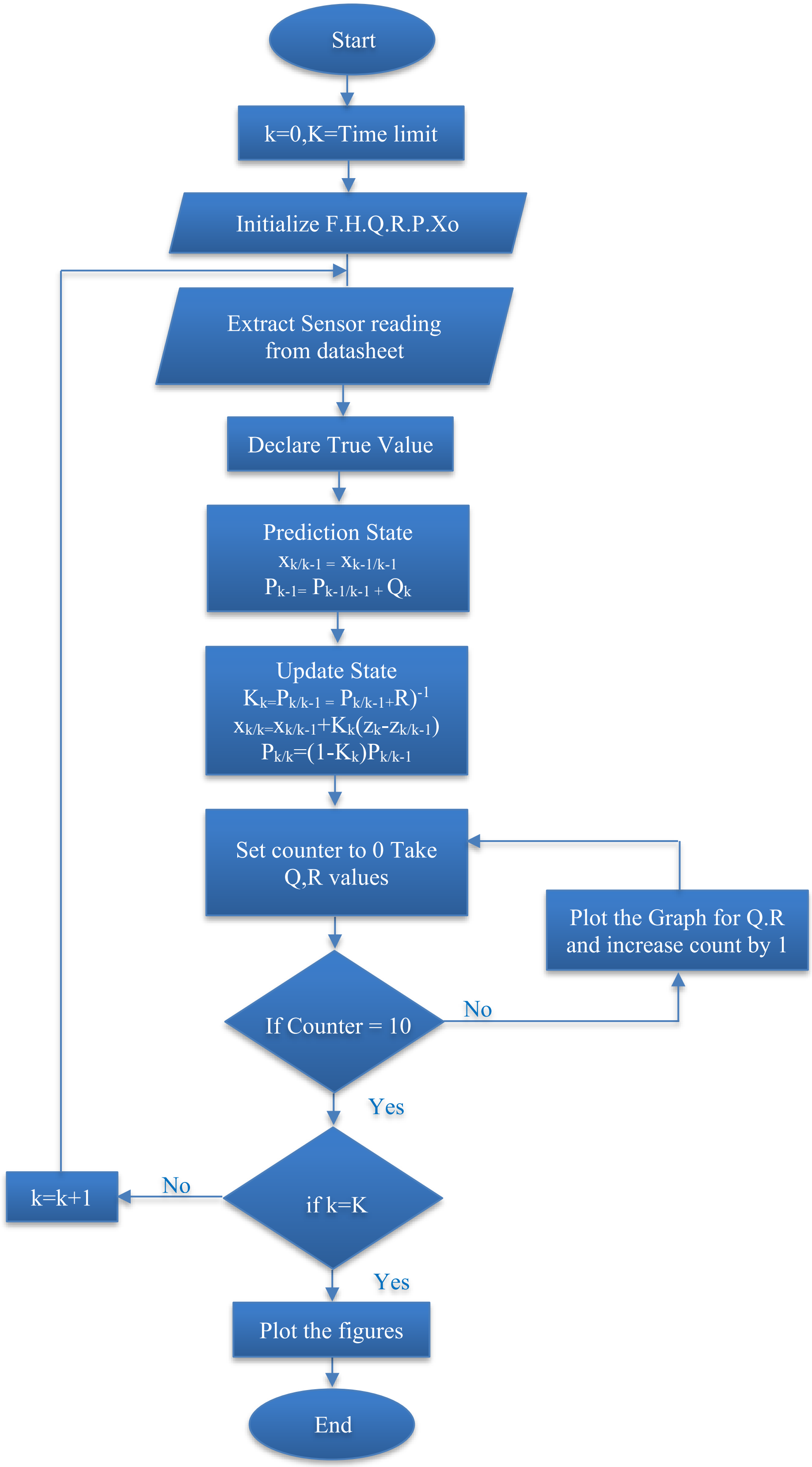

Fig. 4 Proposed Modified Standard Kalman Filter Flow Chart

4.1.2. Error Covariance Prediction

$$P_{k/k-1} = FP_{k-1/k-1}F^T + Q \tag{2}$$

$P_{k/k-1}$=predicted error covariance matrix at time k
$P_{k-1/k-1}$=error covariance matrix at time k-1
Q=process noise covariance matrix

4.2. Update Process

In the update process, new measurements are integrated to find the accurate state estimates made during the prediction step. This step is crucial for adjusting the state estimate based on the most recent observations, in such a way it improves the accuracy of the Kalman filter's predictions. It has three steps: Kalman gain Calculation (3), State Update (4), and Error Covariance update (5).

4.2.1. Kalman Gain Calculation

$$K_k = P_{k/k-1}H^T\left(HP_{k/k-1}H^T + R\right)^{-1} \tag{3}$$

K_k= Kalman Gain
H=Observation Model
R= Measurement Noise covariance Matrix

4.2.2. State Update

$$x^\wedge{}_{k/k} = x^\wedge{}_{k/k-1} + K_k\left(z_k - H_k x^\wedge{}_{k/k-1}\right) \tag{4}$$

$x^\wedge{}_{k/k}$ = Updated State Estimate at time k
z_k= measurement at time k
$H_k x^\wedge{}_{k/k-1}$= Predicted Measurement

4.2.3. Error Covariance Update

$$P_{k/k} = (I-K_kH)P_{k/k-1} \tag{5}$$

$P_{k/k}$=Updated error covariance matrix at time k
I= Identity Matrix

5. Proposed Modified Standard Kalman filter Algorithm

The procedure for applying a standard Kalman filter to a dataset with sensor readings involves several key steps. First, initialization involves setting up the necessary matrices and parameters for the Kalman filter, together with the state transition matrix (F), process noise covariance matrix (Q), observation matrix (H), measurement noise covariance matrix (R), initial error covariance matrix (P), and the initial state vector (Xo). Some of these parameters can be eliminated as we are using the one-dimensional system, but for better understanding, we are assigning 0 or 1 to those parameters based on their respective position in the equations from (1) and (5) for better understanding.

