applied sciences

MDPI

Article

Multi-Dimensional Fuzzy Clustering-Based Trajectory Initialization Algorithm for Infrared Weak Target Trajectories in Robust Clutter Environments

Ziqian Yang [1,2], Hongbin Nie [1], Yiran Li [1,2] and Chunjiang Bian [1,*]

1 National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China; nsscyzq@163.com (Z.Y.); niehongbin@nssc.ac.cn (H.N.); lyr15710002670@163.com (Y.L.)
2 University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: bianchunjiang@nssc.ac.cn

Abstract: When conducting maneuver target tracking, trajectory initialization plays a crucial role in enhancing the accuracy of tracking algorithms. During maneuver target tracking, the accuracy of the tracking algorithm can be significantly improved through trajectory initialization. However, the traditional trajectory initialization algorithms face issues such as susceptibility to noise interference, lack of universality, and poor robustness in environments with high clutter levels. To address these issues, this study proposes a trajectory initialization algorithm based on multidimensional fuzzy clustering (MDF-clustering). The algorithm utilizes multidimensional feature information of the target, such as speed and irradiance, to determine point trajectory affiliation by assigning weights based on the clustering center of each feature type. Subsequently, it updates the clustering center and weight assignment using the new target features, ultimately deriving the correct trajectory through iterative processes. Experimental results demonstrate that the proposed method achieves an average stable initialization frame number of 3.12 frames, an average correct trajectory initialization rate of 99.59%, an average false trajectory occupancy rate of 0.04%, and an average missed batch rate of 0.06%. These results indicate improvements of at least 0.87 frames, 27.11%, 60.28%, and 6.48%, respectively, in terms of initialization rate, false trajectory rate, and missed batch rate, when compared to traditional methods. The algorithm enhances the accuracy and robustness of trajectory initialization in challenging environments characterized by solid clutter and target maneuvers, offering significant practical value for target tracking in complex scenarios.

Keywords: multi-target tracking; trajectory batching; fuzzy clustering; data fusion

check for updates

Citation: Yang, Z.; Nie, H.; Li, Y.; Bian, C. Multi-Dimensional Fuzzy Clustering-Based Trajectory Initialization Algorithm for Infrared Weak Target Trajectories in Robust Clutter Environments. *Appl. Sci.* **2024**, *14*, 9935. https://doi.org/10.3390/app14219935

Academic Editor: Angelo Luongo

Received: 4 July 2024
Revised: 26 September 2024
Accepted: 2 October 2024
Published: 30 October 2024

1. Introduction

Multi-target tracking is widely used in surveillance, autonomous driving, and military applications. Given the escalating complexity of the global security landscape, the significance of this technology within the military domain has become increasingly pronounced, particularly in military early warning, battlefield situational awareness, and target reconnaissance. Multi-target tracking is an essential technology in modern military operations, significantly improving operational efficiency, precision, and real-time strategic decision-making [1].

Recent sensor and image processing advancements introduced novel algorithms for tracking maneuvering targets. For instance, Junwei et al. [2] proposed a novel method to enhance the accuracy and efficiency of maneuvering target tracking by analyzing and processing sensor data in intricate environments. Similarly, RuiPing Ji et al. [3] introduced a concave optimization-based initialization method for multi-target tracking, leveraging optimization algorithms to improve target identification and classification during the early stages. Wei Yi et al. [4] developed a multi-frame tracking pre-detection algorithm tailored to scenarios characterized by high dynamic variations and low signal-to-noise ratios, thereby addressing the complexities of maneuver target tracking. Additionally, Yaakov Bar-Shalom et al. [5] devised a multi-model estimator for tracking sharply maneuvering ground targets, enhancing tracking accuracy through the integration of multiple motion models to adapt to rapid changes in target motion. Current methods work well in certain cases but struggle to generalize across environments and target types.

The trajectory initialization process, as a crucial aspect of target tracking [6], serves as a digital representation of a natural physical phenomenon. It is essential for describing, analyzing, and solving problems related to tracking maneuvering targets. The method converts sensor data into a digital format, enabling dynamic target analysis. Trajectory initialization not only plays a key role in efficient target tracking and monitoring but also significantly impacts various aspects of tracking, such as accuracy, prediction capability, and responsiveness to complex airspace situations. Therefore, the efficiency and accuracy of trajectory initialization are vital for real-time data processing, flight tracking [7], and long-term target motion prediction and analysis. Researching effective and stable trajectory initialization methods is crucial for target detection and tracking, especially in scenarios with solid clutter. Efficient initialization of trajectory parameters is essential for ensuring stable and accurate tracking systems. Enhancing the universality, accuracy, and response speed of trajectory initialization algorithms directly influences the stability and practicality of tracking systems. Trajectory initialization methods are categorized as batch or sequential processing [8]. Sequential processing techniques, such as the intuitive method, logic-based method, and modified logic method, are suitable for clutter-free scenes. On the other hand, batch processing techniques, like the Hough transform, suit high-clutter scenarios. Various trajectory initialization algorithms have been proposed in the literature. Dong Zhirong [9] introduced a method based on kinematics to determine if measured data belongs to the same target. Su Feng et al. [10] enhanced the trajectory initialization algorithm by incorporating constraints in the correlation domain to eliminate V-shaped measurement point traces. Xu Hong et al. [11] improved the traditional correlation wave gate structure with angular constraints using a modified Hough transform for batch processing. Wang Guohong et al. [12] developed a trajectory initialization algorithm that combines the Hough transform and the 3/4 logic method. Jin, Z et al. [13] presented a multi-hypothesis, multi-rule, adaptive trajectory initialization algorithm that filters clutter with a density clustering algorithm, retains historical information, and utilizes an adaptive initial period to suppress false targets. Liu Zeng et al. [14] proposed a trajectory initialization algorithm based on adaptive gates and a fuzzy Hough transform, effectively reducing false alarms and achieving high-quality trajectory initialization. Feng Yang et al. [15] introduced a density-based random sample consensus (DB-RANSAC) algorithm for trajectory initialization in dense clutter environments, leveraging spatiotemporal domain information to enhance initialization efficiency. Alejandro Pastor et al. [16] introduced a TSEPM algorithm that integrates multi-hypothesis tracking with a modified logic-based trajectory initialization method. This approach aims to enhance detection rates and decrease false alarms. In scenarios where trajectory initialization for weak targets in cluttered environments is challenging, conventional algorithms are often hindered by noise, frequent target ID changes, and difficulties in trajectory synthesis. These issues can lead to unstable initial feature parameters for the target, ultimately impacting the accuracy, stability, and robustness of subsequent tracking processes.

The accuracy of these methods relies heavily on precise initial conditions and parameter settings, necessitating a comprehensive understanding of algorithms and application contexts for practical adjustments. Trajectory initialization is vital for obtaining trajectory parameters early and resolving issues from incorrect initial conditions. By optimizing these initial conditions and parameters, trajectory initialization helps mitigate the impact of environmental noise and other disturbances. As such, trajectory initialization not only enhances the accuracy and efficiency of target tracking but also serves as a fundamental strategy to overcome limitations in existing tracking methods and enhance algorithm adaptability and generalization.

Recent advancements in predictive modeling, particularly in complex physical systems, have introduced methods that integrate domain knowledge with data-driven models. One such approach is the Kolmogorov-Arnold Network (KAN), which has shown great promise in improving the robustness and accuracy of systems operating in dynamic and noisy environments. KAN enables the incorporation of expert knowledge into the learning process, allowing for better handling of uncertainty and noise. This has been particularly effective in tasks such as trajectory initialization and human–robot interaction, where the system needs to adapt rapidly to changing conditions and high levels of environmental clutter.

The field of target tracking has seen significant progress in recent years, especially in tackling challenges posed by cluttered environments and dynamic target behaviors. Current research demonstrates a growing emphasis on improving the robustness and adaptability of tracking algorithms. Techniques have become more sophisticated in handling noise interference and adapting to varying target motions. However, many approaches still face challenges in maintaining consistency across different environments. For instance, algorithms often perform well under controlled conditions but encounter difficulties when applied to more unpredictable or real-world scenarios. Furthermore, while accuracy has improved, issues related to false alarms and initialization delays persist. These limitations highlight the need for further advancements in achieving reliable, real-time tracking, especially in environments with high levels of clutter and target maneuverability. Despite these challenges, the continuous refinement of existing algorithms shows promising potential for future developments.

This paper presents a trajectory initialization algorithm using multi-dimensional fuzzy clustering to solve challenges in cluttered environments and target maneuvers. The algorithm combines target motion and appearance features, assigning dynamic feature weights for accuracy, robustness, and adaptability. The key contributions are:

(1) Multi-dimensional feature fusion and dynamic weight allocation mechanism: By integrating multi-dimensional feature information, the algorithm can better differentiate between targets and background noise. The adaptive weight allocation mechanism adjusts the weight of each feature based on target characteristics and environmental conditions, enhancing the algorithm's universality and accuracy.

(2) Enhanced fuzzy clustering processing mechanism: Traditional fuzzy clustering algorithms are prone to noise interference in complex environments, reducing clustering accuracy. This algorithm mitigates the impact of environmental noise and non-target factors by optimizing clustering center selection and updating strategies, thereby improving clustering accuracy and stability.

The remainder of this paper is organized as follows: Section 2 introduces the trajectory initialization algorithm based on MDF-clustering. Section 3 describes the experimental setup and the evaluation metrics used to assess the performance of the algorithm. Section 4 presents and analyzes the experimental results. Finally, Section 5 concludes the paper and discusses potential directions for future research.

2. Trajectory Initialization Based on MDF-Clustering

In complex environments, traditional trajectory initialization algorithms are often hindered by issues such as noise interference, lack of universality, and limited robustness, especially when dealing with high-level clutter. These limitations are particularly evident in scenarios with high clutter and maneuvering targets. To overcome these challenges, this paper proposes a novel trajectory initialization algorithm based on multidimensional fuzzy clustering (MDF-clustering). By incorporating multidimensional feature information such as speed and irradiance, the algorithm dynamically assigns feature weights to accurately determine trajectory affiliation. This approach not only enhances the accuracy and robustness of trajectory initialization but also adapts effectively to rapidly changing and complex environments.

2.1. Basic Concepts of Fuzzy Clustering

In the application of the fuzzy clustering algorithm [17,18], consider a set containing n data points, let the dataset $\boldsymbol{X} = \{\boldsymbol{x_1} \ldots, \boldsymbol{x_n}\}$, and its fuzzy set $\boldsymbol{K}$ is represented by the matrix $\boldsymbol{U} = [\boldsymbol{u_{ij}}]$, where the element $\boldsymbol{u_{ij}}$ denotes the degree of affiliation of the data point $\boldsymbol{x_j}$ to the ith category, and $\boldsymbol{u_{ij}}$ satisfies the following conditions.

$$\forall \boldsymbol{j}, \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{u_{ij}} = \mathbf{1} \tag{1}$$

$$\forall \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{u_{ij}} \in [\mathbf{0}, \mathbf{1}] \tag{2}$$

$$\forall \boldsymbol{i}, \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{u_{ij}} > \mathbf{0} \tag{3}$$

The current widely used clustering criterion is to minimize the sum of squared weighted errors within classes, i.e.,

$$(\boldsymbol{min})\boldsymbol{J_m}(\boldsymbol{U}, \boldsymbol{V}) = \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{c}} \boldsymbol{u_{ij}^m d_{ij}^2}(\boldsymbol{x_j}, \boldsymbol{v_i}) \tag{4}$$

This is usually carried out by determining the clustering center V. In this method, m is used as a weighted index and $d_{ij}(x_j, v_i) = \|v_i - x_j\|$ as a measure of the distance between x_j and v_i.

