International Journal of
Network Dynamics and Intelligence (IJNDI)

Scilight

Review

Machine Learning-Embeded Bayesian Filtering: A Review

Meng Liu [1], and Xiao He [2,*]

[1] State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China
[2] Department of Automation, and the Institute for Embodied Intelligence and Robotics, Tsinghua University, Beijing 100084, China
* Correspondence: hexiao@tsinghua.edu.cn

Received: 20 December 2025; Revised: 2 April 2026; Accepted: 16 April 2026; Published: 18 June 2026

Abstract: Bayesian filtering is a state estimation method based on the Bayesian inference framework, which can be applied to state estimation of linear, nonlinear, and non-Gaussian systems, and is an essential tool in signal processing. However, modern dynamic systems often exhibit complexity and uncertainty, making it challenging for traditional Bayesian filtering to perform robust estimation under such conditions. With the rapid rise of artificial intelligence technologies, renowned for their powerful ability to extract features from statistical data, data-driven deep learning approaches have introduced new perspectives to traditional model-based estimation methods. This has led to the emergence of hybrid methods that combine data-driven and model-driven approaches with distinct functionalities. This paper summarizes and organizes these methods, elaborates on their advantages, and discusses future development directions.

Keywords: bayesian filtering; state estimation; dynamic systems; data-driven methods; model-based methods; hybrid methods

1. Introduction

Bayesian filtering [1,2] is a recursive estimation method based on Bayes' theorem, applied to state estimation in dynamic systems [3,4]. State estimation refers to the process of inferring the current state of a system by estimating its unknown variables. For instance, real-time state estimation of an aircraft typically involves multiple variables such as position, velocity, and attitude [5]. Since these variables cannot be fully observed directly through sensors, state estimation methods are required to infer them [6]. Currently, the most widely used estimation method is Bayesian filtering. The core idea of Bayesian filtering is to continuously update the estimation of the system's state by combining observational data with prior knowledge of the system, thereby effectively addressing uncertainty and noise in dynamic systems. Bayesian filtering plays a critical role in many practical applications, especially in fields such as navigation [7], robotics [8], and communication systems [9].

In state estimation of a system, the first step is to establish a state-space model. The state-space model is used to describe both linear and nonlinear dynamic systems. It characterizes the time evolution of the system through a set of state equations and uses an observation equation to relate the system's state to the observable variables:

$$\begin{aligned} x_t &= Ax_{t-1} + Bu_t + w_t, \\ z_t &= Hx_t + v_t. \end{aligned} \tag{1}$$

Here, x represents the system's state, z denotes the observation data, and u_t is the input signal. A and H are the state transition matrix and the observation matrix, respectively. B is the matrix for control input, while w_t and v_t represent Gaussian noise associated with the state and observation, respectively.

The core idea of Bayesian filtering is to recursively update the system's state based on Bayes' theorem [10]. Bayes' theorem states that, given observation data, the posterior probability can be computed from the prior probability and the likelihood function using the following formula:

$$P\left(x_t \mid z_{1:t}\right) = \frac{P\left(z_t \mid x_t\right) P\left(x_t \mid z_{1:t-1}\right)}{P\left(z_t \mid z_{1:t-1}\right)}. \tag{2}$$

where $P(x_t|z_{1:t})$ is the posterior estimate of the system state x_t given the observation data $z_{1:t}$; $P(z_t|x_t)$ is the observation model, which describes the probability of the observation data given the state x_t; $P(x_t|z_{1:t-1})$ is the prior estimate of the system state, typically propagated from the previous estimate x_{t-1}; and $P(z_t|z_{1:t-1})$ is the normalization constant for the observation data, ensuring that the total posterior probability sums to 1.

Bayesian filtering continuously updates the understanding of the system's state by combining the current observation data with the previous state estimates. This approach demonstrates excellent performance, particularly when dealing with dynamic systems that involve noise.