Following that, the Kalman filter is executed in an iterative manner using different sets of (Q) and (R) values to simulate varying levels of process and measurement noise. For each combination of (Q) and (R), a new instance of the Kalman filter is initialized with defined parameters. It iterates through each observation in the dataset, predicting the next state using the prediction method to project forward based on system dynamics and process noise.

The update method incorporates each observation to refine the state estimate, adjusting for measurement noise and updating the error covariance matrix (P). Finally, our algorithm evaluates the performance of the filter by calculating the mean absolute error between the observed data and predicted positions. It generates plots for each (Q) and (R) pair, visually comparing the observed data, estimated positions from the Kalman filter, and the true value if available. These plots provide insights into how well the Kalman filter mitigates noise and tracks the underlying state over time, aiding in understanding its effectiveness under different noise conditions and informing decisions in various application contexts.

Figure 4 depicts a flow diagram of the proposed Modified Standard Kalman Filter. These modifications are done by the process of calculating the results rather than updating the equations, which is a traditional method. The program is designed in such a way it removes the unwanted parameters and calculates the results.

6. Results and Discussion

The results are taken under different values of R and Q. In that, we have taken real-time data captured from our proposed model with noise as a reference. The Kalman filter is implemented on our real-time data and subjected to validation. In Figure 5, both the variance constant (Q) also the measurement constant (R) are set to 1.0. This suggests that the filter assumes an equal amount of noise in both the process and the measurements. It is just reducing the intensity of noise by reducing the length of the noise spike. Furthermore, there is a wide range of values, which would produce much more stable results in the system.

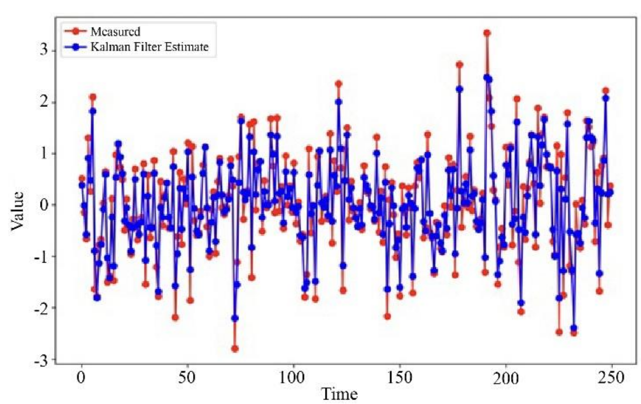

Fig. 5 Q=1.0, R=1 and Mean Error=22.6647

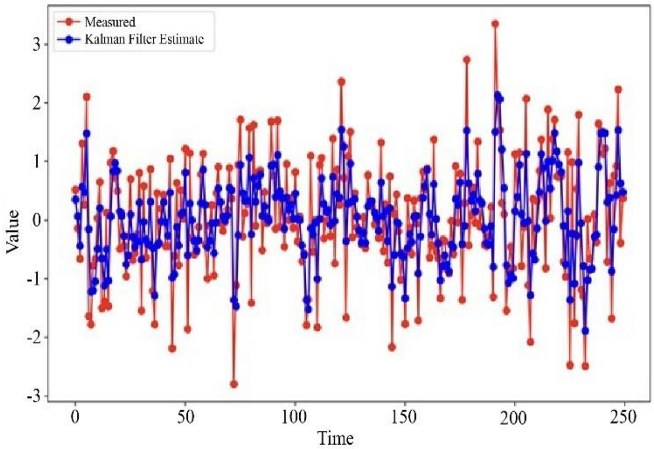

Fig. 6 Q=0.1, R=1 and Mean Error=44.9598

Here, the variance constant (Q) is reduced to 0.1 while the measurement constant (R) remains at 1.0. With a lower variance constant, the filter gives more weight to the process model than to the measured values.

Consequently, the estimates become smoother compared to the measurements. Here, in Figure 6 the measured signals are filtered more effectively when compared with the previous one.

In Figure 7, Q is further reduced to 0.01. The measurement constant (R) is still 1.0. The filter now heavily relies on the process model, resulting in very smooth estimates. The system is giving preference for a smoother prediction rather than closely following noisy observations. Yet the system would give aggressive outputs.

The fourth and fifth inspections are correspondingly shown in Figure 8 using $R = 1$ and $Q = 0.001$ and in Figure 9 using $R = 1$ and $Q = 0.0001$. These configurations demonstrate the effect of nearly complete reliance on the process model, producing very stable but potentially outdated estimates.