The Lagrange multiplier method is employed to solve this objective function, providing and optimal solution for the affiliation matrix U and the clustering center V:

$$\boldsymbol{u_{ij}} = \left[\sum_{\boldsymbol{k=1}}^{\boldsymbol{K}} \left(\frac{\boldsymbol{d_{ij}}}{\boldsymbol{d_{kj}}} \right)^{\frac{\mathbf{2}}{\boldsymbol{m-1}}} \right]^{-\mathbf{1}} \tag{5}$$

$$\boldsymbol{v_i} = \frac{\sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{x_j} \cdot \boldsymbol{\mu_{ij}^m}}{\sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{\mu_{ij}^m}} \tag{6}$$

2.2. Fuzzy Clustering Based on Multidimensional Features

$\chi = (X_1, \ldots, X_M)$ is a set of input data after target detection,. There are n targets in the data, which in turn contain M target feature variables, where these M variables, as a kind of mixed feature variables, possess multiple types of features, such as data types, qualitative data, time series data, etc. That is, the input dataset χ contains S types of feature variables, and for every kind of feature attribute, there are p_s variables, so

$$s = 1, \ldots, S; 1 < S \leq P;\ 1 \leq p_s < P \tag{7}$$

can be interpreted as follows: the feature variables of the first n_1 targets are of the same type of variables (e.g., data type), and the feature variables of the second n_2 targets are also of the same kind of variable, but of a different kind (e.g., qualitative variable) than the characteristic variable of the first n_1, so that

$$\chi = (\chi_1, \ldots, \chi_s, \ldots, \chi_S) \tag{8}$$

where $\chi_s \equiv \{X_{p_1+\ldots+p_{s-1}+1}, \ldots, X_{p_1+\ldots+p_s}\}$ represents the set of feature variables of the sth type. And χ_{is} represents the set of values observed for the ith target on the p_s variable of the sth type.

Since the feature types in this paper include two types, one for data variables and one for qualitative variables, let $\chi = \{\chi_1, \chi_2\}$, where χ_1 is the set of two data type variables and

χ_2 is the set of two qualitative variables. Then, $S = 2$, $n_1 = n_2 = 2$, $P = 4$, $\chi_1 = \{X_1, X_2\}$ and $\chi_2 = \{X_3, X_4\}$.

For data type variables, $\chi_{is} = X_{is}$ is a vector of p_s values observed on the *i*th target. For a data sequence of length Q, $\chi_{is} = X_{is}$ is a $Q \times p_s$ matrix, where the rows of the matrix represent the features entered in the sequence $q(q = 1, \ldots, Q)$; and the columns of the matrix represent the sequence on the p_s variable of the data input on the *i*th target [19]. The same is true for the qualitative variable types.

Multidimensional data presents a unique challenge in trajectory initialization due to the varied nature of the target features. These features can include both numeric variables, such as speed and irradiance, and qualitative variables, such as target shape and contrast. To handle these diverse data types, the MDF-clustering algorithm employs a combination of Mahalanobis distance for numeric variables and a symbolic distance metric for qualitative variables.

The Mahalanobis distance is particularly effective in this context as it adjusts for correlations between different features [20], ensuring that the scale of the features does not skew the clustering results. Meanwhile, the symbolic distance metric is designed to handle qualitative attributes, which do not have a natural numeric representation but are nonetheless crucial for distinguishing between different targets.

By integrating these distance metrics [21], the algorithm calculates the overall weighted distance between the target's feature set and the clustering center. This weighted distance is then minimized through an iterative process, allowing the algorithm to continuously refine its cluster assignments and improve the accuracy of trajectory initialization in cluttered and noisy environments.

On the other hand, traditional metrics for numeric data are not suitable for symbolic data, as symbolic data possess unique properties and structures. The expression and meaning of symbolic data can differ significantly from numerical data. Therefore, a specific metric tailored to symbolic data characteristics, such as the symbolic data difference measure, is necessary to assess similarities or differences between symbolic data accurately. Therefore, the distance between different targets, i.e., the distance between the *i*th unit and the i' unit calculated according to the nature of the *i*th variable type, is as follows:

$$ {}_s d_{ii'} = d(\chi_{is}, \chi_{i's}) \tag{9} $$

We can obtain

$$ d_{ii'}^2 = \sum_{s=1}^{S} (w_s \cdot {}_s d_{ii'})^2 = \sum_{s=1}^{S} [w_s \cdot d(\chi_{is}, \chi_{i's})]^2 \tag{10} $$

The above equation is the overall weighted squared distance considering S attribute types. Since the weights of the squared distances are in quadratic form, the distances are weighted squared.

In the scenario of this paper, ${}_1 d_{ii'} = d(\chi_{i1}, \chi_{i'1})$, ${}_2 d_{ii'} = d(\chi_{i2}, \chi_{i'2})$ are the pairwise distance matrices, such as the Martian distance of χ_1 and the symbolic distance of χ_2, respectively. Then, the total distance is:

$$ d_{ii'}^2 = (w_1 \cdot {}_1 d_{ii'})^2 + (w_2 \cdot {}_2 d_{ii'})^2 \tag{11} $$

According to the fuzzy association matrix in the fuzzy framework, suppose $\widetilde{\chi}_s = \left\{ \widetilde{\chi}_{1s}, \ldots, \widetilde{\chi}_{cs}, \ldots, \widetilde{\chi}_{C's} \right\}$ is a subset of χ_s, the clustered clusters are C, and $\widetilde{\chi}_{C's} \in \widetilde{\chi}_s$ is the eigenvalue in the C'th cluster in $\widetilde{\chi}_s$. Then, $\widetilde{\chi}_s = \left\{ \widetilde{\chi}_{1s}, \ldots, \widetilde{\chi}_{cs}, \ldots, \widetilde{\chi}_{C's} \right\}$ is a subset of χ representing the centroid of the current cluster, and its number of clusters after clustering is C.

The above formulas can be summarized as follows, providing a concise representation of the key mathematical relationships used in this study.

$$\begin{aligned} min: \sum_{i=1}^{n}\sum_{c=1}^{C} u_{ic}^{m} d_{ic}^{2} &= \sum_{i=1}^{n}\sum_{c=1}^{C} u_{ic}^{m} \sum_{s=1}^{S} (w_s \cdot {}_s d_{ic})^2 \\ &= \sum_{i=1}^{n}\sum_{c=1}^{C} u_{ic}^{m} \sum_{s=1}^{S} [w_s \cdot d(\chi_{is}, \chi_{i's})]^2 \\ (s.t.) & \sum_{c=1}^{C} u_{ic} = 1, u_{ic} \geq 0 \\ & \sum_{s=1}^{S} w_s = 1,\ w_s \geq 0 \end{aligned} \tag{12}$$

where u_{ic} denotes the affiliation of the ith object in the cth cluster; $m > 1$ is a weight index to control the ambiguity of the resulting partition; $\widetilde{\chi}_{cs}$ is the cth component of the cth centroid, associated with the sth variable type; ${}_s d_{ic} = d\left(\chi_{is}, \widetilde{\chi}_{C's}\right)$ denotes the distance between the ith observation and the cth centroid according to the sth variable type; to facilitate the comparison of the different attribute types, the S-distance ${}_s d_{ic}$ is normalized to vary in the range [0,1]; $d_{ic}^2 = \sum_{s=1}^{S} (w_s \cdot {}_s d_{ic})^2$ is the overall weighted squared distance between unit i and centroid c based on all variable types; w_s is the weight associated with the sth variable type, and therefore with the sth distance $(s = 1, \ldots, S)$.

The weights w_s constitute the specific parameters to be estimated in the clustering procedure.

The formula for the degree of affiliation and the formula for the variable type weights are derived by solving Equation (12):

$$\boldsymbol{u_{ic}} = \frac{\mathbf{1}}{\sum_{C'=1}^{C}\left[\frac{\sum_{S=1}^{S}(w_s \cdot {}_s d_{ic})^2}{\sum_{S=1}^{S}(w_s \cdot {}_s d_{ic'})^2}\right]^{\frac{1}{m-1}}} \tag{13}$$

$$\boldsymbol{w_S} = \frac{\mathbf{1}}{\sum_{S'=1}^{S}\left[\frac{\sum_{i=1}^{n}\sum_{C=1}^{C} u_{ic}^{m} {}_s d_{ic}^{2}}{\sum_{i=1}^{n}\sum_{C=1}^{C} u_{ic}^{m} {}_{s'} d_{ic}^{2}}\right]^{\frac{1}{m-1}}} \tag{14}$$

In the context of this study, the relevance of different attribute types in the data clustering process is crucial. It is essential to prioritize attribute types that significantly differentiate between groups when clustering target objects [22]. The weight $\boldsymbol{w_S}$ represents the sum of deviations of type S variables within each cluster, indicating the level of consistency within the group. As the deviation of a type S variable decreases within the target group, its weight $\boldsymbol{w_S}$ increases. This highlights that the optimization process values variables that increase similarity among units within clusters. Therefore, it is possible to iteratively adjust weights to gradually reach the final weight assignment.

By emphasizing the importance of variable types in cluster analysis for distinguishing groups, type weights $\boldsymbol{w_S}$ are introduced to quantify the impact of individual variables on group consistency. These weights not only reflect the homogeneity of group characteristics but also unveil the true significance of each variable type as the algorithm dynamically adjusts during iterations. Within this framework, the Lagrange multiplier method is employed to solve the constrained optimization problem, ensuring that the sum of affiliation degrees adheres to the requirements of a probability distribution. A Lagrangian function is formulated to incorporate both affiliation $\boldsymbol{u_{ic}}$ and weight $\boldsymbol{w_S}$, with these parameters optimized by determining the extreme values of this function as shown in Equation (15):

$$\boldsymbol{L_m(u_i, \lambda) = \sum_{i=1}^{n}\sum_{c=1}^{C} u_{ic}^{m} \sum_{s=1}^{S} (w_s \cdot {}_s d_{ic})^2 - \lambda\left(\sum_{c=1}^{C} u_{ic} - 1\right)} \tag{15}$$

where $\boldsymbol{u_i} = (\boldsymbol{u_{i1}}, \ldots, \boldsymbol{u_{ic}}, \ldots, \boldsymbol{u_{iC}})'$ and $\boldsymbol{\lambda}$ is the Lagrange multiplier. Therefore, setting the first order derivatives of Equation (15) with respect to $\boldsymbol{u_{ic}}$ and $\boldsymbol{\lambda}$ to zero yields:

$$\frac{\partial \boldsymbol{L_m}(\boldsymbol{u_i}, \boldsymbol{\lambda})}{\partial \boldsymbol{u_{ic}}} = \mathbf{0} \Leftrightarrow \boldsymbol{m}\boldsymbol{u_{ic}^{m-1}} \sum_{\boldsymbol{s=1}}^{\boldsymbol{S}} (\boldsymbol{w_s} \cdot {}_{\boldsymbol{s}}\boldsymbol{d_{ic}})^{\mathbf{2}} - \boldsymbol{\lambda} = \mathbf{0} \tag{16}$$

$$\frac{\partial \boldsymbol{L_m}(\boldsymbol{u_i}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \mathbf{0} \Leftrightarrow \sum_{\boldsymbol{c=1}}^{\boldsymbol{C}} \boldsymbol{u_{ic}} - \mathbf{1} = \mathbf{0} \tag{17}$$

According to Equation (16) it can be concluded that

$$\boldsymbol{u_{ic}} = \left[\frac{\boldsymbol{\lambda}}{\boldsymbol{m}\sum_{\boldsymbol{s=1}}^{\boldsymbol{S}} (\boldsymbol{w_s} \cdot {}_{\boldsymbol{s}}\boldsymbol{d_{ic}})^{\mathbf{2}}}\right]^{\frac{\mathbf{1}}{\boldsymbol{m-1}}} \tag{18}$$

It can be derived by also considering Equation (17):

$$\frac{\boldsymbol{\lambda}}{\boldsymbol{m}} = \left\{\frac{\mathbf{1}}{\sum_{\boldsymbol{c=1}}^{\boldsymbol{C}'} \left[\frac{\mathbf{1}}{\sum_{\boldsymbol{s=1}}^{\boldsymbol{S}} (\boldsymbol{w_s} \cdot {}_{\boldsymbol{s}}\boldsymbol{d_{ic}})^{\mathbf{2}}}\right]}\right\}^{\frac{\mathbf{1}}{\boldsymbol{m-1}}} \tag{19}$$