In recent years, with the rapid development of deep learning, state estimation based on deep learning has also emerged, owing to its strong nonlinear fitting capabilities and model-free nature [11–13]. End-to-end state estimation methods based on deep learning [14,15] can directly learn the state mapping relationships from data, eliminating the reliance on explicit models. However, compared to learning-based Bayesian filtering methods, the disadvantages of end-to-end methods are mainly as follows: they typically lack the utilization of dynamic system structures and prior knowledge, have limited capabilities for explicit modeling of observation and process noise, and are more prone to performance degradation when data is insufficient or the environment changes. On the other hand, Bayesian filtering combined with learning methods can integrate system dynamics models with data-driven techniques, offering stronger robustness and generalization capabilities. It also allows for explicit quantification of uncertainty, leading to better performance in complex dynamic environments.

A segment of the state estimation research community has focused on combining Bayesian filtering with deep learning, primarily to overcome the limitations of traditional filtering methods in certain practical applications, such as when the dynamics are known. The goal is to leverage the powerful modeling capabilities of deep learning to enhance the performance of the filter, particularly in complex and high-dimensional systems. The combination of deep learning's strong modeling capabilities with the state update advantages of Bayesian filtering can result in higher accuracy, robustness, and flexibility in many real-world applications [16,17].

This paper introduces Bayesian filtering and artificial intelligence, summarizing the excellent work that combines the two fields, covering the development process from the early simple combination to the current organic integration. In the early stages, the combination of artificial intelligence and Bayesian filtering typically took the form of a simple additive approach. For example, in fault diagnosis [18], Bayesian filtering algorithms like Kalman filtering were first used to estimate the fault state, and then deep learning methods were employed to analyze and recognize the estimation results. While this approach complemented the advantages of both methods, it did not fully explore their potential and interaction. Model-based filtering, data-driven filtering, and hybrid filtering methods are classified as shown in Table 1.

Table 1. Comparison of Bayesian filtering, deep learning, and hybrid methods.

Method	Definition	Application Scenarios
Bayesian Filtering (Model-based)	Uses probabilistic models to estimate the state of a dynamic system, updating the state estimation based on prior knowledge and observations. Represented by Kalman Filter.	Suitable for systems with accurate physical models, especially when the model is known and the system dynamics are relatively simple.
Deep Learning (Data-driven)	Utilizes neural networks and other models to learn from large amounts of data, automatically extracting patterns to make predictions and estimations.	Suitable for scenarios with no model, severe model mismatch, or when the model is too complex to explicitly define.
Hybrid Methods	Combines Bayesian filtering and deep learning, using deep learning to supplement the model deficiencies in Bayesian filtering, while retaining optimization estimation.	Suitable for scenarios with inaccurate models or where there is a need to improve the interpretability of data-driven methods.

As research progressed, the integration of artificial intelligence and Bayesian filtering gradually evolved into a more organic fusion. In this phase, researchers, through in-depth analysis of the points of intersection between the two, proposed new models such as KalmanNet [19] and DynaNet [20]. These models, leveraging the powerful capabilities of deep learning, redefined the dynamic modeling and state estimation processes of the state-space model. These novel methods are better equipped to address complex problems in high-dimensional and unknown environments, while still preserving the advantages of Bayesian filtering in recursive updates.

2. Development of Bayesian Filtering

Bayesian filtering is a state estimation method based on Bayesian theory, with its core idea being to recursively update the state of a dynamic system by combining prior information and observational data in a probabilistic manner, while also quantifying the uncertainty of the estimate. In theory, the Kalman filter (KF) [21] can be considered a special case of Bayesian filtering under certain assumptions—namely, when the system is linear and the noise is Gaussian. In this case, the Kalman filter provides the optimal Bayesian estimate by mapping low-dimensional innovations to high-dimensional state estimates using an analytical optimal gain. However, as system complexity increases, the traditional Kalman filter begins to reveal its limitations, particularly when dealing with nonlinear systems and complex noise environments. For example, the Extended Kalman Filter (EKF) extends the Kalman filter to nonlinear systems by applying a Taylor expansion to the system. The Unscented Kalman Filter (UKF) [22], on the other hand, addresses nonlinear systems and more complex noise environments by introducing the unscented transformation. Unlike the EKF, the UKF does not require a Taylor expansion and is able to more accurately capture the dynamics of nonlinear systems. Furthermore, the introduction of Particle Filtering (PF) [23] further expands the applicability of Bayesian filtering, enabling it to handle systems with highly nonlinear and non-Gaussian noise.

