Approaches for Sensor Data Fusion

Md Fuad Hasan
Technische Hochschule Ingolstadt

***Abstract*—Sensors are basically used to connect with the environment of the system by using data.Sensor fusion is the process of integrating data from multiple sensors, and sensor data is normally used for more accurate and reliable information. Sensor data fusion is used in very practical fields such as wireless sensor networks, gesture recognition, obstacle detection, airborne radar, and IFF(Identification of Friends) and so on.There are several approaches for data fusion for combining various sensors data, reducing errors, improving reliability,and enabling real-time decision-making. This paper represents some fusion data approaches such as Bayesian approach, State Estimation and Kalman Filtering, Decision-Level Fusion, Fuzzy logic, etc.**

I. Introduction

Data fusion is a framework that includes techniques and resources for combining information from various sources to create more precise, accurate, and perceptive data. Strong data fusion methods are very important for reducing measurement errors and improving observation accuracy. Also, in dynamic and complex situations, it improves the dependability of systems functioning. Data fusion makes it possible to integrate information from multiple sources,which makes real-time decision-making easier. This is important in the rapidly evolving technological world of today [1].
Data fusion is important for many applications, such as distributed radar systems, robotics, wireless sensor networks, and driverless cars. It is specifically used for tasks like robust hand gesture identification, obstacle detection in autonomous systems, human activity recognition, and wireless network power optimization [4], [5].
There are a lot of methods of data fusion that have been created to meet requirements. These include data-level fusion, which directly combines raw sensor data;feature-level fusion, which extracts and combines relevant features from pre-processed sensor data; and decision-level fusion, which combines individual decisions from multiple sensors to reach a goal. Advanced techniques such as state estimation and Kalman filtering are used to estimate system states across time . While machine learning-based methods create complex fusion rules by training on multi-sensor datasets, Bayesian approaches use probabilistic reasoning to manage uncertainties. Each of these approaches has unique benefits and is selected according to the task's requirements and complexity [2], [7], [10].

II. Literature review

Data fusion is a framework which represents the means and tools for the tune of data and originating from various sources[1].Data fusion is very important for various aspects such as [1],

1) It reduces error and improve measurement accuracy.
2) It improves reliability
3) It gives comprehensive understanding in complex scenario
4) It increase robustness
5) It gives real time decision making

Data fusion is widely used in various fields for example [4], [5],

- Autonomous Vehicles and Robotics
- Distributed radar, acoustic and visual sensor network.
- airborne radar and IFF (Identification of Friends)
- For power saving in wireless sensor networks.
- Human activity recognition
- Obstacle detection in the autotaxi system
- Robust hand gesture recognition.

There are several approaches for data fusion such as,

- Data level fusion: In Data level fusion raw data from various sensors are merged without any any pre processing [11].

- Feature level fusion:In the Feature level fusion,the data from every sensor is pre-processed to extract relevant features (such as peaks in signals or edges in pictures). For additional processing, these attributes are then integrated [12].

- Decision-Level Fusion:In this case, every sensor analyzes its data separately to take a decision (such as classification or detection), and the individual judgments are then combined to reach a final, consensus-based decision [2].

- State Estimation and Kalman Filtering: This method combines sensor measurements across time to estimate the true state of a system using probabilistic models, such as particle filters (for non-linear systems) or Kalman filters (for linear systems) [3], [7].

- Bayesian Approaches:This approach effectively manages uncertainty by combining sensor data based on likelihoods and past knowledge using Bayesian inference [10].

- Fuzzy logic: Fuzzy logic is a mathematical framework in data fusion that manages ambiguous , precise the data, and uncertain information when combining data from various sources. It basically works like human model such as reasoning and decision making. It also very helpful when conventional probabilistic or deterministic methods are failed [6] .
- Machine Learning-Based Fusion:To develop complex fusion rules, machine learning methods like support vector machines(SVM) and neural networks are trained on multi-sensor data [9].

III. Approaches

A. Data level fusion

Data level fusion algorithm primarily uses adaptive weighted and support degree methods for reducing error among in same kind of sensors [11].

1) Adaptive weighted: The algorithm handle real time data and this is the main advantage of this algorithm.It also reduces mean square error [11].