Finally, Equation (19) is substituted into Equation (18) to obtain u_{ic} as shown in Equation (13). u_{ic} is then fixed to yield w_s. The Lagrangian function is

$$L_w(\boldsymbol{w}, \xi) = \sum_{i=1}^{n} \sum_{c=1}^{C} u_{ic}^m \sum_{s=1}^{S} (w_s \cdot {}_s d_{ic})^2 - \xi \left(\sum_{c=1}^{C} u_{ic} - 1 \right) \tag{20}$$

where $\boldsymbol{w} = (w_1, \ldots, w_s, \ldots, w_S)'$ and ξ are Lagrange multipliers. Setting the first order derivatives of Equation (20) with respect to w_s and ξ equal to zero, one obtains respectively

$$\frac{\partial L_m(\boldsymbol{w}, \lambda)}{\partial w_s} = 0 \Leftrightarrow 2w_s \sum_{i=1}^{n} \sum_{c=1}^{C} u_{ic}^m \cdot {}_s d_{ic}^2 - \xi = 0 \tag{21}$$

$$\frac{\partial L_m(\boldsymbol{w}, \lambda)}{\partial \xi} = 0 \Leftrightarrow \sum_{s=1}^{S} w_s - 1 = 0 \tag{22}$$

It can be derived from Equation (21):

$$w_s = \frac{\xi}{2\sum_{i=1}^{n} \sum_{c=1}^{C} u_{ic}^m \cdot {}_s d_{ic}^2} \tag{23}$$

After that, it is derived using Equation (21):

$$\frac{\xi}{2} = \frac{1}{\sum_{s=1}^{S} \left(\frac{1}{\sum_{i=1}^{n} \sum_{c=1}^{C} u_{ic}^m \cdot {}_s d_{ic}^2} \right)} \tag{24}$$

Then, the weight formula in Equation (14) is obtained by replacing Equation (24) with Equation (23). In this way, using the combination of the weight information and the affiliation degree information of fuzzy clustering, the multi-dimensional fuzzy clustering is achieved to differentiate the target and background clutter.

The detailed step-by-step process can be found in Algorithm 1, which illustrates the initialization membership update, weight update, and centroid recalculation steps of the MDF-clustering algorithm.

Algorithm 1 Multi-Dimensional Fuzzy Clustering Method (MDF-Clustering) Algorithm

Input:
- χ: Set of detected target points
- C: Number of clusters
- Max_iter: Maximum number of iterations
- S: Number of feature types
- m: Fuzziness parameter

Output:
- Cluster centroids $\widetilde{\chi}_{Cs}$
- Affiliation matrix U

1. Initialize:
   1.1. Set ***iteration*** = 0;
   1.2. Set the **c** value and Max_iter
   1.3. Randomly select initial medoids: $\widetilde{\chi}_s = \left\{\widetilde{\chi}_{1s}, \ldots, \widetilde{\chi}_{cs}, \ldots, \widetilde{\chi}_{Cs}\right\}, s = 1, \ldots, S$
2. Repeat the following steps until convergence or max iteration reached:
   2.1. Store the current medoids: $\widetilde{\chi}_{OLD,s} = \widetilde{\chi}_s, s = 1, \ldots, \mathrm{S}$
   2.2. Update affiliation matrix $\boldsymbol{u}$ for each point i and cluster c
   2.3. Update weights for each feature type w_s
   2.4. Recalculate centroids for each cluster $\widetilde{\chi}_{Cs}$
   For each cluster c, recalculate the centroid $\widetilde{\chi}_{Cs}$ by finding the point q that minimizes the total distance:
   $q = argmin_{1 \le i' \le n} \sum_{i''=1}^{n} u_{i''}^{m} \sum_{s=1}^{S} \left(w_s \cdot {}_s d_{i',i''}\right)^2$
   2.5. Update centroids for each feature type S with the newly calculated medoids $\widetilde{\chi}_{Cs}$
   2.6. Check for convergence:
   2.7. Increment iteration counter by 1
3. Output final cluster centroids $\widetilde{\chi}_{Cs}$ and affiliation matrix U

2.3. MDF-Clustering Target Trajectory Initialisation Algorithm

The effectiveness of MDF-clustering in the scenario of multiple maneuver target initialization in a complex environment is verified by simulating a dataset containing three maneuver targets. The target features contain three numerical attribute variables, including speed, irradiance and contrast, and two qualitative attribute variables, including target position and target silhouette features. Therefore, the variable type $S = 2$ and the set of feature variables is:

$$\chi = \{X_1, X_2, X_3, X_4, X_5\} = \{\chi_1, \chi_2\} \tag{25}$$

where $\chi_1 = \{X_1, X_2, X_3\}$ and $\chi_2 = \{X_4, X_5\}$.

3. Scenario Setting and Experimental Metrics

3.1. MDF-Clustering Simulation Experiments in Complex Environments

The testing scenarios were designed to closely reflect real-world conditions, ensuring the relevance of the cases for practical applications. For instance, scenarios with missed alarms, trajectory crossover, and cloud clutter were chosen because they represent typical challenges in infrared target tracking environments. The presence of clutter, in particular, creates complex noise conditions that are common in satellite-based tracking systems. By simulating various motion patterns, such as linear, curvilinear, and variable-speed movements, we ensured that the test cases reflect a broad spectrum of potential target behaviors in realistic tracking scenarios. This design allows for an accurate assessment of the proposed algorithm's robustness and reliability under varying conditions, particularly in high-clutter environments.

The effectiveness of the MDF-clustering algorithm is evaluated in three distinct simulation scenarios: (1) target leakage, where some alarms fail to detect targets; (2) trajectory crossover, where multiple target trajectories intersect, making differentiation difficult; and (3) cloud clutter, where background noise mimics the target features, complicating the initialization process.

In each scenario, the algorithm is tasked with initializing target trajectories based on a combination of numerical and qualitative features. The point trace data collected from these simulations serves as the input for MDF-clustering, which then assigns affiliation degrees to the detected points. The goal is to verify whether the algorithm can successfully initialize the target trajectory even in the presence of significant noise and clutter. The objective is to evaluate the algorithm's ability to successfully initialize target trajectories, particularly in the presence of significant noise and clutter, and to quantify its performance in such challenging environments.

In order to verify the robustness of the algorithms, we have compared the methods proposed in this paper with classical methods (NN, Slide window, etc.). All the above algorithms were implemented on a PC (I7-2.8 GHz, I7-8700, 8 GB RAM, MSI, Beijing, China) using Visual Studio 2019.

After applying simple filtering, 20 sets of point-trajectory data were randomly chosen for validation in each scenario, resulting in a total of 60 datasets. The marginal distance and symbolic data difference metric were utilized as distance calculation methods within the clustering cluster for the two types of variables. The integrated distance was calculated using Equations (11) and (13) to determine the affiliation results of the point traces. Subsequently, the authenticity of the target was assessed based on the affiliation results.

Additionally, statistical weights of different variable types in the three scenarios were computed using Formula (14), and the statistical information was recorded in Tables 1–3, respectively.

The figure displays all the point traces and trajectories following the initialization of each algorithm. Figure 1a illustrates a dataset in the simulation scene with trajectory intersection when there are 3 targets, consisting of 60 frames of point trace data and one frame. Figure 1b–f depict the sliding window method, the Hough transform method, the intuition method, the nearest neighbor method, and the initialization results of this paper for this particular set of simulation data in this scene.

Table 1. MDF-clustering assigns average weights under target leakage alarms.

Variable Type	Variable	Distance Method	Weights
Number	velocity spoke brightness contrast coordinates	Marginal distance Euclidean distance	0.734
symbolic variable	Contour information	Symbolic data discrepancies	0.266

Table 2. MDF-clustering under trajectory crossover assigns average weights.

Variable Type	Variable	Distance Method	Weights
Number	velocity spoke brightness contrast coordinates	Marginal distance Euclidean distance	0.775
symbolic variable	Contour information	Symbolic data discrepancies	0.225

Table 3. MDF-clustering assigns average weights in the context of cloud clutter.

Variable Type	Variable	Distance Method	Weights
Number	velocity spoke brightness contrast	Marginal distance	0.517
	coordinates	Euclidean distance	
symbolic variable	Contour information	Symbolic data discrepancies	0.483

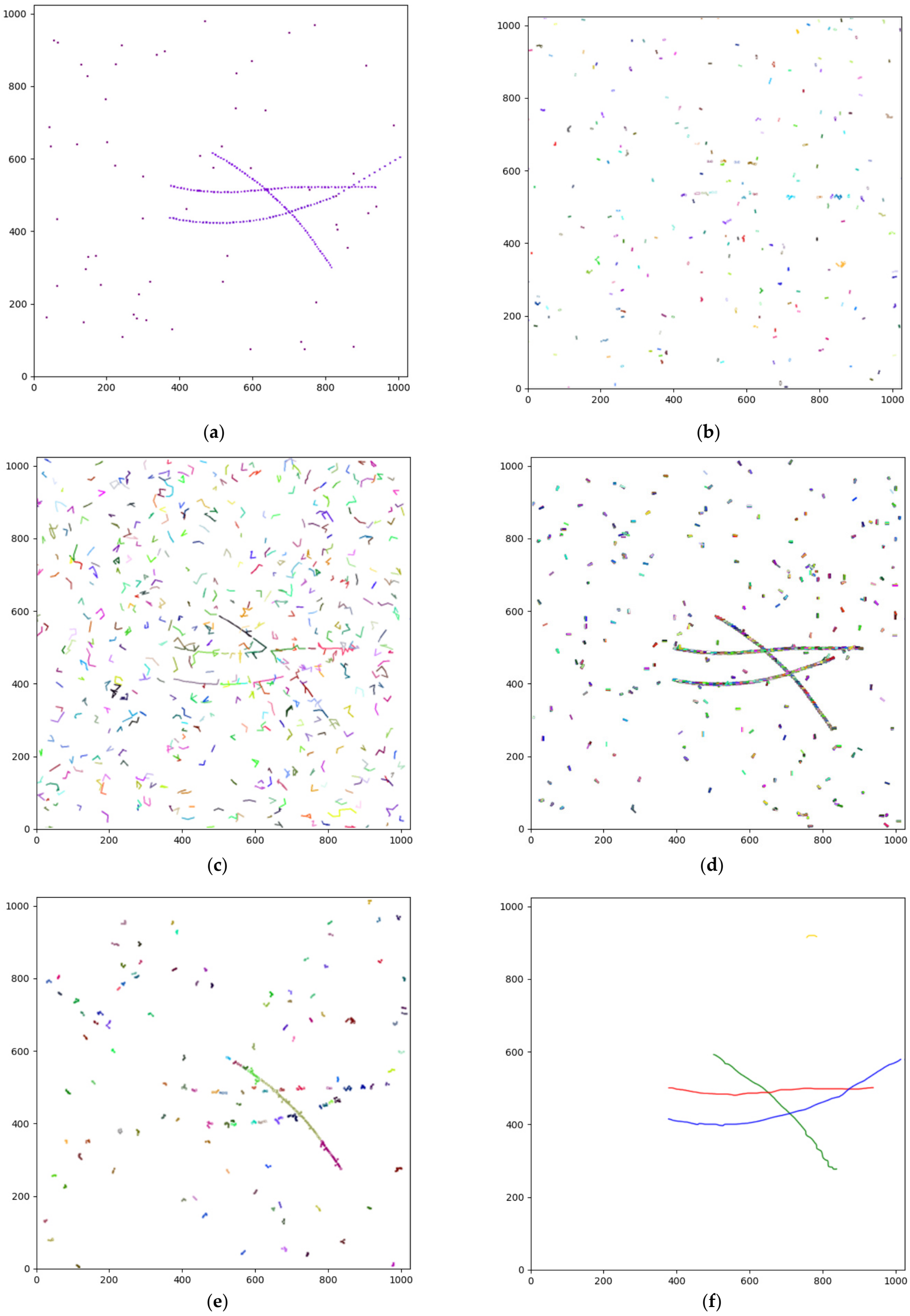

Figure 1. Simulation of point trace diagrams and starting batch results for each algorithm. (**a**) Ingle-frame noise and track traces, (**b**) Sliding window method, (**c**) Hough transform method, (**d**) Intuitive method, (**e**) Nearest neighbor method, (**f**) Methodology of the paper. (Each colour represents an associated trajectory).