2.1. KF for Linear System

The KF, introduced by Rudolf E. Kálmán in 1960, is an optimal recursive algorithm used for state estimation in linear dynamic systems with Gaussian noise [24]. The filter operates by predicting the next state based on the system's dynamics and then updating this prediction with new observations to minimize the estimation error. The process consists of two key steps: prediction and update. In the prediction step, the system's state at time t is predicted based on the state at time $t-1$ using the system's dynamics model. The predicted state is given by:

$$\hat{x}_{t|t-1} = A\hat{x}_{t-1|t-1} + Bu_t, \tag{3}$$

where $\hat{x}_{t|t-1}$ represents the predicted state at time t, A is the state transition matrix, $\hat{x}_{t-1|t-1}$ is the state estimate from the previous time step, B is the control input matrix, and u_t is the control input at time t. Additionally, the uncertainty (covariance) of the predicted state is calculated as:

$$P_{t|t-1} = AP_{t-1|t-1}A^T + Q, \tag{4}$$

where $P_{t|t-1}$ is the predicted error covariance matrix, $P_{t-1|t-1}$ is the covariance of the state estimate from the previous time step, and Q is the process noise covariance matrix that models uncertainty in the system's dynamics.

In the update step, the predicted state is corrected using the new measurement z_t. The Kalman gain, K_t, which determines the weight of the new measurement, is computed as:

$$K_t = P_{t|t-1}H^T(HP_{t|t-1}H^T + R)^{-1}, \tag{5}$$

where H is the observation matrix that maps the state to the measurement space, and R is the measurement noise covariance matrix. The state estimate is then updated as:

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(z_t - H\hat{x}_{t|t-1}), \tag{6}$$

where $\hat{x}_{t|t}$ is the updated state estimate and z_t is the actual measurement at time t. Finally, the error covariance is updated to reflect the new uncertainty in the state estimate:

$$P_{t|t} = (I - K_tH)P_{t|t-1}, \tag{7}$$

where I is the identity matrix. The KF provides an optimal estimate of the system state by minimizing the mean square error of the estimate under the assumption that the system is linear and the noise is Gaussian. Its efficiency and ability to handle noisy measurements make it widely used in fields such as automatic control, navigation, robotics, and signal processing.

2.2. EKF&UKF for Non-Linear System

The KF is designed for linear systems, but its limitations in handling nonlinear dynamics led to the development of the EKF and UKF. These methods adapt the recursive estimation framework for nonlinear systems.

The EKF approximates nonlinear systems by linearizing the state transition $f(\cdot)$ and observation $h(\cdot)$ functions around the current state estimate using a first-order Taylor expansion. For a system represented as:

$$x_t = f(x_{t-1}, u_t) + w_t, \quad z_t = h(x_t) + v_t. \tag{8}$$

The state prediction and covariance update equations follow the KF's structure but use the Jacobian matrices $F_t = \frac{\partial f}{\partial x}|_{\hat{x}_{t-1}}$ and $H_t = \frac{\partial h}{\partial x}|_{\hat{x}_{t|t-1}}$. This approach introduces inaccuracies in cases of significant nonlinearity.

The UKF avoids linearization by employing the Unscented Transform to propagate the mean and covariance through the nonlinear functions. It selects a set of sigma points around the current estimate, transforms them through the nonlinear functions, and reconstructs the mean and covariance. This approach provides higher accuracy in capturing nonlinear effects without requiring explicit derivatives.

Both EKF and UKF improve the KF's applicability to nonlinear systems. However, the EKF is computationally efficient but less accurate for highly nonlinear systems, while the UKF is more accurate but computationally more intensive [25].