Suppose, X=Unknown quality as observation when n sensors work at the same time. The other observations of multi-sensor are:

$$\{X_i\} \quad (i = 1, 2, \cdots, n)$$

The observation of the number i sensor can be expressed as follows:

$$X_i(t) = X(t) + n_i(t) \tag{1}$$

Where $n_i(t)$ represents the white noise added by real signal $X(t)$. The variance of $n_i(t)$ can be expressed as $\sigma_i^2 = E[n_i^2(t)]$

If the observations are non-interfering, the estimation of X can be defined as:

$$\hat{X}(k) = \frac{\sum_{i=1}^{n} W_i X_i}{\sum_{i=1}^{n} W_i} = \sum_{i=1}^{n} W_i X_i, \quad i = 1, 2, \cdots, n \tag{2}$$

Where, W_i is the weighting coefficient. And

$$\sum_{i=1}^{n} W_i = 1.$$

The estimated variance is:

$$\sigma^2 = \sum_{i=1}^{n} W_i^2 \sigma_i^2 \tag{3}$$

Here, σ_i^2 = noise variance of the ith sensor.

To get the minimum total variance σ^2, the auxiliary function should be:

$$\cdots$$

The auxiliary function can be expressed as:

$$f(W_1, W_2, \cdots, W_n, \lambda) = \sum_{i=1}^{n} W_i^2 \sigma_i^2 + \lambda \left(\sum_{i=1}^{n} W_i - 1 \right) \tag{4}$$

The minimization problem under the condition $\sum_{i=1}^{n} W_i = 1$ can be equivalent to the following problem:

$$\begin{cases} \frac{\partial f}{\partial W_1} = 2W_1\sigma_1^2 + \lambda = 0 \\ \frac{\partial f}{\partial W_2} = 2W_2\sigma_2^2 + \lambda = 0 \\ \vdots \\ \frac{\partial f}{\partial W_n} = 2W_n\sigma_n^2 + \lambda = 0 \\ \sum_{i=1}^{n} W_i - 1 = 0 \end{cases}$$

From this, we obtain:

$$\begin{cases} W_i = \frac{\mu}{\sigma_i^2}, \quad i = 1, \cdots, n; \quad \mu = -\frac{\lambda}{2} \\ W_1 + W_2 + \cdots + W_n = 1 \end{cases}$$

Through above formula , we can obtain:

$$\sum_{i=1}^{n} W_i = \mu \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} + \cdots + \frac{1}{\sigma_n^2} \right) \implies 1 = \mu \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \tag{5}$$

The final result is:

$$\mu = \frac{1}{\sum_{i=1}^{n} \frac{1}{\sigma_i^2}} \tag{6}$$

Substituting formula (6) into $W_j = \frac{\mu}{\sigma_j^2}, \quad j = 1, 2, \cdots, n$, we get the final optimal weighting factor:

$$W_j = \frac{\frac{1}{\sigma_j^2}}{\sum_{i=1}^{n} \frac{1}{\sigma_i^2}}, \quad j = 1, 2, \cdots, n \tag{7}$$

2) Support degree: The array consists of i sensors can adopt direct measurement to identify static parameters or gradient parameters X, namely [11]:

$$z_i(k) = X + v_i(k) \quad (i = 1, 2, \cdots, n) \tag{8}$$

Here, $z_i(k)$ = the observation identified by the number of i sensors at the moment of k, X = target; $v_i(k)$ = observation noise while $E[v_i]$ and $D[v_i]$ are unknown.

It is possible two sensors have a low support degree and deviate from each other when their difference between $z_i(k)$ and $z_j(k)$ is large.

If we want to see the observation support degree of each sensor at the same time,its need an exponential attenuation function and a support degree matrix.The exponential attenuation function calculates the non zero values and it provide good reliability and accuracy.At this time, the observation support degree of sensor i and sensor j is:

$$r_{ij} = \exp\left(-\tau[z_i(k) - z_j(k)]^2\right) \tag{9}$$

Here, τ= adjustable parameter. By adjusting the size of τ, it can easily regulate the measurement scale. At this time the support degree matrix will be:

$$R_n = \begin{bmatrix} 1 & r_{12} & \cdots & r_{1n} \\ r_{21} & 1 & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & 1 \end{bmatrix} \tag{10}$$

To represent the comprehensive support degree of the other $n-1$ sensors in z_i, a comprehensive support function s_i is defined:

$$s_i = \sum_{j=1, j\neq i}^{n} r_{ij}, \quad i = 1, 2, \cdots, n \tag{11}$$

Here, size of s_i is important because it reflects the other sensors.