3.2. MDF-Clustering Algorithm Evaluation Index

The evaluation of the algorithm's effectiveness involves comparing the results of the other four algorithms in three scenarios with different false alarm rates using three evaluation methods.

(1) The average number of frames required for target stabilization initialization

One of the evaluation methods employed to assess the algorithm's performance involves calculating the average number of frames required for target stabilization and initialization. This metric is crucial as it reflects the efficiency of the target tracking algorithm by indicating the time needed for the algorithm to achieve a stable state from the initialization of tracking. A lower value suggests that the algorithm can quickly and accurately lock on and track the target stably, demonstrating the responsiveness of the algorithm. The average number of frames for stable initialization is calculated using the following formula:

$$\overline{f_s} = \frac{\sum_{q=3}^{Q} \overline{f_{sq}}}{Q} \tag{26}$$

(2) Average occupancy rate of false trajectories

The average false track occupancy is a metric used to assess the ratio of false tracks detected to all tracks detected by the target tracking algorithm. A lower value suggests that the algorithm is successful in differentiating between actual targets and noise, resulting in a lower false alarm rate and ensuring the accuracy and reliability of the generated results. The formula for calculating this metric is as follows:

$$F_{Pq} \triangleq \frac{\frac{\sum_{i=1}^{N} F_i}{\sum_{i=1}^{N} R_i}}{N} \tag{27}$$

The false trajectory occupancy rate F_{Pq} is calculated under various false alarm rates q, with F_i representing the false trajectory and R_i representing the actual trajectory of simulations at a specific false alarm rate. The variable N denotes the total number of simulation runs at this false alarm rate.

$$F_P \triangleq \frac{\sum_{q=3}^{Q} F_{Pq}}{Q-3} \tag{28}$$

F_P represents the average false track occupancy and Q represents the maximum false alarm rate.

(3) Average Initialization Rate of Correct Trajectory

The average initialization rate of correct trajectory measures the ability of the target tracking algorithm to accurately identify and start tracking the real target at the initial stage. This index reflects the algorithm's capability to identify and initiate tracking of the actual target. A higher value indicates a more accurate tracking of the target in the initialization phase. The formula for this is:

$$D_{Pq} \triangleq \frac{\frac{\sum_{i=1}^{N} T_i}{T_{All}}}{N} \tag{29}$$

D_{Pq} represents the correct trajectory initialisation rate at different false alarm rates q, T_i, T_{All} represent the number of accurate trajectories initialized and the total number of actual target trajectories for the ith simulation at a given false alarm rate, respectively, and N represents the total number of times the simulation has been performed at that false alarm rate.

$$D_P \triangleq \frac{\sum_{q=3}^{Q} D_{Pq}}{Q-3} \tag{30}$$

D_P represents the average false track occupancy and Q represents the maximum false alarm rate.

(4) Trajectory leakage rate

The miss rate is the proportion of targets that should be detected by the system but are not detected. This metric reflects the tracking system's inability to detect targets, and its formula is as follows:

$$R_m = \frac{N_{non\_det}}{N_{all_{tar}}} \tag{31}$$

R_m represents the missed batch rate of the trajectory, N_{non_det} represents the number of undetected targets, and N_{all_tar} represents the total number of valid targets.

4. Analysis of Experimental Results

4.1. Simulation Scene Setting

This study investigates the effectiveness of using MDF-clustering for initializing maneuvering target trajectories in complex environments. The proposed method is compared with the nearest neighbor method [23], the Hough transform method [24,25], the sliding window method [26], and the intuitive method [27].

The Nearest Neighbor (NN) method is a classical approach that has been extensively applied in multi-target tracking due to its computational simplicity and efficiency in environments with low clutter. However, its performance deteriorates significantly in high-noise conditions, as it lacks the inherent ability to discriminate between targets and environmental clutter. Similarly, the Hough Transform is recognized for its robustness in detecting straight-line trajectories, yet it encounters difficulties when tracking targets with complex motion dynamics, such as curvilinear or highly maneuverable behaviors. The Sliding Window method, though effective in handling short-term fluctuations in data, fails to sustain accuracy when tracking targets over longer durations, particularly in scenarios with trajectory crossovers or dense clutter. These methods were chosen as baselines in this study because they represent well-established, standard approaches in the field of target tracking, offering a comprehensive framework for evaluating the performance of the proposed MDF-clustering algorithm. By comparing against these traditional methods, the enhancements of the MDF-clustering approach in terms of accuracy, robustness, and adaptability in cluttered and dynamic environments are effectively demonstrated.

The scene construction for space-based infrared sensor observation and tracking scenarios of airborne targets is conducted using STK 12.1 simulation software. The sensor platform's flight altitude is set at 1000 km, and three targets are placed within the scene based on aerial target motion characteristics [28] and scene characteristics [29]. The initial speeds of the targets in different simulation groups are all 300 m/s, exhibiting various motion patterns such as uniform linear motion, uniform curvilinear motion, uniformly accelerated linear motion, uniformly accelerated curvilinear motion, variable-speed linear motion, and variable-speed curvilinear motion. The infrared sensor's sampling frequency is 0.1 s, with a spatial resolution of 25 m. Three scenarios are simulated: target existence with missed alarms, target trajectory existence with crossover, and target presence in cloud clutter background. Poisson distribution [30] is used to add clutter to the sensor output, with data ranging from 3% to 11% based on the false alarm rate. Each scenario comprises 30 datasets per false alarm rate.

4.2. Evaluation Results

To verify the statistical significance of the observed improvements, we calculated confidence intervals for the performance metrics (e.g., average initialization frames, false trajectory occupancy rates). We ran 60 iterations for each dataset and averaged the results to minimize random variations. The improvements in the MDF-clustering algorithm's performance were consistent across iterations, with 95% confidence intervals demonstrating that the differences are statistically significant.

(1) Average number of frames required for target stabilization initialization

The number of stable initialization frames for each algorithm across different false alarm rates was counted, and the average stable initialization frames for each scenario were calculated. The results are shown in the following figures.

Tables 4–6 and Figures 2–4 depict the average number of frames required for stable initialization by each algorithm at varying false alarm rates across three scenarios. The performance comparison reveals a consistent trend in the image curves. The MDF-clustering algorithm outperforms other algorithms in all scenarios and false alarm rates, reducing the number of stabilized initialized frames by 2.14, 0.87, 4.15, and 9.07 frames compared to the nearest-neighbor, Hough transform, intuition, and sliding-window methods, respectively. Notably, as the false alarm rate increases, this algorithm shows a more moderate growth rate in the average number of frames for stable initialization at 0.013, whereas other algorithms experience growth rates of 0.035, 0.039, 0.028, and 0.030, indicating a higher level of robustness and adaptability of the MDF-clustering algorithm.

Table 4. The average number of stable starting frames for each algorithm with different false alarm rates in target leakage scenarios.

False Alarm Rate	MDF Clustering	NN	Hough Transform	Intuitivenotion	Sliding Window
3%	3	3.4	3.15	4.9	8.05
4%	3	3.55	3.15	5	8.15
5%	3	3.65	3.2	5.1	8.45
6%	3	3.85	3.25	5.3	8.55
7%	3	4.05	3.35	5.6	9.95
8%	3.15	4.25	3.4	5.8	10.6
9%	3.2	4.55	3.6	6.05	10.9
10%	3.25	4.75	3.75	6.3	11.15
11%	3.3	4.95	3.95	6.55	11.6

Table 5. The average number of stabilized starting frames for each algorithm with different false alarm rates in the trajectory crossing scenarios.

False Alarm Rate	MDF Clustering	NN	Hough Transform	Intuitivenotion	Sliding Window
3%	3.1	4.25	3.6	5.2	8.45
4%	3.15	4.3	3.65	5.35	8.6
5%	3.1	4.45	3.75	5.45	8.75
6%	3.2	4.55	3.85	5.6	8.95
7%	3.25	4.75	4	5.7	9.2
8%	3.35	5.1	4.2	5.9	9.35
9%	3.4	5.3	4.35	6.2	9.55
10%	3.4	5.4	4.55	6.35	9.7
11%	3.45	5.7	4.75	6.5	10.2

Table 6. Stabilized starting batch average frames for each algorithm with different false alarm rates in cloud clutter backgrounds.

False Alarm Rate	MDF Clustering	NN	Hough Transform	Intuitivenotion	Sliding Window
3%	3	4.55	3.3	6.4	10.4
4%	3	4.7	3.4	6.55	10.75
5%	3	4.8	3.5	6.85	10.9
6%	3	5.05	3.6	7.05	11.35
7%	3.05	5.15	3.9	7.25	11.95
8%	3.1	5.3	4.05	7.35	12.5
9%	3.2	5.55	4.3	7.55	13
10%	3.25	5.7	4.55	7.7	13.45
11%	3	4.55	3.3	6.4	10.4

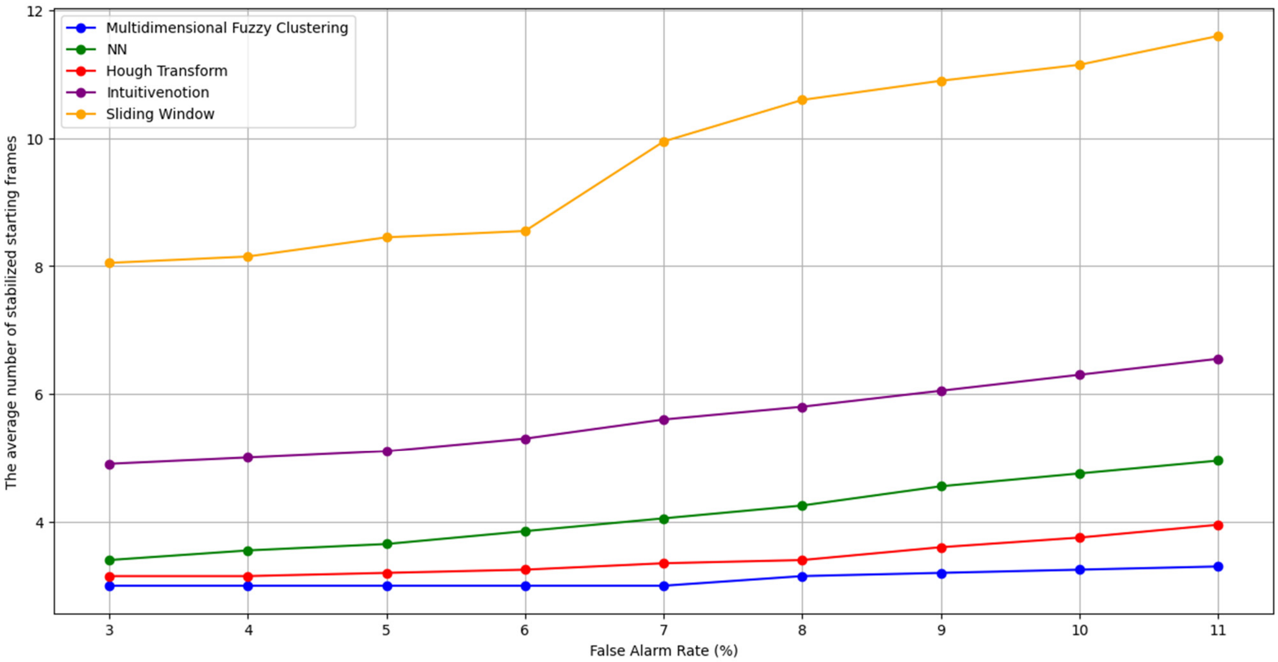

Figure 2. The average number of stable starting frames for each algorithm with different false alarm rates in target leakage scenarios.