2.3. PF for Non-Gaussian System

The PF is a Monte Carlo-based Bayesian filtering method designed for nonlinear and non-Gaussian systems, offering flexibility beyond Kalman-based approaches [26]. It approximates the posterior distribution of the state x_t using a set of particles $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^N$, where $x_t^{(i)}$ represents the i-th particle, and $w_t^{(i)}$ is its weight. Particles are initially sampled from the prior $p(x_0)$ and propagated through the state transition model $p(x_t|x_{t-1})$. The weights are updated based on the likelihood of the observation z_t as $w_t^{(i)} \propto p(z_t|x_t^{(i)})$. The resampling step eliminates particles with low weights, focusing computational resources on high-probability regions of the state space. PF is highly versatile, effectively capturing complex distributions, but it can be computationally intensive, especially in high-dimensional systems. It has found applications in robotics, tracking, and other domains requiring robust state estimation.

2.4. STF&SVSF for Model Mismatch in Dynamic System

All of the aforementioned filters require accurate model parameters; however, model mismatch/uncertainty is a common challenge in state estimation. This necessitates the development of robust filtering techniques. The model mismatch in filtering refers to the situation where the system model used by the filter (including the state transition model, observation model, and noise statistical characteristics) deviates from or is inconsistent with the actual operating mode of the real system. Such mismatch can lead to performance degradation of the filter or even divergence. Currently, effective filtering algorithms for handling model mismatch include the Strong Tracking Filter (STF) and the Smooth Variable Structure Filter (SVSF).

The STF [27,28] is based on the core idea of introducing a fading factor λ into the filtering process to enhance the filter's sensitivity to changes in the system state, thereby improving robustness in non-stationary or rapidly changing environments. When the state of a dynamic system undergoes a sudden change, the time-varying fading factor automatically increases, causing the system's prior estimation covariance to grow. Consequently, the KF gain K increases, placing greater weight on the current measurement in the posterior state estimation. This allows the filter to rapidly and accurately track abrupt changes in the signal. In contrast, when the state changes gradually, the time-varying fading factor remains equal to 1, and the strong tracking filter degenerates into a standard EKF.

The SVSF [29,30] is a robust filtering method designed to address uncertainties in dynamic systems, particularly under model mismatch conditions. SVSF integrates principles of sliding mode control into the filtering process to improve stability and robustness.

Its key feature is the use of a smoothing function to adaptively modify the system state trajectory, mitigating the effects of disturbances and modeling errors. The filtering gain of the SVSF forces the state estimation to converge within the boundaries of the true state trajectory.

SVSF has been successfully applied in various fields, including fault diagnosis [31], control systems [32] , and tracking applications [33].

3. Data-Driven Bayesian Filtering

Deep learning is a branch of machine learning within the field of artificial intelligence (AI), with its core concept being the use of multi-layer neural networks (i.e., deep neural networks) to learn and represent complex patterns in data [34,35]. The key advantage of deep learning is its ability to automatically extract features from raw data without the need for manual feature design [36]. In research combining deep learning with filtering techniques, Recurrent Neural Networks (RNNs) [37], known for their powerful sequence data processing capabilities, are particularly suitable for filtering tasks. Traditional filtering methods, such as KF and PF, are typically used for state estimation in dynamic systems. However, deep learning, by incorporating neural networks—especially RNNs and their variants such as LSTM (Long Short-Term Memory) [38] and GRU (Gated Recurrent Units) [39]—can handle more complex temporal data, and when combined with filtering techniques, it provides enhanced modeling and estimation capabilities.

3.1. Hierarchical Overlay Strategy of Deep Learning and Bayesian Filtering

In the early applications of deep learning combined with Bayesian filtering, researchers primarily adopted a layered integration of deep learning and KF, leveraging the state estimation capability of filters and the classification ability of deep learning separately. This integration typically followed three main architectures, as illustrated in Figure 1, with primary applications in fields such as fault diagnosis [18].