The concordance measurement between sensor i's observation and the observations of the other $n-1$ sensors at time k can be expressed as follows:

$$\xi_i(k) = \frac{s_i}{n-1}, \quad i = 1, 2, \cdots, n \tag{12}$$

Equation (11), r_{ij} represents the proximity of two sensors' observations and equation (12), $\xi_i(k)$ represents the proximity of concordance measurement between the observation of sensor i and the observations of the other sensors at the same time.

After combining the observations from n sensors, the target's evaluation at time k can be written as follows:

$$\hat{X}(k) = \frac{\sum_{i=1}^{n} [\xi_i(k) z_i(k)]}{\sum_{i=1}^{n} \xi_i(k)}, \quad i = 1, 2, \cdots, n \tag{13}$$

Now we can identify the concordance mean value of the ith sensor, which can be represented as $\bar{\xi}_i(k)$ at the moment of k:

$$\bar{\xi}_i(k) = \begin{cases} \xi_i(1), & k = 1 \\ \frac{k-1}{k}\bar{\xi}_i(k-1) + \frac{1}{k}\xi_i(k), & k > 1 \end{cases} \tag{14}$$

The concordance variance of the ith sensor at time k is:

$$\sigma_i^2(k) = \begin{cases} 0, & k = 1 \\ \frac{k-1}{k}\sigma_i^2(k-1) + \frac{1}{k}\left[\bar{\xi}_i(k) - \xi_i(k)\right]^2, & k > 1 \end{cases} \tag{15}$$

The support degree based data fusion algorithm ignores the measurement variance caused by many factors such as, interference, other environmental impacts, sensors dependability [11].

When a fusion algorithm is based on adaptive weighting, it reduces the overall variance of the target parameter during fusion by adjusting the impact of each sensor observation variance on the weight coefficient [11].

Advantages [11]:

- Enhanced accuracy: Ensure accurate and reliable data fusion by reducing noise and volatility.
- Dynamic Adaptability: Modifies the weights instantly to account for the dependability of the sensor.
- Noise Robustness: Takes into account the surrounding conditions to bring noise closer to its actual value.
- Unbiased Estimation: Produces precise findings without requiring prior sensor data information.
- Real-Time Processing: Handles multisensor data in dynamic contexts with efficiency.

Disadvantages [11]:

- Parameter Sensitivity: Precise tuning of parameters is essential for performance.
- Computational Overhead: High resource demands for large sensor networks.
- Limited Fault Tolerance: Better at handling damaged sensors than malfunctioning ones.
- Noise Assumptions: Depends on ideal noise properties, which might not be applicable in all situations.

Challenges [11]:

- Scalability: The capacity to manage several sensors with minimal latency.
- Integrating data from sensors with varying precision and rates is known as sensor heterogeneity.
- An anomaly Reliability in the face of inaccurate or excessive sensor readings is known as robustness.
- Dynamic environment: Adjusting for quick changes in target behavior or sensor dependability.

B. *Feature level fusion*

Feature-level fusion, an intermediate method in multisensor data fusion, focuses on merging characteristics taken from unprocessed sensor data. Before making judgments, feature-level fusion incorporates the most instructive characteristics of sensor data, in contrast to raw-data-level fusion, which combines raw data directly or decision-level fusion, which combines decisions or outputs of different sensors. There are several steps for feature-level data fusion, these are [12],

Steps in Feature-Level Fusion [12]:

1) Data Acquisition: Various sensors record unprocessed data, such as temperature, sound, and pictures.
2) Pre-processing: To prepare the raw data for feature extraction, they should be cleaned, normalized, and transformed.
3) Feature Extraction: Statistical analysis, wavelet transforms, Fourier transforms, and deep learning algorithms are some of the methods used to extract important attributes from raw sensor data.For instance, texture, edges, or color histograms may be extracted by image sensors.Spectral or temporal features can be obtained from audio sensors.
4) Feature Fusion: A single feature vector is created by combining the features of every sensor. Fusion can be accomplished by: Direct joining of feature vectors is known as concatenation.Dimensionality reduction is the process of combining features and lowering dimensionality through the use of methods like PCA, t-SNE, or auto encoders.
5) Weighted Combination: Identifying characteristics according to their dependability or importance.

Advantages of Feature level fusion [8], [12]:

- Information improvement: combines complementing data from several sensors to increase the accuracy and reliability of the system as a whole.