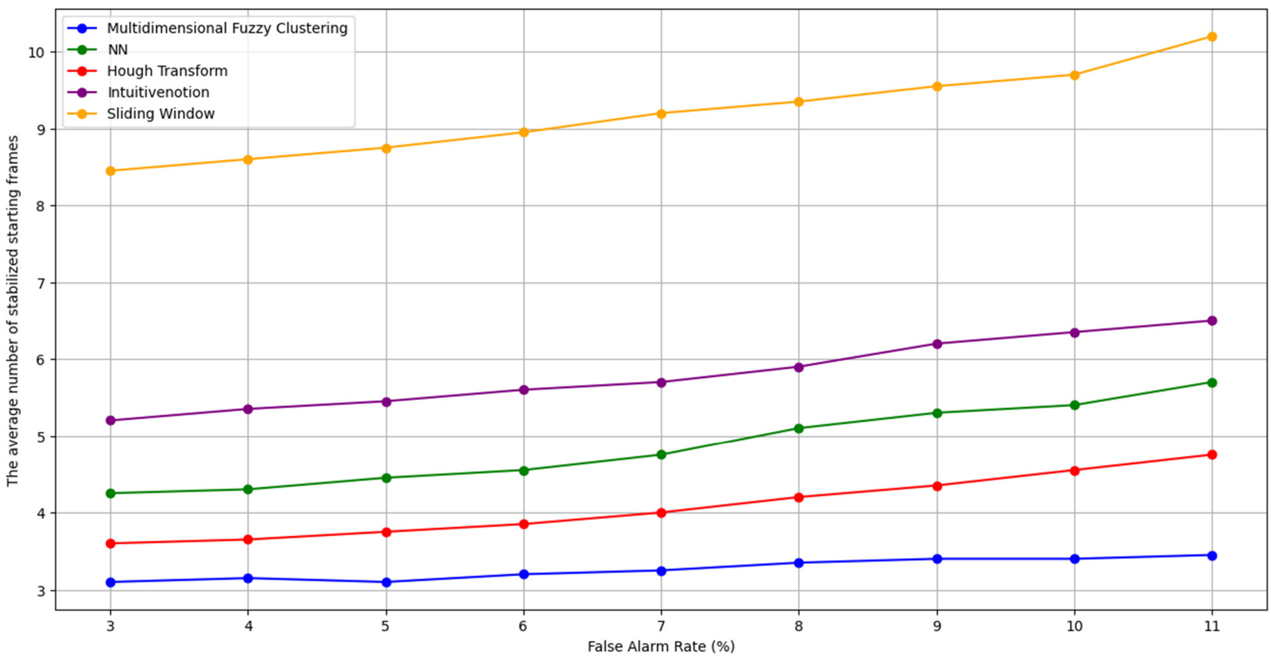

Figure 3. The average number of stabilized starting frames for each algorithm with different false alarm rates in the trajectory crossing scenarios.

(2) Average occupancy rate of false trajectories

By counting the occupancy rate of false trajectories in each scenario and calculating the average occupancy rate of false trajectories for different algorithms in each scenario, the following figures are derived.

Tables 7–9 and Figures 5–7 display the average false trajectory occupancy of each algorithm across three scenarios with varying false alarm rates. Through comparative analysis, the MDF-clustering algorithm exhibits the lowest false trajectory occupancy rates of 0.04%, 0.05%, and 0.04% in all scenarios, showcasing its superior performance in mitigating false initializations. On the other hand, while the nearest-neighbor, Hough transform, and sliding-window methods show moderate effectiveness in specific scenarios, their ability to suppress false trajectories diminishes as the false alarm rate increases. Hence, the MDF-clustering algorithm outperforms other methods in terms of tracking accuracy, especially in complex dynamic environments.

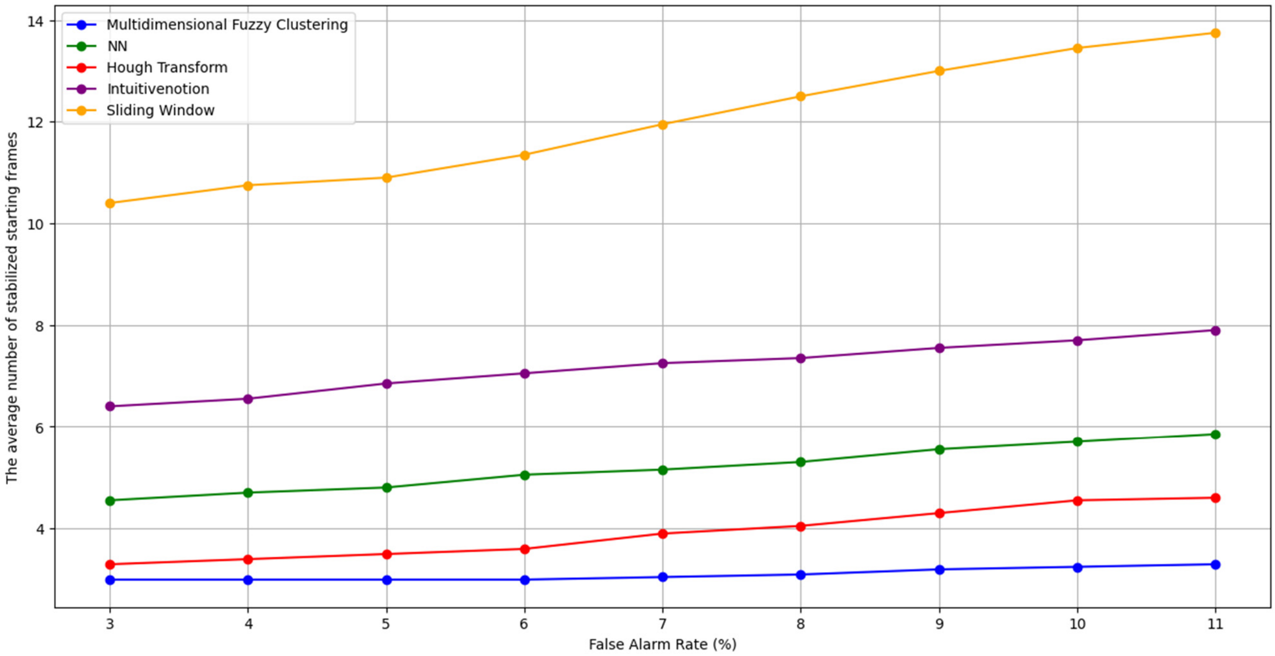

Figure 4. Stabilized starting batch average frames for each algorithm with different false alarm rates in cloud clutter backgrounds.

Table 7. Average false trajectory occupancy by algorithms with different false alarm rates in target leakage scenarios.

False Alarm Rate	MDF Clustering	NN	Hough Transform	Intuitivenotion	Sliding Window
3%	0.00%	63.54%	51.36%	78.29%	85.44%
4%	0.00%	63.82%	51.42%	78.46%	85.87%
5%	0.00%	64.03%	52.17%	80.20%	86.91%
6%	0.00%	65.10%	53.44%	79.93%	87.26%
7%	0.00%	65.83%	54.26%	81.63%	87.84%
8%	0.16%	66.36%	56.47%	81.59%	89.15%
9%	0.07%	67.88%	57.19%	83.01%	89.96%
10%	0.14%	68.96%	57.30%	83.77%	90.10%
11%	0.17%	69.22%	58.76%	84.56%	90.82%

Table 8. Average false trajectory occupancy of each algorithm with different false alarm rates in trajectory intersection scenarios.

False Alarm Rate	MDF Clustering	NN	Hough Transform	Intuitivenotion	Sliding Window
3%	0.00%	76.22%	42.92%	68.33%	80.35%
4%	0.00%	77.35%	43.11%	68.73%	81.66%
5%	0.00%	77.54%	43.52%	69.32%	82.62%
6%	0.00%	78.16%	43.90%	69.78%	83.65%
7%	0.00%	78.57%	44.22%	70.35%	84.27%
8%	0.05%	79.20%	44.28%	70.82%	85.16%
9%	0.07%	79.50%	44.98%	71.16%	85.98%
10%	0.09%	80.32%	45.27%	71.84%	86.27%
11%	0.14%	80.43%	45.83%	72.22%	87.43%

Table 9. Average false track occupancy of each algorithm with different false alarm rates in the cloud context.

False Alarm Rate	MDF Clustering	NN	Hough Transform	Intuitivenotion	Sliding Window
3%	0.00%	84.52%	54.73%	78.63%	95.35%
4%	0.00%	84.64%	55.36%	79.85%	95.48%
5%	0.00%	85.34%	57.16%	80.21%	95.83%
6%	0.00%	86.47%	59.63%	82.15%	95.92%
7%	0.00%	87.34%	60.12%	84.28%	96.03%
8%	0.00%	88.62%	60.94%	85.44%	96.31%
9%	0.06%	89.24%	61.79%	86.93%	96.58%
10%	0.14%	89.86%	63.28%	88.35%	96.75%
11%	0.16%	90.04%	64.88%	89.65%	97.01%

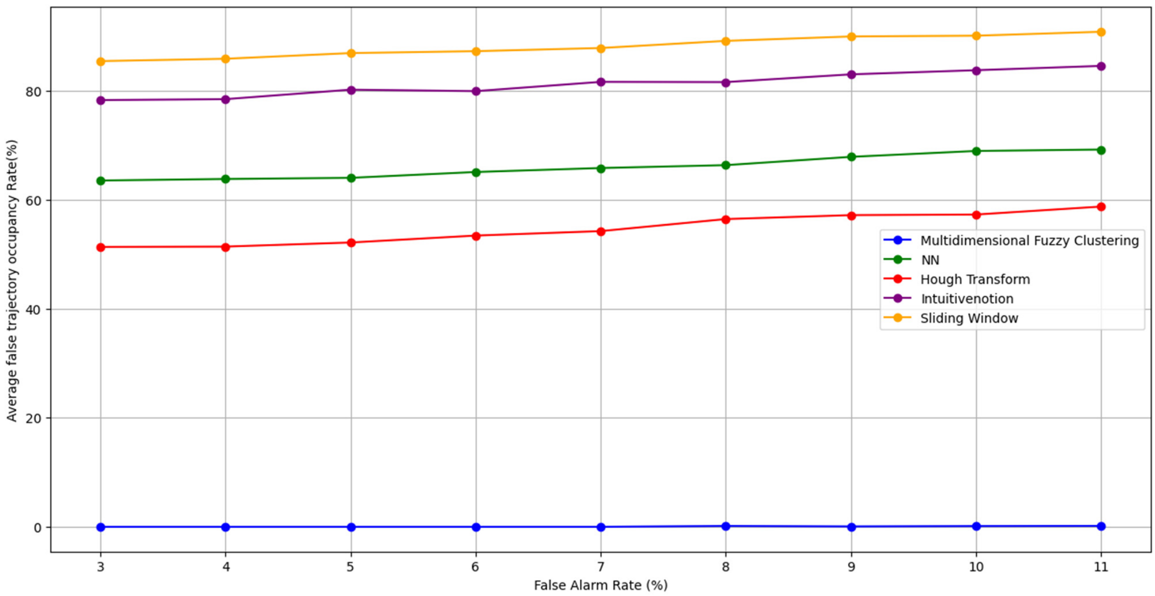

Figure 5. Average false trajectory occupancy by algorithms with different false alarm rates in target leakage scenarios.

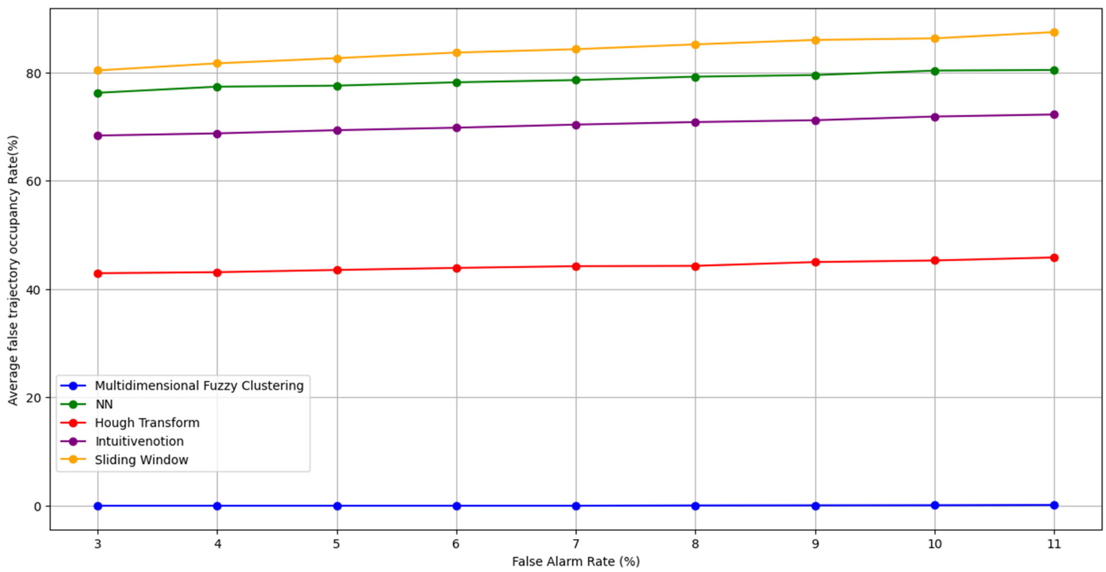

Figure 6. Average false trajectory occupancy of each algorithm with different false alarm rates in trajectory intersection scenarios.