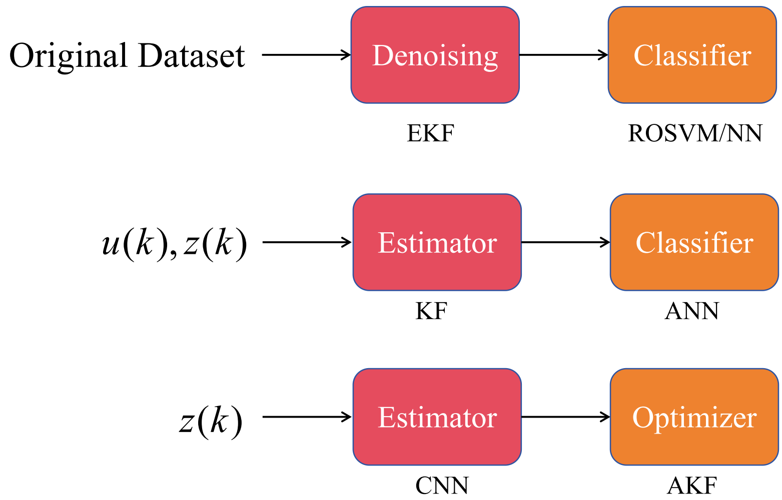

Figure 1. Three common patterns of simple integration between deep learning and Bayesian filtering, where $u(t)$ represents the control signal and $z(t)$ denotes the measurement.

In [18], a hybrid fault diagnosis method is proposed, which uses KF for fault detection. The method estimates the system's state (such as blade angle and valve position) and calculates residuals to identify faults. The state parameters generated by the KF, along with the system's input-output signals, are then fed into an artificial neural network [40] for fault classification. The neural network learns fault patterns through training and validation, enabling high-precision fault type identification. This approach combines the real-time capability of KF with the nonlinear classification ability of artificial neural networks, offering good robustness and accuracy in complex dynamic systems.

Ref. [41] proposes a hybrid fault detection method that combines EKF and Recursive One-class Support Vector Machine (ROSVM). Specifically, EKF serves as a preprocessing module, converting time-series data into parameter vectors, filtering noise, and making the data more suitable for classification. Then, ROSVM generates a boundary in high-dimensional space and recursively absorbs new normal data samples, continuously optimizing the classification model to detect anomalous data (i.e., faults). This method fully leverages the statistical modeling capability of EKF and the strong classification performance of ROSVM, making it particularly suitable for online fault detection of progressively deteriorating systems, significantly reducing the number of sensors required while ensuring a high detection rate.

When using Convolutional Neural Networks (CNN) for state estimation, data anomalies can cause errors to amplify layer by layer [42], with the anomalies reflecting in the final prediction results, leading to larger errors. Additionally, overfitting during neural network training can result in poor model generalization and robustness [43,44], both of which negatively affect estimation accuracy. To reduce estimation errors, Res. [45] proposed the use of adaptive KF to optimize the neural network's estimation results, addressing the limitations of CNN.

Currently, this hybrid combination has wide applications in recognition and tracking tasks, such as smoke detection [46], kinematics estimation [47], and traffic flow prediction [48].

3.2. Training Neural Networks Using Bayesian Filtering

The construction of deep learning models relies on the network's weights [49]. However, the loss function of the neurons within the neural network grows exponentially with depth, often leading to convergence at local minima [50]. Additionally, the weight parameters have limited adaptability [51] to dynamic and long-term features in pattern recognition tasks. From the perspective of estimation theory [52], the KF is a feasible alternative for training robust neural networks and has demonstrated advantages over traditional backpropagation [53] methods. As a result, researchers have introduced Bayesian filtering to update layer parameters in deep neural networks, as shown in Figure 2. The effectiveness of Bayesian filter-assisted deep neural networks in capturing tracking ability lies in its ability to capture the first two moments (mean and variance) of the posterior distribution of random variables.