- Noise Reduction: Reduces noise and redundancy by removing unnecessary data at the raw level and concentrating on important attributes.
- Ability to adjust with advanced models.
- Scalable: Suitable for a variety of uses, including health diagnostics,object detection and gesture recognition.

Challenges of Feature level fusion[[8], [12]:

- Feature Compatibility: Combining features from several modalities like audio and image can be more complex due to differences in scale, type, and representation.
- Problems with Dimensionality: The "curse of dimensionality" and higher processing requirements may result from the combined feature vector being high-dimensional.
- Feature Selection: In feature level data fusion needs to be careful data analysis because its important to identify which features are relevant and which are redundant.
- Data Imbalance: Careful weighing during fusion is necessary because sensors may have different level of accuracy or dependability.

C. Decision-Level Fusion

Decision level fusion is a multi sensor data fusion method where a final decision is made by combining the separate decision or outputs from multiple sensors or system. This method normally works with higher level information which makes it computationally efficient [2].

Steps in Decision-Level Fusion:

1) Data Acquisition: Multiple sensors simultaneously collect raw data.
2) Preprocessing: To reduce noise and enhance data quality, every sensor performs locally preprocess on its own data.
3) Local Analysis: To produce decisions or outputs, each sensor processes its raw data. Classifications, detections, probabilities can be used to represent these choices.
4) Decision Collection: A central fusion unit or node receives the outputs or decisions from every sensor.
5) Decision Fusion: A central fusion unit or node receives the outputs or decisions from every sensor.It uses fusion techinque such as Bayesian Inference, Fuzzy Logic,Dempster-Shafer Theory and voting.
6) Final Decision Making: The intended task, such as classification, detection, or control, is carried out using the fused output.

Advantages of Decision-Level Fusion [2]:

- Bandwidth Efficiency: There is less communication overhead because only the decisions are transmitted.
- Modularity: Enables distributed systems in which decisions are centralized and sensors function independently.
- Fault Tolerance:Reduces the impact of a single faulty sensor on the overall system.
- Flexibility: Adapts effectively to a variety of sensors with varying processing power and different data types.

Disadvantages of Decision-Level Fusion [2]:

- During local decision making process , sometime lost details information from raw data
- The overall fusion accuracy depends on individual sensor quality.
- Designing optimal fusion rules for complex systems can be challenging.
- The sequential decision-making process may take long time.

D. State Estimation and Kalman Filtering

In multi-sensor data fusion, state estimation and Kalman filtering are very popular approaches for estimating a dynamic system's state when noise and uncertainty are present. To get a proper assessment of the system's state, they combine noisy readings from different sensors with predictions from a system model [3], [7].

The Kalman filter algorithm mostly has two stages: prediction and measurement updates. The standard Kalman filter equations for the prediction stage are given as below [7]:

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1} + Gu_{k-1} \tag{1}$$

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

And the measurement update equations are presented as follows:

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

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k \left(\hat{y}_k - H\hat{x}_{k|k-1}\right) \tag{4}$$

$$P_k = (I - K_kH)\, P_{k|k-1} \tag{5}$$

Here,

- $\hat{x}_k$ = the predicted state vector,
- P_k = the state error covariance matrix,
- K_k = the Kalman gain matrix,
- $\hat{y}_k$ = the measurement output,
- I = the identity matrix.

To start the loop, the initial state estimation matrix $\hat{x}_0$ and the initial state error covariance matrix P_0 are required [7].

First, the state vector can be predicted by using Eq. (1). Secondly, the state error covariance matrix can also be predicted using Eq. (2). Once the predicted values are obtained, the Kalman gain matrix K_k is calculated by Eq. (3). To make a better estimation, the predicted state vector is updated through Eq. (4). Using equation (5) the state error covariance matrix is restored.

In fusion algorithms, observation matrix H is very crucial because H maps the state vector into the output vector.H matrix provides a linear transformation for linear Kalman filter. The H matrix represents which state variables are included and which are not.

For example, if the first and second states of a 3-dimensional state vector are measured, the H matrix should be selected as:

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \tag{6}$$

$$y_k = Hx_k \implies y_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (7)$$

Additionally, the H matrix can be treated as a state combination, which serves as a fusion tool. For example, in a system where the measured output depends on two steps, the observation matrix is assigned as shown in Eq. (7), providing the linear combination of input states:

$$H = \begin{bmatrix} 1 & 1 \end{bmatrix}; \quad (8)$$

$$y_k = Hx_k \implies y_k = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 + x_2 \quad (9)$$

State estimation and Kalman filtering in data fusion can performs to find optimal state estimates and easily handle noisy measurements. Also it supports in real-time applications through their recursive feature. However, their performance depends on accuracy and noise models, and they assume Gaussian noise, limiting effectiveness in non-Gaussian scenarios [3], [7].