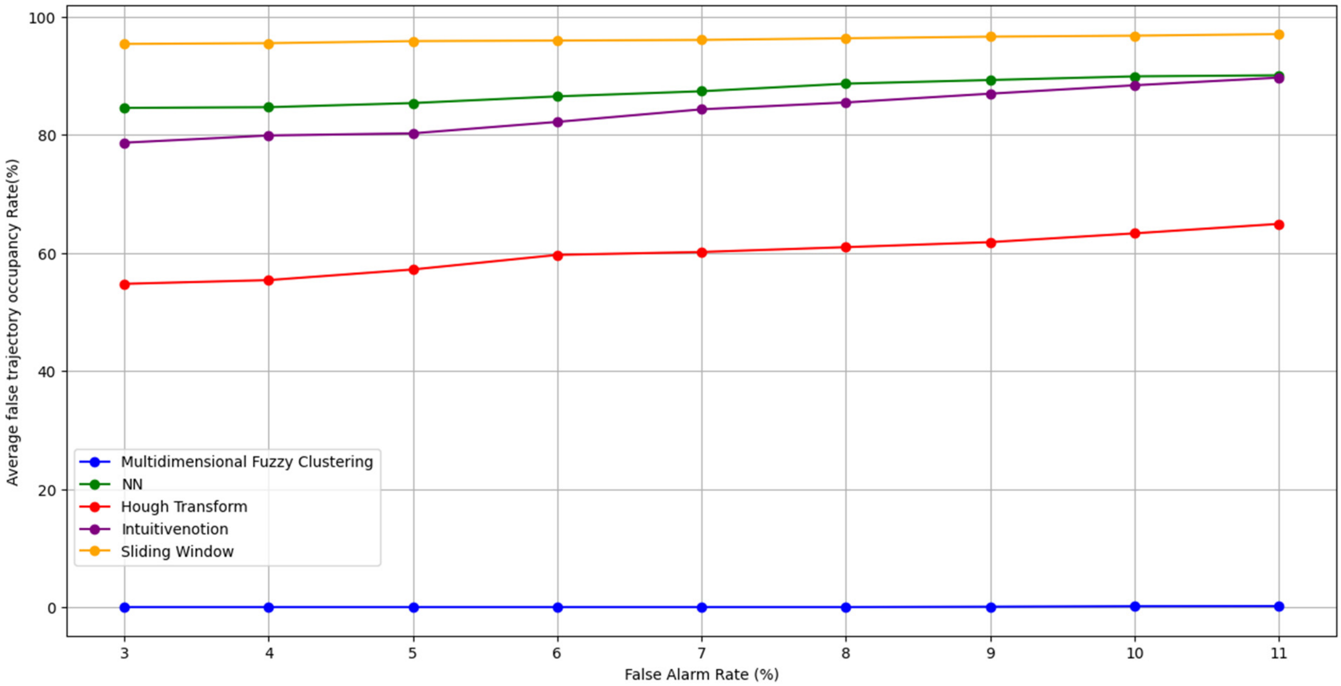

Figure 7. Average false track occupancy of each algorithm with different false alarm rates in the cloud context.

(3) Average initialization rate of correct trajectories

Statistics on the average initialization rate of correct trajectories in each scenario and calculation of the average initialization rate of correct trajectories for different algorithms in each scenario lead to the following figure.

Tables 10–12 and Figures 8–10 display the average initialization rate of each algorithm for correct trajectories across three different scenarios with varying false alarm rates. Upon examining the data presented in the figures, it is evident that the MDF-clustering algorithm maintains an average initialization rate of correct trajectories at 99.63%, 99.59%, and 99.53% in the respective scenarios. This algorithm consistently achieves a high correct initialization rate under low and medium false alarm rates, with only marginal decreases of 0.21%, 0.22%, and 0.22% as the false alarm rate increases. In contrast, when compared to other algorithms, the average initialization rate experiences reductions of 4.76%, 3.71%, 5.32%, and 7.48%. The notable finding is the significantly smaller decrease in the correct initialization rate of the MDF-clustering algorithm in comparison to other methods, particularly evident in visually complex cloud backgrounds. Consequently, the MDF-clustering algorithm demonstrates superior performance in correct tracking relative to alternative approaches.

Table 10. Average starting batch rate of correct trajectories for each algorithm with different false alarm rates for target leakage scenarios.

False Alarm Rate	MDF Clustering	NN	Hough Transform	Intuitivenotion	Sliding Window
3%	100.00%	95.00%	98.33%	96.67%	83.33%
4%	100.00%	93.33%	96.67%	95.00%	80.00%
5%	100.00%	91.67%	95.00%	93.33%	78.33%
6%	100.00%	90.00%	93.33%	91.67%	73.33%
7%	100.00%	86.66%	91.67%	86.66%	75.00%
8%	100.00%	83.33%	86.66%	80.00%	60.00%
9%	98.33%	73.33%	81.66%	73.33%	56.66%
10%	98.33%	73.33%	81.66%	73.33%	56.66%
11%	98.33%	68.33%	78.33%	64.99%	48.33%

Table 11. Average starting batch rate of correct trajectories for different algorithms with different false alarm rates in trajectory intersection scenarios.

False Alarm Rate	MDF Clustering	NN	Hough Transform	Intuitivenotion	Sliding Window
3%	100.00%	96.67%	98.33%	93.33%	88.33%
4%	100.00%	95.00%	96.67%	90.00%	83.33%
5%	100.00%	93.33%	93.33%	86.66%	80.00%
6%	100.00%	90.00%	91.67%	85.00%	76.66%
7%	100.00%	83.33%	86.66%	81.66%	66.66%
8%	100.00%	78.33%	83.33%	78.33%	60.00%
9%	100.00%	75.00%	81.66%	75.00%	58.33%
10%	98.33%	71.66%	78.33%	71.66%	55.00%
11%	98.33%	66.66%	76.66%	65.00%	50.00%

Table 12. Average batch rate of correct trajectories for different algorithms with different false alarm rates in the cloud context.

False Alarm Rate	MDF Clustering	NN	Hough Transform	Intuitivenotion	Sliding Window
3%	100.00%	96.67%	98.33%	93.33%	88.33%
4%	100.00%	95.00%	96.67%	90.00%	83.33%
5%	100.00%	93.33%	93.33%	86.66%	80.00%
6%	100.00%	90.00%	91.67%	85.00%	76.66%
7%	100.00%	83.33%	86.66%	81.66%	66.66%
8%	100.00%	78.33%	83.33%	78.33%	60.00%
9%	100.00%	75.00%	81.66%	75.00%	58.33%
10%	98.33%	71.66%	78.33%	71.66%	55.00%
11%	98.33%	66.66%	76.66%	65.00%	50.00%

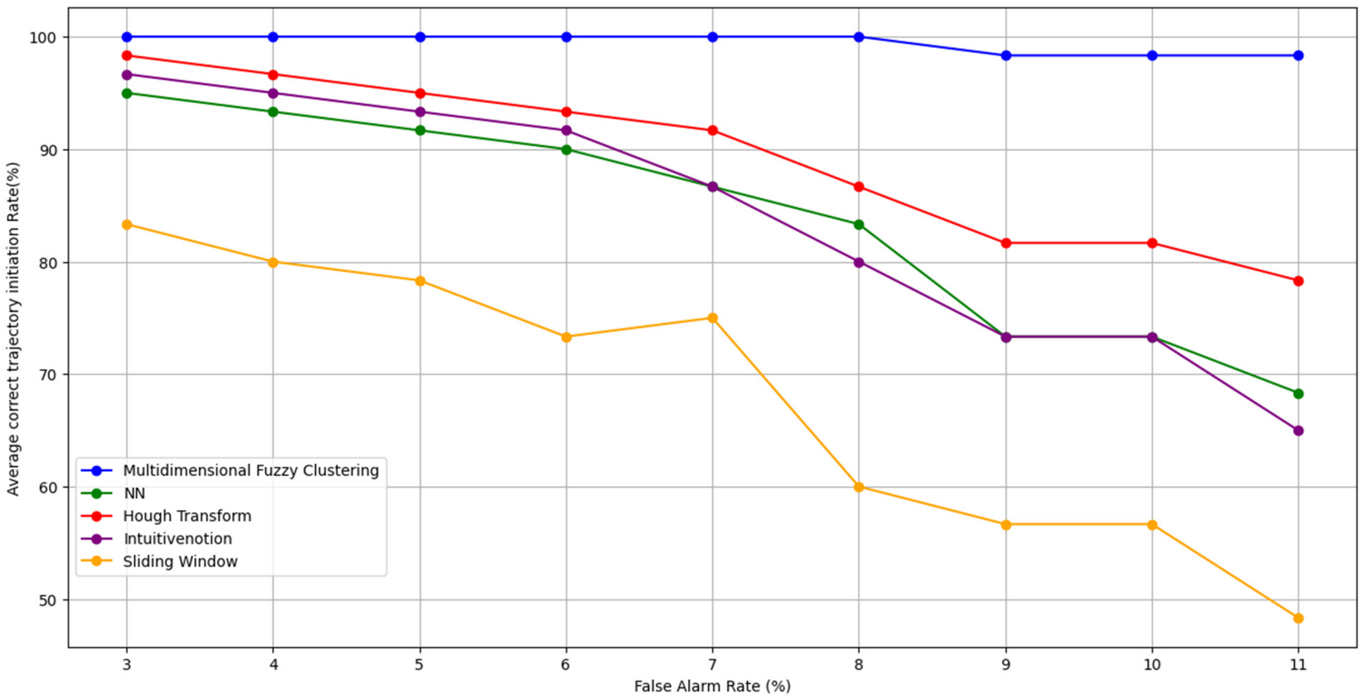

Figure 8. Average starting batch rate of correct trajectories for each algorithm with different false alarm rates for target leakage scenarios.

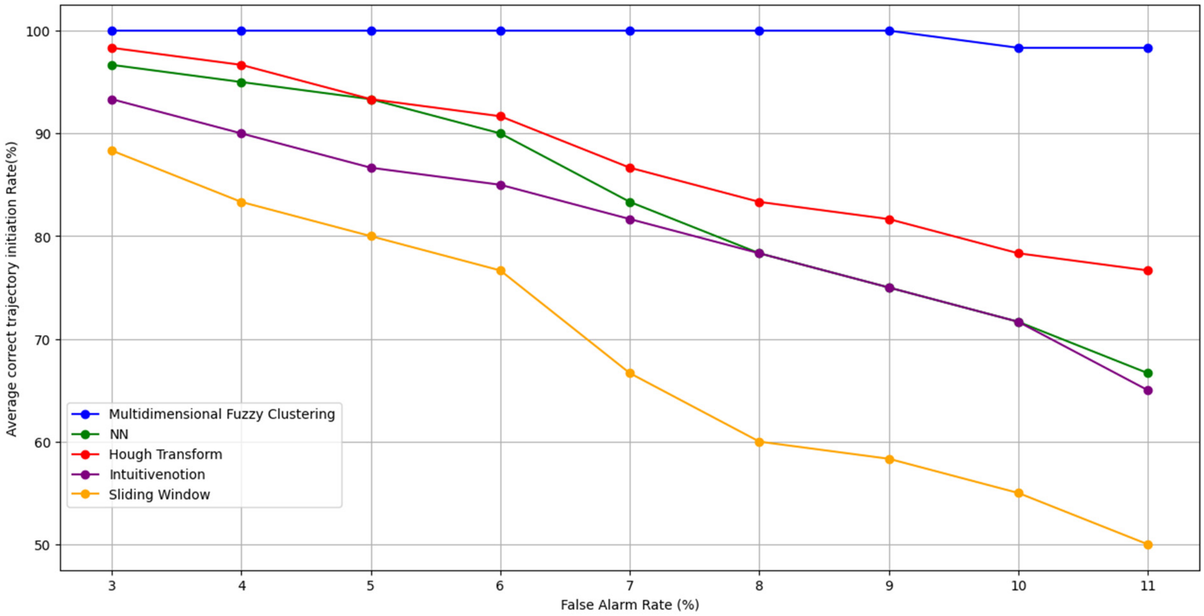

Figure 9. Average starting batch rate of correct trajectories for different algorithms with different false alarm rates in trajectory intersection scenarios.