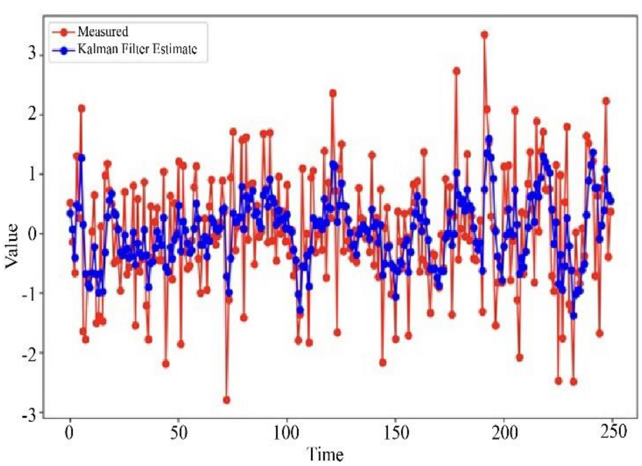

Fig. 7 Q=0.01, R=1 and Mean Error=58.5228

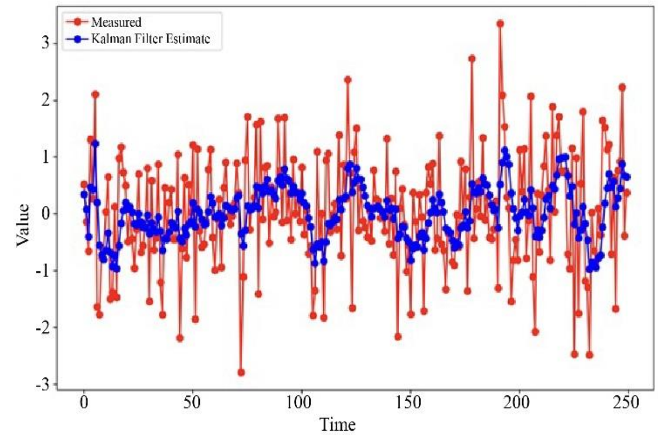

Fig. 8 Q=0.001, R=1 and Mean Error=67.5613

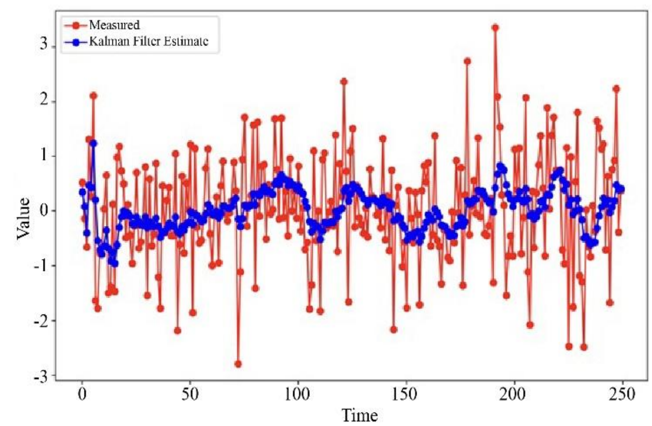

Fig. 9 Q=0.0001, R=1 and Mean Error=72.0613

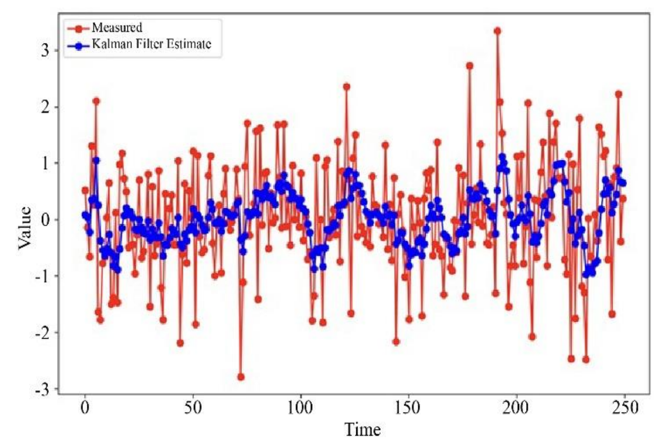

Fig. 10 Q=0.01, R=10 and Mean Error=67.9314

These two analyses are nearly the best as, in two of the filter results, noise can be decreased while maintaining the original properties of the data. To get better results than this, let's increase the R-value by keeping the Q value constant. Here, in Figure 10, the variance constant (Q) is 0.01, and the measurement noise covariance (R) is increased to 10.0. The high R-value indicates that the filter assumes significant noise in the measurements. The resulting estimates should be smooth and less influenced by the noisy observations.

But in Figure 11, we can clearly observe the estimated values being influenced by the noisy observations. In this configuration, Q remains at 0.01, but R is further increased to 100.0. The very high measurement noise covariance suggests that the filter assumes a substantial amount of noise in the measurements, heavily relying on the process model. The estimates will be very smooth and largely ignore the noisy measurements. In Figure 11, we can clearly observe the estimated values, ignoring the noisy observations. This looks to be the best among the examinations made till now.

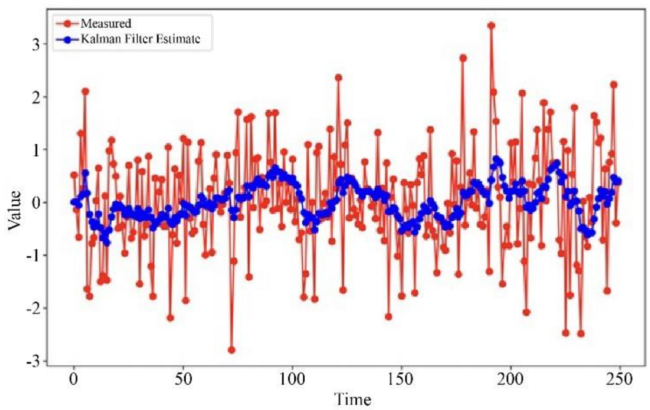

Fig. 11 Q=0.01, R=100 and Mean Error=73.0040

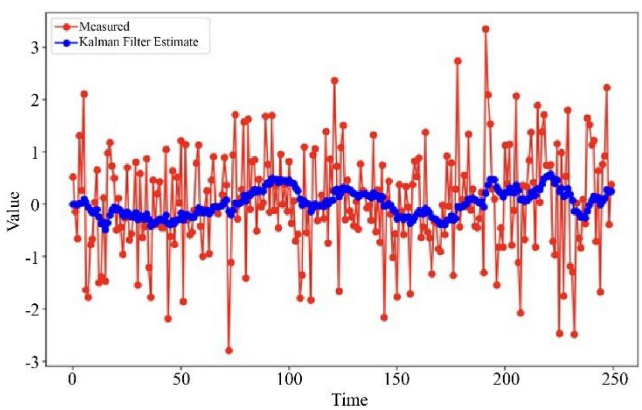

Fig. 12 Q=0.01, R=1000 and Mean Error=75.4286

In the following Figure 12, Q is 0.01, and R is set to an extremely high value of 1000.0. The filter now assumes an excessive amount of noise in the measurements, leading to almost completely depending on the process model. The estimates are very smooth and nearly unaffected by the measurements. This cannot be preferred as the characteristics of the graph are started to be ignored**.**

Figures 13 and 14 display the results of the ninth and tenth examinations, respectively, using $R = 10000$ and $Q = 0.01$ and $R = 100000$ and 0.01 correspondingly. In both these examinations, the filter completely ignores the measurements as they are assuming the measurements to be extremely noisy. The estimates will follow the process model with minimum influence of the observations. It can be observed that the characteristics of the graph completely vanished in both of these examinations. So, these values shouldn't be considered.