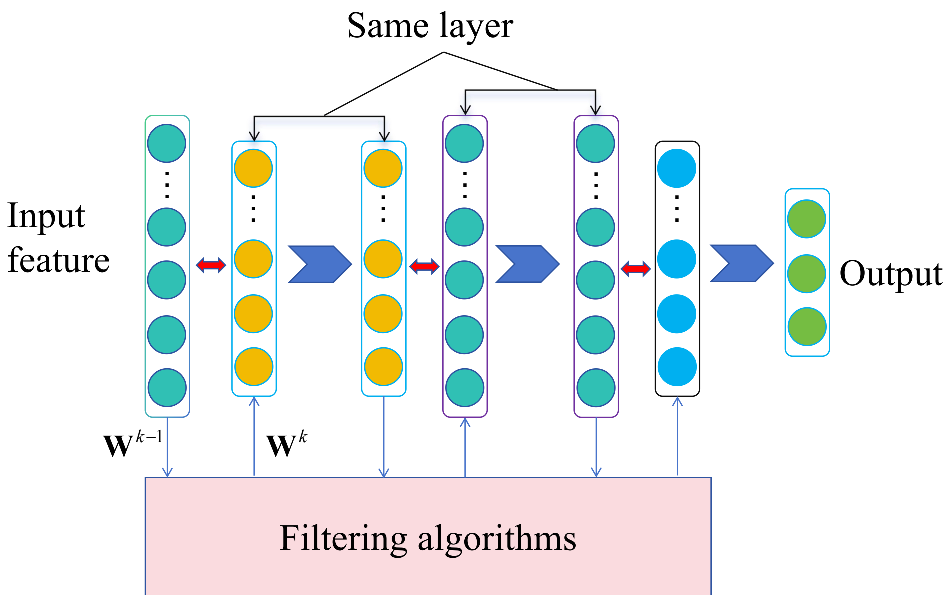

Figure 2. Framework diagram of training deep neural networks using filtering algorithms, where **W** represents the weights.

In [54], minimum UKF provides powerful nonlinear tracking capabilities, enabling the adjustment of weights and biases in the Deep Belief Network to adapt to new conditions. This dynamic adjustment capability allows the model to respond to key parameter changes during the bearing degradation process in fault diagnosis, thereby improving the accuracy and adaptability of the diagnosis.

Ref. [55] adopts a dual optimization strategy by combining Cubic Kalman Filtering (CKF) with Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF) neural networks. The former optimizes the MLP weights to adapt to dynamic changes, while the latter compensates for filtering errors. This significantly improves the positioning accuracy and robustness of the navigation system in the case of GPS failure. These methods enhance the system's real-time performance, accuracy, and the versatility of its application scenarios by integrating statistical modeling with machine learning techniques.

Ref. [56] proposes a new resilient control strategy to maintain the stability and performance of a multi-area load frequency control system under the threat of FDIA (False Data Injection Attacks). In [57], the decoupled EKF algorithm used for training neural networks is applied to image classification in deep convolutional neural networks. The decoupled EKF algorithm offers advantages over the popular Adam optimizer, especially in challenging classification tasks. Fully Decoupled EKF, a second-order algorithm, has a higher probability of finding the global optimum, leading to faster convergence and higher accuracy.

3.3. Learning SSM Parameters Based on Deep Learning

State-space models are core tools in control systems for characterizing dynamic behavior, playing a vital role in both system analysis and control design. In classical control theory, they provide a comprehensive representation of the internal state evolution and the input–output relationships of a system, forming the foundation for modern controller design methods such as state feedback and Kalman filtering. However, in practical applications, parameter identification—that is, determining model parameters from experimental data—has long been a highly challenging task. Traditional identification methods, such as least squares, maximum likelihood estimation, and subspace identification, often require prior assumptions about the model's structure and order, and rely heavily on manual trial-and-error. For highly nonlinear, time-varying, or high-dimensional complex systems, these approaches frequently fall short: inaccurate model assumptions may lead to significant deviations, while improper order selection can result in underfitting or overfitting. As the complexity of real-world systems continues to increase, traditional identification techniques face severe challenges in modeling accuracy and scalability, making it difficult to fully uncover the latent dynamic patterns hidden in data.