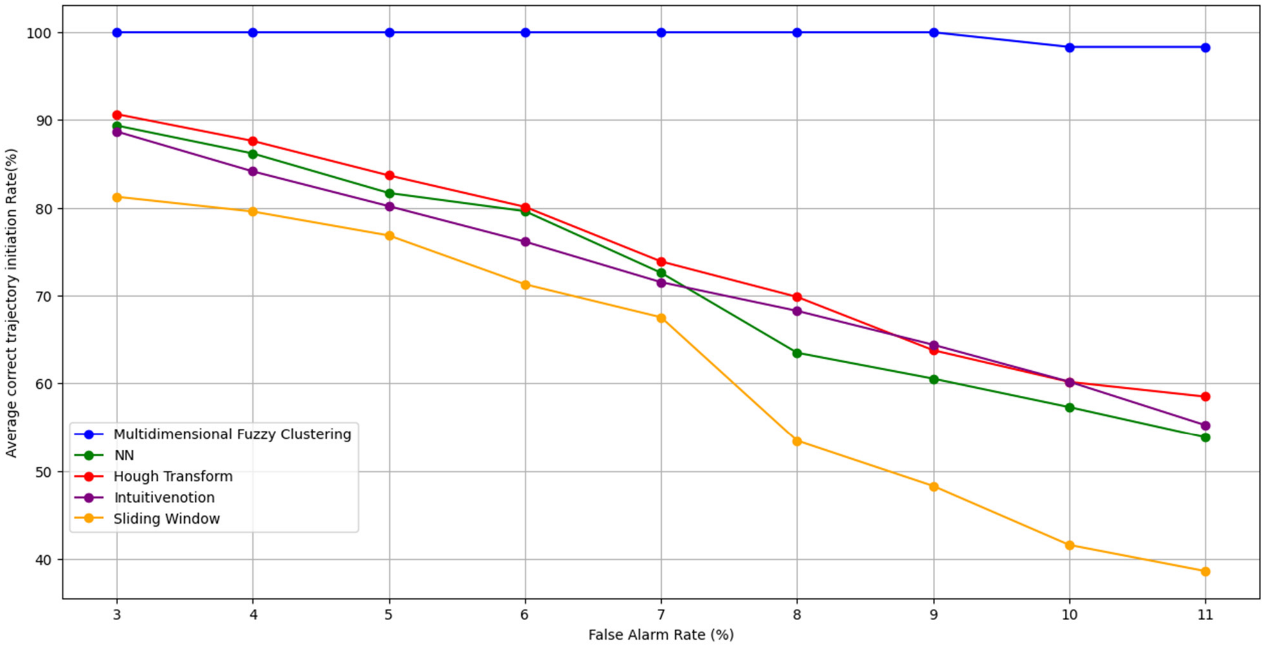

Figure 10. Average batch rate of correct trajectories for different algorithms with different false alarm rates in the cloud context.

(4) Trajectory leakage rate

The statistics of the trajectory leakage rate in each scenario and the calculation of different algorithmic trajectory leakage rates in each scenario result in the following figures.

Figures 11–13 depict the leakage rates of each algorithm across three distinct scenarios with varying false alarm rates. The performance evaluation highlights notable differences in the MDF-clustering algorithm's performance across all test scenarios. Specifically, in the leakage scenario, this algorithm consistently maintains a low leakage rate across different false alarm rates, only slightly increasing to 1.67% at the highest false alarm rate. Conversely, other algorithms demonstrate a significant increase in leakage rates as the false alarm rate rises, with the sliding-window method peaking at a leakage rate of 58.34% at an 11% false alarm rate. In the trajectory intersection scenario, the MDF-clustering algorithm also shows minimal miss-approval rates, remaining at 0% overall. In contrast, the Hough transform and sliding window algorithms exhibit leakage rates of 21.66% and 45.00%, respectively, under the same false alarm rate condition, posing challenges in the trajectory intersection scenario. Lastly, in the cloud background scenario, the MDF-clustering algorithm maintains a consistently low leakage rate. In comparison, the other algorithms display average leakage rates ranging from 1.85% to 35.55% when compared to the algorithm proposed in this paper under similar conditions.

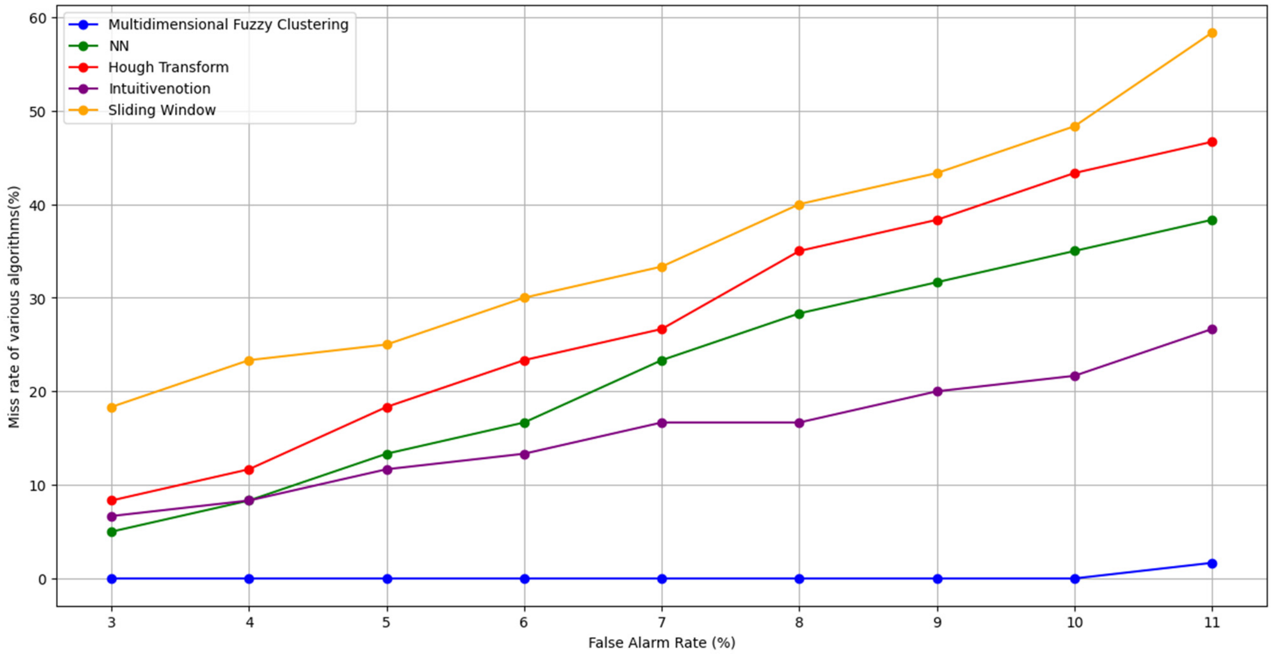

Figure 11. Miss rate of various algorithms with different false alarm rates in target missed detection scenarios.

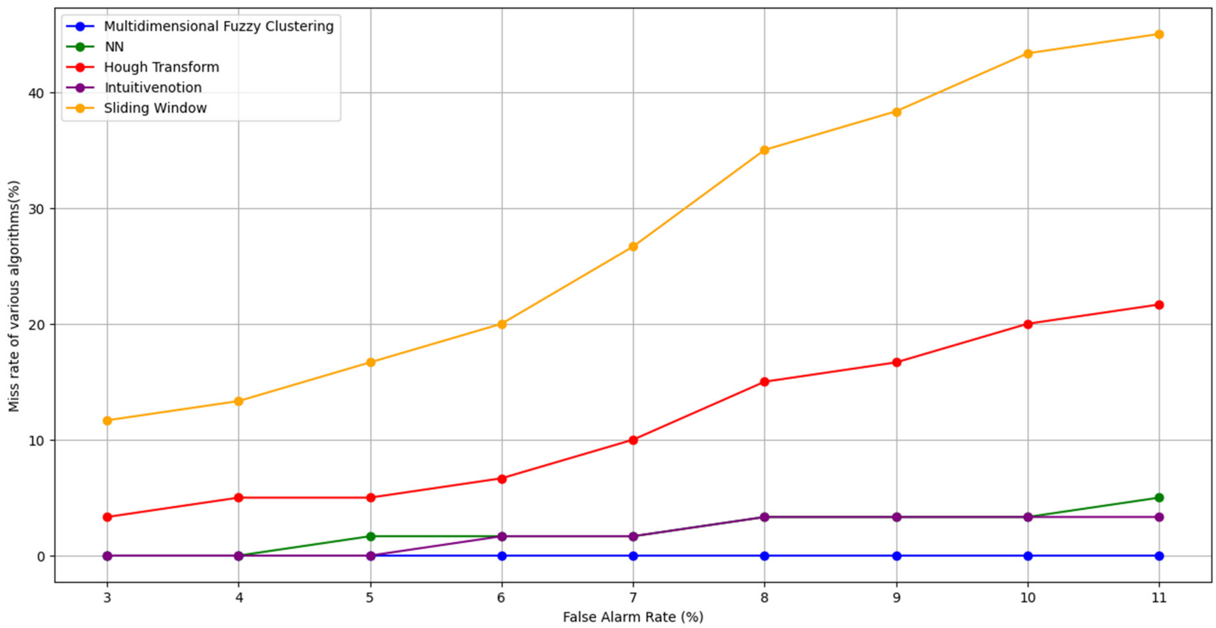

Figure 12. Miss rate of various algorithms with different false alarm rates in target crossover scenarios.

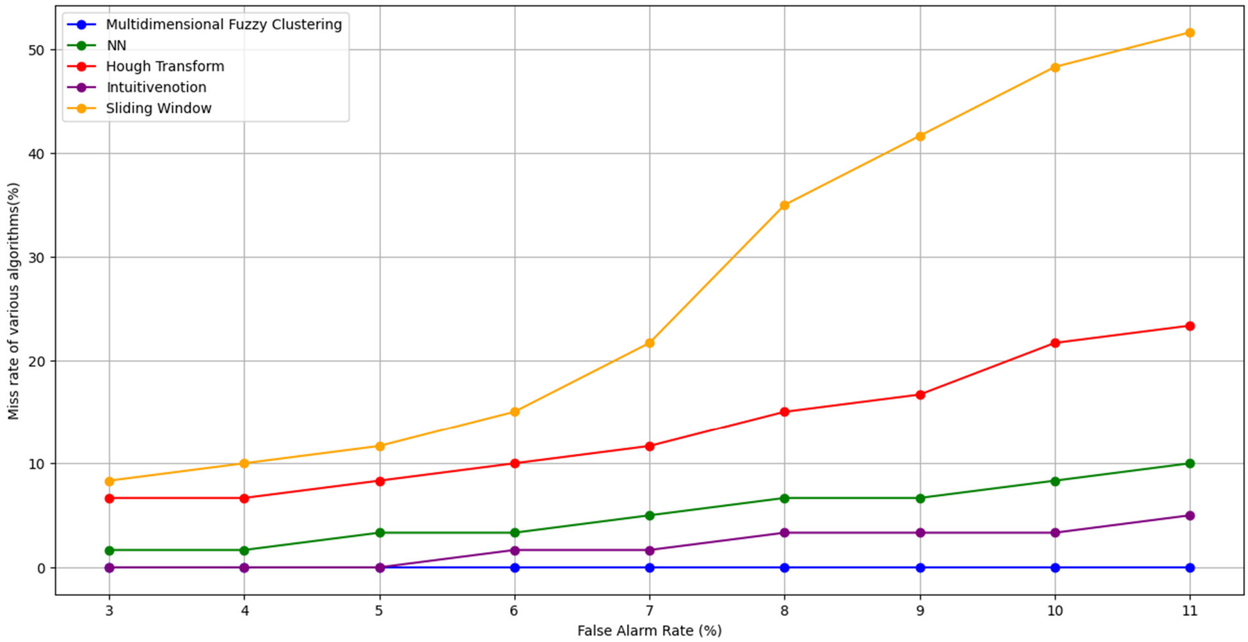

Figure 13. Miss rate of various algorithms with different false alarm rates in cloud background scenarios.