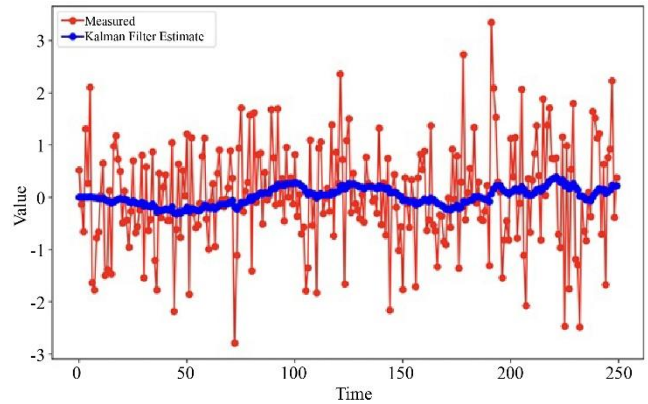

Fig. 13 Q=0.01, R=10000 and Mean Error=77.0596

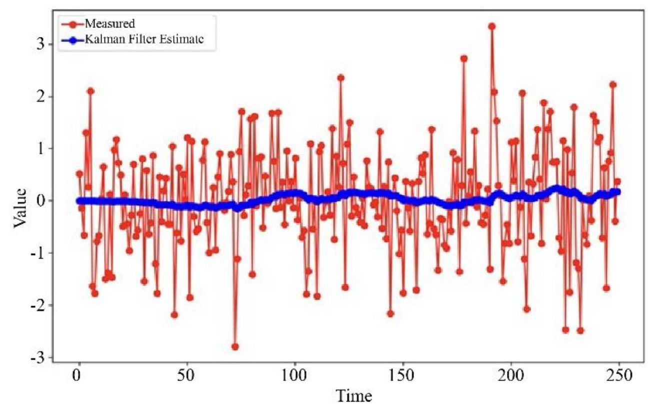

Fig. 14 Q=0.01, R=100000 and Mean Error=77.9718

These parameters have a significant impact on the noise reducer, according to the analysis from Figure 5 to Figure 14. The noise-dampening effect increases with increasing the R/Q ratio. After that, the filter result is contrasted with the actual values to look into the damping effect. Subsequently, the difference can be expressed as an error equation, presented below:

$$\bar{e} = \left| x_n - y_n \right| \tag{6}$$

The mean error is expressed as $\bar{e}$, in which y represents the filter results and x represents the data values. Table 2 shows that larger differences between R and Q result in larger mean error values. Furthermore, regardless of the values of y and z, the same y difference values yield a comparable number for mean error. The ideal parameters that yield a result with the original data features have mean error values from 70 to 75 when Table 2 is compared to earlier analysis from the figures. The mean Error rate is the average difference between the actual and estimated values multiplied times a hundred. The method we followed for deciding Q and R values is the trial and error method. Suppose we have an idea of how our result should look in the graph. We can adjust the Q and R values correspondingly. If we think there are high noises in the graph, we can increase the R-value and reduce the noise. This can be done until we achieve the best results. Suppose the model seems to be performing well and our goal is for the filter to place greater emphasis on the model predictions. In that case, we can decrease the Q value by iteratively adjusting Q and R and observing the results in the graph. Also, we derived the relationship between mean error with R and Q values, which is shown in Equation (7).

$$\text{Mean Error} = 70.9136 - 0.2973*R - 308.1559*Q - 0.0000*R^2 + 29.8362*R*Q + 230.3669*Q^2 \quad (7)$$

In data preparation, the different iteration data is stored in a pandas DataFrame and then used to train the models. The ratio R/Q is transformed into its logarithmic form, which helps improve model accuracy for certain types of regression, particularly when dealing with wide-range values or exponential trends. Once the models are trained, their predictions are classified into cattle activities. The binning function assigns each predicted value into one of the following categories based on its value. This classification helps in determining the specific activity or behavior of the cattle, e.g., grazing, resting, walking, etc. When modeling this behavior, polynomial regression of degree 3 and decision tree regression provide a much more realistic representation compared to linear regression. Among these, decision tree regression stands out for its ability to closely follow the actual data pattern, demonstrating its strength in capturing abrupt changes and plateaus in system behavior.

Figure 15 compares three distinct regression models, linear regression, polynomial regression, and decision tree regression, using various R and Q values along with the mean error value. Compared to the other two algorithms decision tree algorithms gave better results, which are very close to the actual data.

Table 2. Mean Error Value for different R and Q Values

Iteration No	R-Value	Q Value	R and Q Ratio	Mean Error Value Of Existing Kalman Filter [17]	Mean Error Value Of Proposed Modified Kalman Filter
1	1.0	1.0	1	26.0677	22.6647
2	1.0	0.1	10	44.7392	43.9598
3	1.0	0.01	100	53.4966	51.5228
4	1.0	0.001	1000	69.4621	67.5613
5	1.0	0.0001	10000	76.3324	72.0613
6	10.0	0.001	1000	70.4755	67.9314
7	100.0	0.01	10000	76.2589	73.0040
8	1000.0	0.01	100000	79.3356	75.4286
9	10000.0	0.01	1000000	82.3349	77.0596
10	100000.0	0.01	10000000	82.9657	77.9718