In contrast to classical models based on prior assumptions, deep neural networks depend less on prior model knowledge and can approximate system dynamics within a broader function space. Leveraging the flexible parameterization capability of deep learning, it becomes possible to expand the range of candidate models and capture the hidden properties of complex systems with greater accuracy. Compared with state estimation methods purely based on deep learning, state-space models (SSMs) [58,59] are capable of constructing parameterized models and providing explicit transition relations to describe the evolution of system states. A representative example is the

Kalman filter (KF), a model-driven state estimator based on Bayesian principles, which realizes state prediction and update through a prediction–correction mechanism.

Against this backdrop, recent studies have explored the use of deep neural networks to learn SSM parameters, including the state transition model, observation model, and noise statistics [60] (as illustrated in Figure 3). When using deep learning to learn the parameters of SSM, neural networks are typically used to learn a probability distribution, with the parameters of the SSM being sampled from this distribution. Deep learning can automatically learn the complex patterns in data through neural networks, without relying on the model structures that need to be assumed in traditional system identification. It is suitable for handling complex dynamic systems that are often difficult to model using traditional methods. Compared with traditional methods, deep learning not only enables efficient modeling of high-dimensional complex systems or systems with incomplete models, but also eliminates the need for manual feature extraction. Furthermore, it demonstrates strong generalization ability, thereby improving the accuracy of state estimation [20] and opening new possibilities for modeling and control of complex systems.

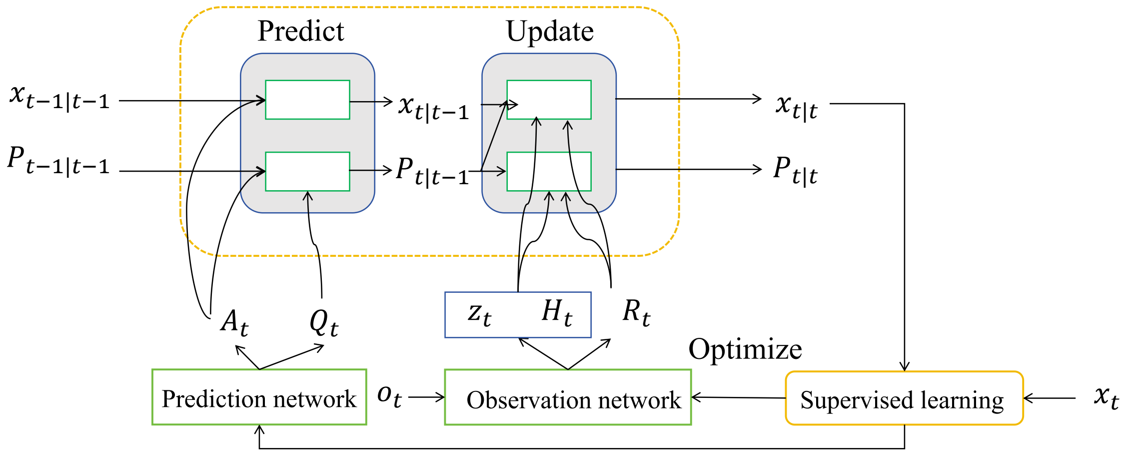

Figure 3. Framework diagram of learning SSM parameters using deep learning.

Currently, deep learning-based filtering parameter learning has achieved success in various fields, including noise estimation [61] , harmonic current compensation [62], visual odometry [63], and camera pose estimation [64].

Ref. [65] proposed a robust Kalman filter (RKF) method that combines the heavy-tailed distribution RKF framework with deep learning techniques, eliminating the need for precise parameter estimation of noise models. By estimating the mixing parameters and scale matrix using neural network structures, it reduces the variational Bayesian approximation error. Additionally, the unsupervised scheduled sampling (USS) method is introduced to stabilize the training process. RKFNet demonstrates higher estimation accuracy and efficiency under heavy-tailed noise compared to traditional RKFs and recurrent neural networks (RNNs).

The main research challenge lies in how to perform effective estimation when the dimensions of the observation data and the state are inconsistent.