In the experimental comparison of performance evaluation of trajectory initialization algorithms, the study analyzed and compared the performance of the MDF-clustering algorithm with several other commonly used algorithms in the context of target tracking. The analysis included handling false trajectories, maintaining accurate initialization rates, initializing frames, and controlling the missed batch rate, showcasing significant advantages in accuracy and robustness. Target tracking in complex dynamic environments requires maintaining a high initialization rate and a low false alarm rate while also ensuring accuracy and stability in monitoring.

5. Conclusions

A novel multidimensional fuzzy clustering algorithm (MDF-clustering) is proposed for initializing point target trajectories in complex environments. The algorithm effectively determines the trajectory's affiliation degree by extracting the multidimensional features of the target, integrating the clustering center and weight assignment, and subsequently updating the clustering center and weights. Experimental results demonstrate that this method significantly enhances the accuracy and robustness of trajectory initialization when compared to traditional approaches in environments characterized by high clutter and target maneuvers. This improvement is evidenced by enhancements in the average number of stable initialization frames, the rate of correct trajectory initialization, and a minimal false trajectory occupancy rate. Consequently, the MDF-clustering algorithm elevates the performance of trajectory initialization, thereby improving the overall efficiency and reliability of the tracking system. This advancement holds substantial practical significance for target tracking in complex environments.

While the MDF-clustering algorithm achieves notable improvements, several limitations remain. One key limitation is its sensitivity to the initialization of clustering centers, which could impact performance in highly dynamic environments with varying clutter densities. Additionally, although the algorithm integrates an adaptive clustering mechanism, further enhancements may be needed to better handle extreme clutter conditions or highly dense target environments.

Moreover, when applying the algorithm to larger datasets or real-time systems, there could be some challenges. In particular, the computational complexity may result in a slight increase in processing time, leading to occasional delays in real-time applications. However, this delay is not expected to be overly significant but may cause minor latency in systems that require very rapid updates. Similarly, while the memory usage remains efficient for moderate datasets, processing large-scale multi-dimensional data could result in increased memory consumption, particularly in environments with limited computational resources. Though these impacts are not critical, they warrant consideration when deploying the algorithm in real-time, high-demand environments. Future research could explore optimizing the algorithm to address these challenges and ensure better performance in such scenarios.

For future research, we propose the following directions:

1. Real-time system optimization: While the algorithm shows strong performance in experimental conditions, further optimization is needed to enhance its efficiency for real-time applications, particularly in scenarios involving fast-moving, closely spaced targets.
2. Improving robustness under dense clutter: We plan to explore techniques that further reduce false alarm rates in environments where the density of noise or clutter is exceptionally high.
3. Advanced error handling in multi-target scenarios: Extending the algorithm's capability to handle more complex multi-target tracking problems, where targets frequently cross paths or exhibit highly erratic movement, will be a focus of future work.
4. Integration of KAN for improved trajectory initialization: KAN, which integrates domain knowledge with data-driven models, has shown great potential in predictive modeling for complex physical systems, including recent advancements in human–robot interaction. Building on recent advancements in the KAN method, particularly its successful application in human–robot interaction [31], we aim to explore its integration into trajectory initialization tasks. By leveraging KAN's ability to incorporate domain knowledge with data-driven models, we expect to further enhance the robustness and accuracy of our algorithm, especially in scenarios with high noise and low signal conditions, such as infrared weak target tracking.

These areas of future exploration will help broaden the practical applicability of the MDF-clustering algorithm, ensuring its adaptability and reliability in both military and civilian scenarios.

Author Contributions: Conceptualization, Z.Y. and H.N.; methodology, Z.Y.; software, Z.Y.; validation, Z.Y; formal analysis, Z.Y. and H.N.; investigation, Z.Y.; resources, Z.Y. and Y.L.; data curation, Z.Y. and Y.L.; writing—original draft preparation, Z.Y.; writing—review and editing, Z.Y. and H.N.; visualization, Z.Y.; supervision, H.N. and C.B.; project administration, C.B.; funding acquisition, C.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Civil Space Technology Advance Research Project (grant number D030312).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the first authors upon request.

Acknowledgments: We wish to thank all the project team members.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhao, M.; Zhang, J.; Li, Q.; Yang, J.; Siga, E.; Zhang, T. GAT-ABiGRU Based Prediction Model for AUV Trajectory. *Appl. Sci.* **2024**, *14*, 4184. [CrossRef]
2. Junwei, Z. A Strong Maneuvering Target-Tracking Filtering Based on Intelligent Algorithm. *Int. J. Aerosp. Eng.* **2024**, *2024*, 9981332. [CrossRef]
3. Ji, R.; Liang, Y.; Xu, L.; Zhang, W. A Concave Optimization-Based Approach for Joint Multi-Target Track Initialization. *IEEE Access* **2019**, *7*, 108551–108560. [CrossRef]

4. Yi, W.; Fang, Z.; Li, W.; Hoseinnezhad, R.; Kong, L. Multi-Frame Track-Before-Detect Algorithm for Maneuvering Target Tracking. *IEEE Trans. Veh. Technol.* **2020**, *69*, 4104–4118. [CrossRef]
5. Visina, R.; Bar-Shalom, Y.; Willett, P. Multiple-Model Estimators for Tracking Sharply Maneuvering Ground Targets. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 1404–1414. [CrossRef]
6. Han, W.; Chen, C.; Li, Z. A Method of Track Segment Association in Doppler Blind Zone. *Radar Sci. Technol.* **2020**, *18*, 239–246.
7. Zhai, H. Implementation of Radar Data Processing Based on Low-Small-Slow Targets. Master's Thesis, Nanjing University of Science and Technology, Nanjing, China, 2022.
8. Leung, H.Z.; Blanchette, M. Evaluation of multiple target track initiation techniques in real radar tracking environments. *Radar Sonar Navig. IEE Proc.* **1996**, *143*, 246–254. [CrossRef]
9. Dong, Z.H. On the Method of Trajectory Initiation. *Command. Control. Simul.* **1999**, 1–7.
10. Su, F.; Wang, G.W.; He, Y. Track initiation algorithm based on the modified logic. *Mod. Def. Technol.* **2004**, *32*, 66–68.
11. Xu, H.; Xie, W.C.; Gao, F. Track Initiation of Sea Surface Multi-Targets for Airborne Radar. *Radar Sci. Technol.* **2015**, *13*, 655–659+666.
12. Wang, G.H.; Su, F.; Mao, S.Y. Fast Track Initiation Algorithm in Clutter Environments Based on Hough Transform and Logic. *J. Syst. Simul.* **2002**, *14*, 874–876.
13. Jin, Z.; Liu, G.; Lu, Y.; Zhang, J.; Yang, D.; Zhao, W. Multi-platform track initiation method in dense clutter environment. In Proceedings of the 2019 International Conference on Control, Automation and Information Sciences (ICCAIS), Chengdu, China, 23–26 October 2019.
14. Liu, Z.; Gang, X.; Ding, C.S. Track Initiation Based on Adaptive gates and Fuzzy Hough Transform. *Signal Image Video Process.* **2022**, *17*, 4057–4065. [CrossRef]
15. Yang, F.; Tang, W.; Liang, Y. A novel track initialization algorithm based on random sample consensus in dense clutter. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 172988141881263. [CrossRef]
16. Pastor, A.; Sanjurjo-Rivo, M.; Escobar, D. Initial orbit determination methods for track-to-track association. *Adv. Space Res.* **2021**, *68*, 2677–2694. [CrossRef]
17. Trauwaert, E.; Kaufman, L.; Rousseeuw, P. Fuzzy clustering algorithms based on the maximum likelihood priciple. *Fuzzy Sets Syst.* **1991**, *42*, 213–227. [CrossRef]
18. Vahldiek, K.; Klawonn, F. Cluster-Centered Visualization Techniques for Fuzzy Clustering Results to Judge Single Clusters. *Appl. Sci.* **2024**, *14*, 1102. [CrossRef]
19. D'urso, P.; Massari, R. Fuzzy clustering of human activity patterns. *Fuzzy Sets Syst.* **2013**, *215*, 29–54. [CrossRef]
20. Everitt, B.S.; Landau, S.; Leese, M.; Stahl, D. *Cluster Analysis*, 5th ed.; Wiley: Hoboken, NJ, USA, 2011.
21. Gowda, K.C.; Diday, E. *Symbolic Clustering Using a New Dissimilarity Measure*; Elsevier Science Inc.: Amsterdam, The Netherlands, 1991.
22. Yeung, D.S.; Wang, X.Z. Improving performance of similarity-based clustering by feature weight learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 556–561. [CrossRef]
23. Liu, J.-X.; Li, D.-Z.; Yang, R.-G. Multitarget Track Initiation and Track Correlation based on Tracking While Scan System. *Fire Control. Command. Control.* **2005**, *30*, 4. [CrossRef]
24. He, Y.; Xiu, J.J.; Zhang, J.W. *Radar Data Processing with Applications*, 4th ed.; Publishing House of Electonics Industry: Beijing, China, 2006.
25. Śledziowski, J.; Terefenko, P.; Giza, A.; Forczmański, P.; Łysko, A.; Maćków, W.; Stępień, G.; Tomczak, A.; Kurylczyk, A. Application of Unmanned Aerial Vehicles and Image Processing Techniques in Monitoring Underwater Coastal Protection Measures. *Remote Sens.* **2022**, *14*, 458. [CrossRef]
26. Yan, H.; He, L.; Song, X.; Yao, W.; Li, C.; Zhou, Q. Bidirectional Statistical Feature Extraction Based on Time Window for Tor Flow Classification. *Symmetry* **2022**, *14*, 2002. [CrossRef]
27. Yang, C.Y.; Qu, J.M.; Mao, S.Y.; Li, S.H. An initialization method for group tracking. In Proceedings of the IEEE 1995 National Aerospace and Electronics Conference (NAECON 1995), Dayton, OH, USA, 22–26 May 1995; pp. 303–308.
28. Zhao, S.Y.; Li, X.B.; He, Y.G. Flight maneuvering ability analysis and flight path simulation of the F-22A fighter. *Flight Dyn.* **2018**, *36*, 21–24.
29. Yin, Z.P.; Huang, Y.S.H.; Tian, J.W.; Cao, S. A novel aircraft maneuver division and recognition method based on 2D flight path and perceptually important point of time sequence. *Adv. Aeronaut. Sci. Eng.* **2023**, 1–9. Available online: http://kns.cnki.net/kcms/detail/61.1479.V.20231212.1322.006.html (accessed on 1 October 2024).
30. Bar-Shalom, Y.; Daum, F.; Huang, J. The probabilistic data association filter. *IEEE Control. Syst.* **2010**, *29*, 82–100.
31. Koenig, B.C.; Kim, S.; Deng, S. KAN-ODEs: Kolmogorov–Arnold network ordinary differential equations for learning dynamical systems and hidden physics. *Comput. Methods Appl. Mech. Eng.* **2024**, *432 Pt A*, 117397. [CrossRef]