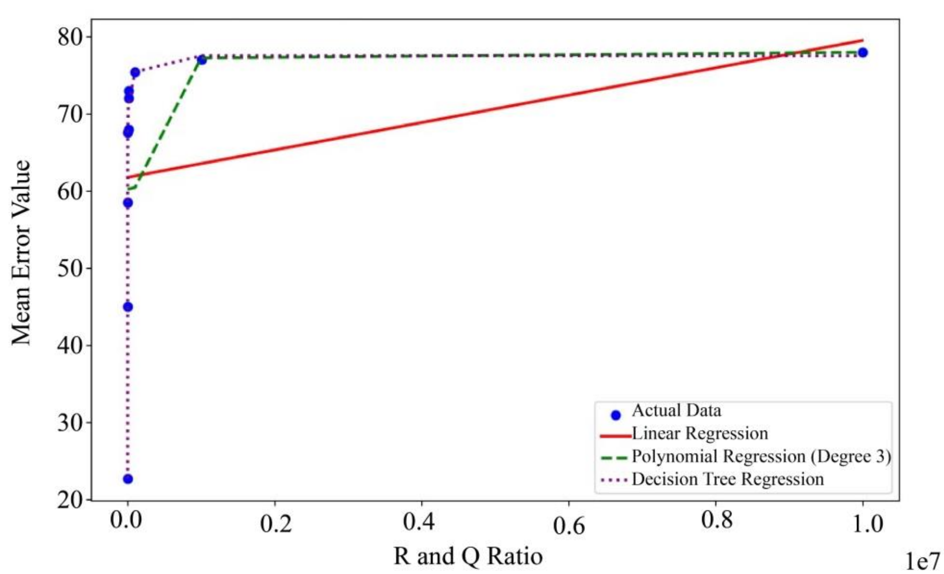

Fig. 15 Comparison of Regression models for different R and Q ratios with Mean Error value

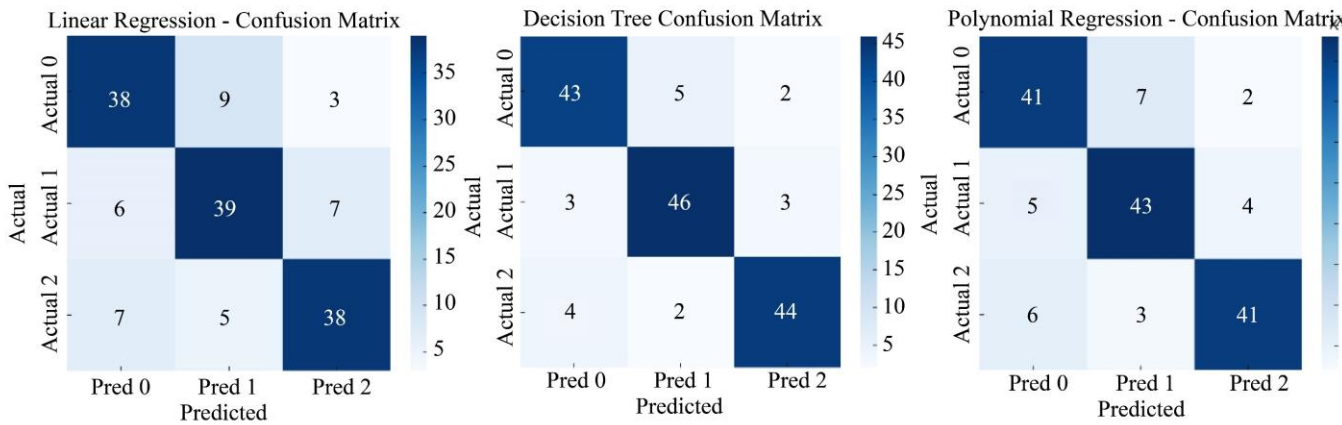

Fig. 16 Confusion matrix of regression models

The effectiveness of the Machine Learning models used in this work is evaluated with standard classification matrices such as Accuracy, Precision, Recall, and F1 Score. These were calculated using multi-class confusion matrices and macro-averaging. The confusion matrices for each model illustrate the model's ability to correctly predict each class, such as grazing, resting, and walking of cattle. The confusion matrix of linear regression, decision tree, and polynomial regression is shown in Figure 16. The Decision Tree algorithm is a widely used supervised machine learning technique applicable to both classification and regression tasks. It operates by recursively splitting the dataset based on feature values to form a tree-like structure, where internal nodes represent feature tests, branches represent the outcomes of these tests, and leaf nodes denote the final prediction or class label. The process starts at the root node and follows a path based on decision rules until a conclusion is reached at a leaf. Decision Trees are easy to understand and interpret, do not require feature scaling, and can handle both numerical and categorical variables. However, they are prone to overfitting, especially when the tree grows too deep, and can be sensitive to small variations in the data. Techniques like pruning and setting depth limits are used to overcome these issues. Despite their limitations, Decision Trees serve as the foundation for powerful ensemble methods like Random Forest and gradient-boosted trees. Decision Tree Regression was used to predict cattle behavior based on motion sensor data filtered. After denoising the dataset via the modified standard Kalman filter, the ratio R/Q and mean value error were calculated using eqn (7). This feature is used to train the decision tree model to identify the mean value error for a new set of values in real time. The comparison between all algorithms is shown in Figure 17, which shows that the accuracy of the decision tree is 88.67% and the F1 score is 87.47%, comparatively better than the other two algorithms. The following equations show how the accuracy, precision, recall, and F1 score are calculated.