E. Bayesian Approaches

Bayesian method is a classical probabilistic method for fusion images that created by several sensors. And Bayesian classifier is an optimal statistical classifier..By ensuring that all sensor data is combined probabilistically, Bayesian techniques offer a strict framework for data fusion that takes uncertainties into account [10].

Steps of Bayesian Approaches in Data Fusion [10]:

1) Prior Definition: Begin with a prior probability distribution (P(X)) denoting the initial state of the system before receiving any data.
2) Acquisition of Sensor Data: Sensor Data Acquisition: Gather data (Z) from various sensors or sources, which provide observations about the state.
3) Likelihood Calculation: Compute the likelihood (P(Z|X)) that the observed data Z would occur for a given state X. It represents the relationship between observed data and the state.It defines the relationship between observed data and state.
4) Bayesian Update: To calculate the posterior probability, apply the Bayes theorem to update the prior belief using the new sensor data. Here,

$$P(X|Z) = \frac{P(Z|X)P(X)}{P(Z)}$$

   Here, $P(Z)$ normalizes the result to ensure the probabilities sum to 1.
5) Fusion of Data: If multiple sensors provide independent data $(Z_1, Z_2, \ldots, Z_n)$, combine the likelihoods iteratively:

$$P(X|Z_1, Z_2, \ldots, Z_n) \propto P(Z_1|X) \cdot P(Z_2|X) \cdot \cdots \cdot P(X)$$

6) State Estimation: Derive the most probable state estimate from the posterior distribution, such as:
   - Maximum a posterior (MAP) estimate:

$$X = \arg\max P(X|Z)$$

   - Expected value of the posterior:

$$\mathbb{E}[X] = \int X P(X|Z)\, dX$$

7) Iterative Process: when new data becomes available,repeat the steps.

Bayesian approaches in data fusion are highly beneficial because they can handle uncertainty. They provide mathematical framework for combining data because they use prior knowledge. In dynamic systems, it gives more accurate and reliable state estimations. However, Bayesian techniques can be computationally demanding, especially when dealing with huge datasets or high-dimensional situations [10] .

F. Fuzzy logic

Sometimes information is frequently noisy, incomplete, or ambiguous when extracted from various data sources. Fuzzy logic uses degrees of truth instead of binary logic, enabling more complex models. Here is a well-organized structure of the use of fuzzy logic in data fusion [6]:

1) Define the Problem and Data Sources:
   - Identify the data sources(e.g., sensors, systems, models) and also objective of the fusion (e.g., classification, prediction, estimation)
2) Fuzzification (Input Processing): Use membership functions to transform the sources' numerical data into fuzzy values.It takes some few steps:
   a) Define Input Variables: Identify the measurable attributes from each data source.
   b) Identify Membership Functions: Create fuzzy sets for each input variable using linguistic term such as low, medium, high.
   c) Map numerical Values to Fuzzy Sets: Apply membership functions to transform numerical data into fuzzy degrees of membership (values between 0 and 1)
3) Create Fuzzy Rules:To specify how fuzzy inputs are merged to create a fuzzy output, create a series of if-then rules.
4) Fuzzy Inference (Decision Making):To generate fuzzy outputs, combine the fuzzy inputs and evaluate the fuzzy rules.In here some inference Methods are used such as Mamdani Method,Sugeno Method and Tsukamoto Method.
5) Defuzzification (Output Processing): Return the fuzzy output to a numerical value so that it can be interpreted or processed further.Also in here some methods are used for defuzzification such as,Centroid (Center of Gravity),Mean of Maximum (MoM) and Weighted Average.

6) Post-Fusion Analysis:Integrate the numerical output for decision-making, visualization, or additional calculations, and also evaluate the performance of fuzzy logic.

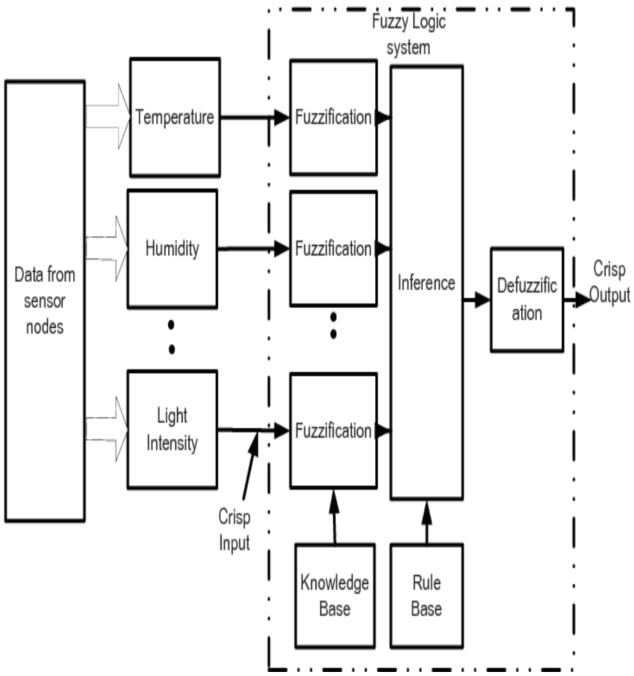

Fig: fuzzy logic system with multisensor [6]

Fuzzy logic is very popular because it can handle uncertainty and ambiguity effectively, mimicking human-like reasoning with linguistic terms and simple if-then rules. It is also flexible, robust to noise, and easy to implement. However, it is computationally intensive for large systems, and it is not always optimal[9].

IV. Conclusion

Data fusion is a crucial framework for combining various sensors data, lowering mistakes, improving reliability, and enabling real-time decision-making. From data-level fusion to advanced models like Kalman filtering, Bayesian approaches,Fuzzy logic, and machine learning, it can handle complex problems like noise and uncertainty while adjusting to a variety of applications, including robots and autonomous systems. By improving these approaches and addressing challenges like heterogeneity and scalability, data fusion will keep innovation in dynamic environments and intelligent systems.

References

[1] R.M. Akita. User based data fusion approaches. In *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997)*, volume 2, pages 1457–1462 vol.2, 2002.

[2] Peijun Du, Wei Zhang, Shubi Zhang, and Junshi Xia. Hyperspectral remote sensing image classification based on decision level fusion. In *2009 IEEE International Geoscience and Remote Sensing Symposium*, volume 4, pages IV–940–IV–943, 2009.

[3] Adelina Ioana Ilieş, Gabriel Chindriş, and Dan Pitică. A comparison between state of charge estimation methods: Extended kalman filter and unscented kalman filter. In *2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pages 376–381, 2020.

[4] Bing Liang and Qun Liu. A data fusion approach for power saving in wireless sensor networks. In *First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)*, volume 2, pages 582–586, 2006.

[5] Kui Liu, Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Fusion of inertial and depth sensor data for robust hand gesture recognition. *IEEE Sensors Journal*, 14(6):1898–1903, 2014.

[6] P. Manjunatha, A.K. Verma, and A. Srividya. Multi-sensor data fusion in cluster based wireless sensor networks using fuzzy logic method. In *2008 IEEE Region 10 and the Third international Conference on Industrial and Information Systems*, pages 1–6, 2008.

[7] Sadra Mousavi, Duygu Bayram, and Serhat Seker. Current data fusion through kalman filtering for fault detection and sensor validation of an electric motor. In *2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, pages 155–160, 2019.

[8] Santiago Planet and Ignasi Iriondo. Comparison between decision-level and feature-level fusion of acoustic and linguistic features for spontaneous emotion recognition. In *7th Iberian Conference on Information Systems and Technologies (CISTI 2012)*, pages 1–6, 2012.

[9] Karsten Schwalbe, Alexander Groh, Frank Hertwig, and Ullrich Scheunert. Data fusion strategy to improve the realiability of machine learning based classifications. In *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–5, 2019.

[10] Yu-qiu Sun, Jin-wen Tian, and Jian Liu. Target recognition using bayesian data fusion method. In *2006 International Conference on Machine Learning and Cybernetics*, pages 3288–3292, 2006.

[11] Chang-tao Wang, Yi Zhu, and Zhong-hua Han. The application of data-level fusion algorithm based on adaptive-weighted and support degree in intelligent household greenhouse. In *2017 9th International Conference on Modelling, Identification and Control (ICMIC)*, pages 728–733, 2017.

[12] Stefan Wender and Klaus C. J. Dietmayer. A feature level fusion approach for object classification. In *2007 IEEE Intelligent Vehicles Symposium*, pages 1132–1137, 2007.