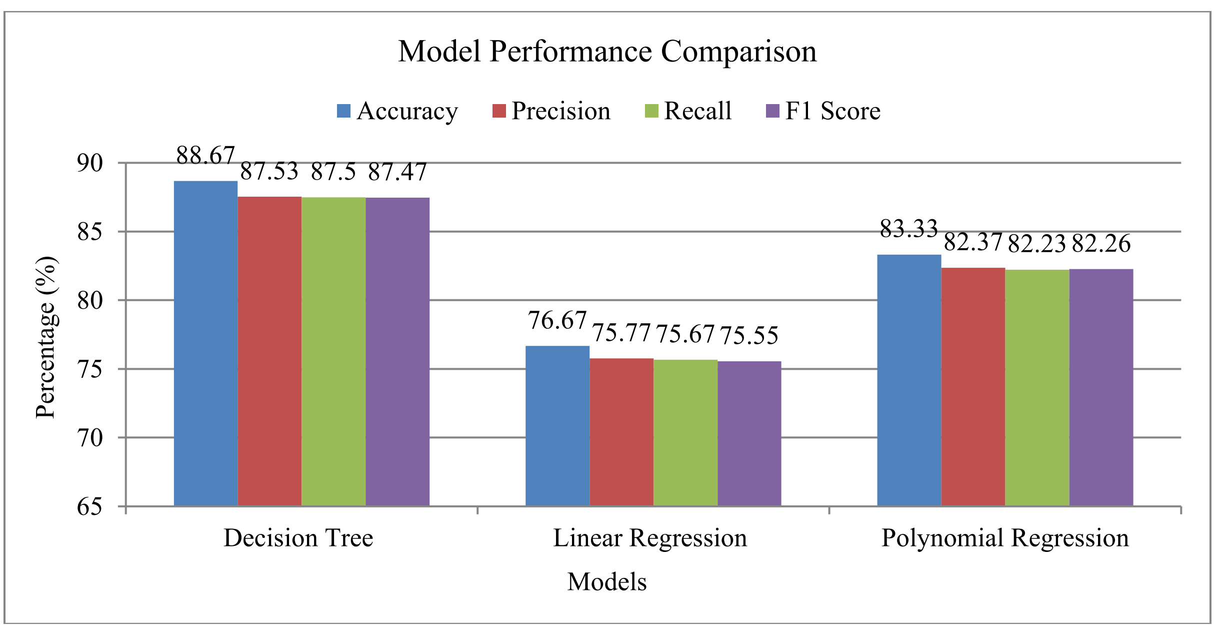

Fig. 17 Performance Comparison of Different ML models

$$Accuracy = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N}\left(TP_i + FP_i + FN_i + TN_i\right)} = \left(\frac{(43+46+44)}{150}\right) \approx 0.8867$$

$$\Pr ecision = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{\left(TP_i + FP_i\right)} = \frac{\left[\frac{43}{(43+7)}+\frac{46}{(46+7)}+\frac{44}{(44+5)}\right]}{3} \approx 0.8753$$

$$\operatorname{Re} call = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{\left(TP_i + FN_i\right)} = \frac{\left[\frac{43}{(43+7)}+\frac{46}{(46+6)}+\frac{44}{(44+6)}\right]}{3} \approx 0.875$$

$$F1Score = \frac{1}{N}\sum_{i=1}^{N}\frac{\left(2\times \Pr ecision_i \times \operatorname{Re} call_i\right)}{\left(\Pr ecision_i + \operatorname{Re} call_i\right)} \approx 0.8747$$

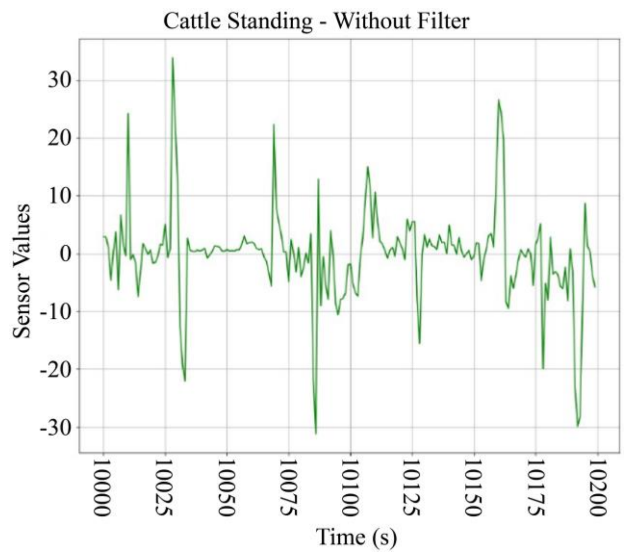

Fig. 18 Cattle Standing Data Without Kalman Filter

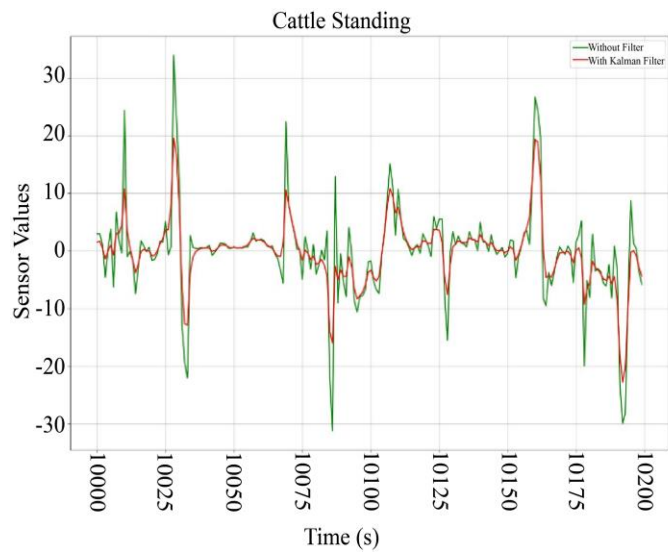

Fig. 19 Cattle Standing Data With Kalman Filter

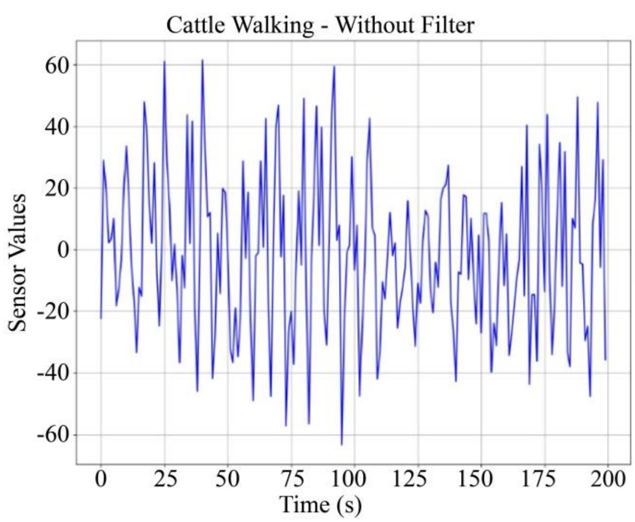

Fig. 20 Cattle Walking Data Without Kalman Filter

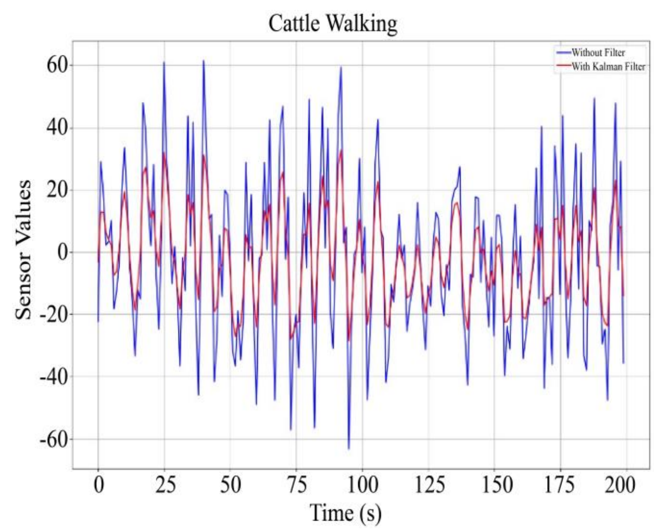

Fig. 21 Cattle Walking Data With Kalman Filter

In this research, three machine learning algorithms were used to model the performance trend of the Kalman filter with varying R/Q ratios. Linear regression, being the simplest, attempted to fit a straight line to the data but failed to capture the non-linear behavior effectively. Polynomial regression of degree 3 improved upon this by introducing curvature, which allowed it to better represent the saturation effect observed at higher R/Q values. Decision tree regression outperformed both, as it is well-suited for modeling complex, non-linear patterns and was particularly effective in capturing sudden shifts and plateaus in the system's behavior. Figures 18-23 show the real-time.

Data was collected from our proposed device under different positions of cattle such as standing, walking, and grazing. From that, we can notice that the original data are affected by external disturbances. We applied our proposed modified Kalman filter equipped with a decision tree algorithm on those noisy data. From the following figures, we observe that our proposed modified Kalman filter equipped with a decision tree algorithm effectively predicts the data. One of the key advantages of this filter is its minimal computational demand, straightforward implementation, rapid convergence, and dependability.

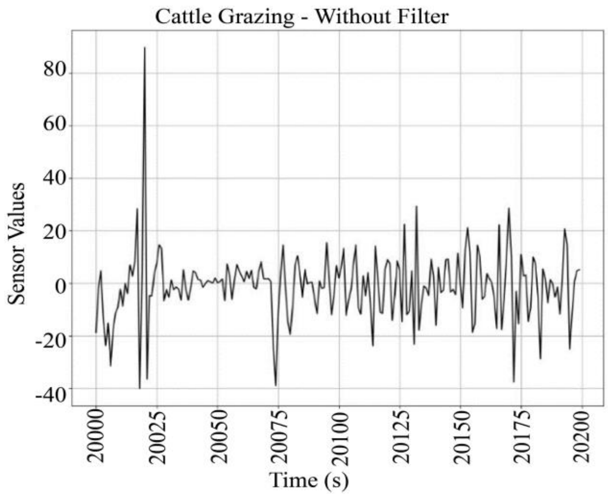

Fig. 22 Cattle Grazing Data Without Kalman Filter

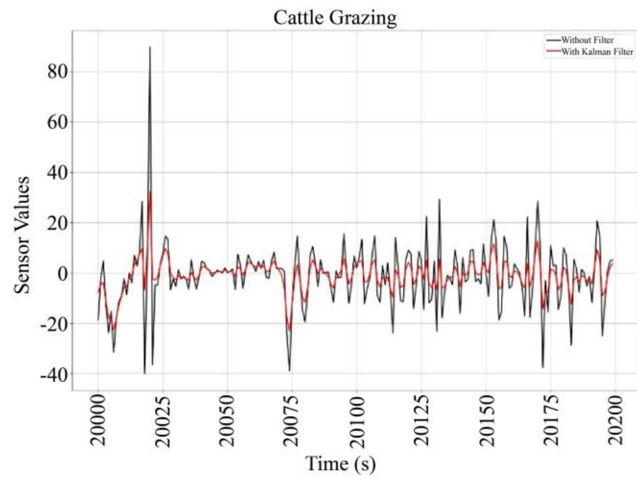

Fig. 23 Cattle Grazing Data With Kalman Filter

7. Conclusion

The research that has been done explains the proposed modified algorithm of the standard Kalman filter and tests its ability to reduce noise in the livestock management system. The MPU6050 uses the modified Kalman filter technique with a decision tree machine learning algorithm to minimize or eliminate noise at the final output, and this process is reliant on the combinational value of the Q and R variables. Theoretical analysis and component equation design will be done for the Kalman Filter equations. When Q > R in the combinational value of Q and R, the Kalman filter could not function properly; as a result, the modified results and the initial reading from the sensor output appeared to be identical. When Q = R in the combinational value of Q and R, the algorithm still failed, as the damping results began to deviate from the original data from the sensor, though the magnitude was still relatively small. When Q < R in the combinational value of Q and R is changed in such a way, the algorithm performs well, and the damping results deviate from the output of the sensor's original value. We obtained the best results at Q = 0.01, R = 100, and Q = 0.01, R = 1000. Attenuation rose along with a larger disparity between variables Q and R. A very substantial attenuation, where the original data from the sensor was deleted, was observed when the discrepancy of the two parameters (Q and R) was quite great. Finally, the proposed modified standard Kalman filter is equipped with a decision tree algorithm on those noisy data. Linear Regression and Polynomial Regression showed significantly lower performance in terms of accuracy and F1 score when compared with decision tree regression. The decision tree model achieved perfect results, with 88.67% accuracy, precision 87.53%, recall 87.5%, and 87.47% F1 score, indicating its strong ability to capture the underlying patterns in the dataset. This suggests that the data likely contains complex non-linear relationships that the Decision Tree model is well suited to handle. In contrast, the Linear Regression model may be underfitting due to its inability to model such non-linearity effectively. This innovative idea adopted an advanced system that enhances animal welfare management and optimizes farm operations. By strategically deploying sensors on cattle, real-time data on animal movements and rotations are collected, capturing valuable insights into their daily activities are possible. By effectively filtering signals from noise, this technique significantly enhances system reliability and efficiency in our livestock management system. Future work is to investigate adaptive filtering techniques that can dynamically adjust the Q and R values in real time based on changing sensor conditions or environmental factors. Another area that needs to be addressed is the integration of more advanced machine learning models, such as ensemble learning methods or lightweight deep learning algorithms, which could further enhance noise reduction and signal accuracy.

References

[1] Rudolph Emil Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35-45, 1960. [CrossRef] [Google Scholar] [Publisher Link]

[2] Claudio Urrea, and Rayko Agramonte, "Kalman Filter: Historical Overview and Review of Its Use in Robotics 60 Years after Its Creation," *Journal of Sensors*, vol. 2021, no. 1, pp. 1-21, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] S.A. Quadri, and Othman Sidek, "Error and Noise Analysis in an IMU using Kalman Filter," *International Journal of Hybrid Information Technology*, vol. 7, no. 3, pp. 39-48, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[4] Xianliang Chen et al., "Kalman Filter and Neural Network Fusion for Fault Detection and Recovery in Satellite Attitude Estimation," *Acta Astronautica*, vol. 217, pp. 48-61, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Kunal Kunal et al., "Accelerometer Implementation as Feedback on 5 Degree of Freedom Arm Robot," *Journal of Robotics and Control (JRC)*, vol. 1, no. 1, pp. 31-34, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[6] Xie Pengfei, Chu Shiwen, and Yan Zhang, "Design of Pose Measurement and Display System Based on STM32 and MPU6050," *IEEE International Conference on Intelligent Computing, Automation and Systems*, Chongqing, China, pp. 71-74, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] K. Kalaiarasi et al., "Advanced Deep Learning Approaches for Enhanced Satellite Image Analysis and Comprehensive Land Use Classification in Diverse Environmental Contexts," *IEEE International Conference on Cybernation and Computation (CYBERCOM)*, Dehradun, India, pp. 603-608, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] G.R. Kanagachidambaresan et al., "Harnessing Machine Learning Based Sequence Prediction for Industry 4.0 Applications," *IEEE 15th International Conference on Computing Communication and Networking Technologies*, Kamand, India, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] C. Aquilani et al., "Review: Precision Livestock Farming Technologies in Pasture-Based Livestock Systems," *Animal*, vol. 16, no. 1, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] E.S. Fogarty et al., "Can Accelerometer Ear Tags Identify Behavioural Changes in Sheep Associated with Parturition?," *Animal Reproduction Science*, vol. 216, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] S. Maheswari, and E. Malarvizhi, "IoT-Powered Real-Time Soldier Retrieval and Surveillance System for Intervention in Critical Combat Environments," *Communications on Applied Nonlinear Analysis*, vol. 32, no. 9s, 2025. [CrossRef] [Publisher Link]

[12] Kristina Dineva, and Tatiana Atanasova, "Health Status Classification for Cows Using Machine Learning and Data Management on AWS Cloud," *Animals*, vol. 13, no. 20, pp. 1-28, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] A. Suganya et al., "Innovative System for Environmental Monitoring with Connected Sensors and Data Analysis," *IEEE International Conference on Intelligent Systems and Advanced Applications*, Pune, India, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Reza Arablouei et al., "In-Situ Classification of Cattle Behavior using Accelerometry Data," *Computers and Electronics in Agriculture*, vol. 183, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] ESP-12F Datasheet, Ai-Thinker, 2018. [Online]. Available: https://docs.ai-thinker.com/_media/esp8266/docs/esp-12f_product_specification_en.pdf

[16] Yan Pei et al., "An Elementary Introduction to Kalman Filtering," *Communications of the ACM*, vol. 62, no. 11, pp. 122-133, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] Alfian Ma'arif et al., "Kalman Filter for Noise Reducer on Sensor Readings," *Signal and Image Processing Letters*, vol. 1, no. 2, pp. 50-61, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[18] Anggun Winursito, Ibnu Masngut, and Gilang N.P. Pratama, "Development and Implementation of Kalman Filter for IoT Sensors: Towards a Better Precision Agriculture," *IEEE 3rd International Seminar on Research of Information Technology and Intelligent Systems*, Yogyakarta, Indonesia, pp. 360-364, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Rio Ikhsan Alfian, Alfian Ma'arif, and Sunardi Sunardi, "Noise Reduction in the Accelerometer and Gyroscope Sensor with the Kalman Filter Algorithm," *Journal of Robotics and Control*, vol. 2, no. 3, pp. 180-189, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Qian-Qian Qian, and Yu-Qing Bao, "Application of Kalman-Filtering with Disturbance Estimation in Power System Frequency Control," *IEEE/IAS Industrial and Commercial Power System Asia*, Chengdu, China, pp. 782-787, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] Dah-Jing Jwo, and Amita Biswal, "Implementation and Performance Analysis of Kalman Filters with Consistency Validation," *Mathematics*, vol. 11, no. 3, pp. 1-19, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Arvind Dagur et al., *Artificial Intelligence, Blockchain, Computing and Security,* 1st eds, CRC Press, London, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Mustapha Dakkak et al., "Indoor Localization Method Based on RTT and AOA Using Coordinates Clustering," *Computer Networks*, vol. 55, no. 8, pp. 1794-1803, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[24] Suliman Abdulmalek et al., "IoT-Based Healthcare-Monitoring System towards Improving Quality of Life: A Review," *Healthcare*, vol. 10, no. 10, pp. 1-32, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] V. Sharmila et al., *Challenges in Information, Communication and Computing Technology*, Proceedings of the 2nd International Conference on Challenges in Information, Communication, and Computing Technology (ICCICCT 2024), Namakkal, Tamil Nadu, India, 1st eds, CRC Press, London, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[26] Shuang Wang et al., "Machine-Learning-Based Human Motion Recognition via Wearable Plastic-Fiber Sensing System," *IEEE Internet of Things Journal*, vol. 10, no. 20, pp. 17893-17904, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[27] Hyun Myung et al., "Mobile Robot Localization Using a Gyroscope and Constrained Kalman Filter," *IEEE SICE-ICASE International Joint Conference*, Busan, Korea (South), pp. 2098-2103, 2006. [CrossRef] [Google Scholar] [Publisher Link]