3.4. Data-Driven Kalman Gain Learning

The essence of the Kalman Gain lies in balancing the uncertainties between the predicted value and the observed value, determining which side to rely on more when updating the state estimate [64,66]. It reflects the relative credibility between the predicted value (provided by the dynamic model) and the observed value (measured by sensors) at the current time step [67].

The Kalman Gain is a critical parameter in the update step of the KF, adjusting the weights of the predicted value and the observed value based on the ratio of process noise (Q) to observation noise (R). When the uncertainty in the predictive model is high (higher process noise), the Kalman Gain increases, relying more on the observations [68]. Conversely, when the observation noise is higher, the gain decreases, placing more weight on the predictions. This mechanism ensures the optimality of state estimation by minimizing estimation errors.

However, in state estimation problems where model parameters are uncertain or partially known, explicitly computing the Kalman Gain can be challenging [69,70]. The advantage of leveraging deep learning for Kalman Gain computation lies in its ability to adaptively optimize the gain through learning from large datasets, overcoming the strict assumptions on process and observation noise covariance matrices in traditional KF. Deep learning models reduce dependency on manual parameter tuning and improve the efficiency and accuracy of Kalman Gain computation through an end-to-end training process, especially in environments with incomplete model knowledge.

Revach [19] innovatively proposed KalmanNet, a hybrid system combining Recurrent Neural Networks (RNNs) with the traditional KF for real-time state estimation. As shown in Figure 4, KalmanNet is designed to handle dynamic systems well-represented by linear Gaussian state-space (SS) models, particularly when model parameters are partially known (model mismatch) or involve nonlinear dynamics. The computational complexity of KalmanNet scales linearly with the RNN's dimensionality and does not involve matrix inversion, making it suitable for high-dimensional state-space models and devices with limited computational resources.

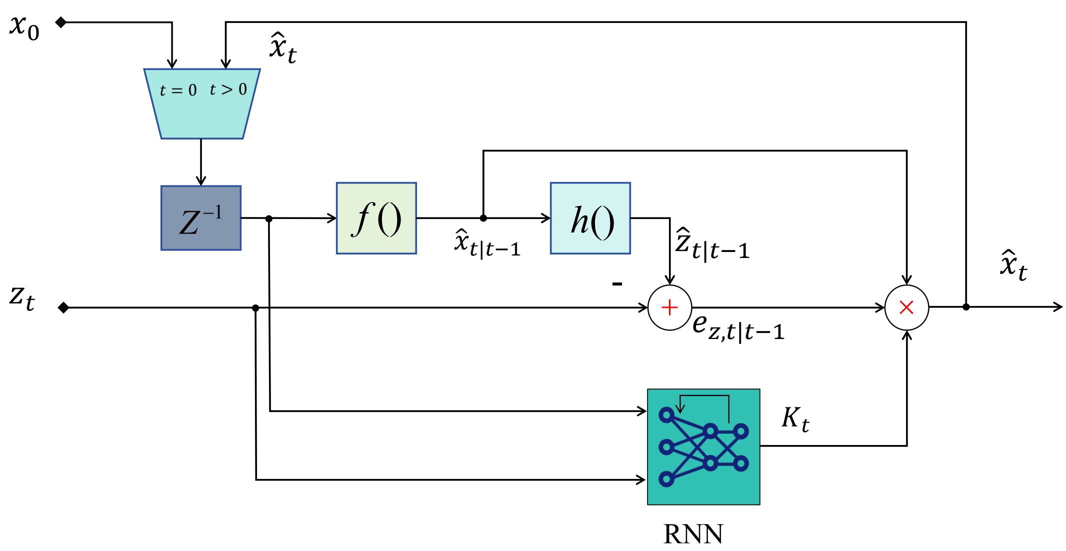

Figure 4. KalmanNet block diagram.

However, KalmanNet does not fully exploit the model-based structure when computing the Kalman gain, leading to performance degradation in state estimation when mismatched state and measurement models coexist. To address this, Choi [71] proposed Split-KalmanNet, which employs two parallel Deep Neural Networks to train the prior state estimate and the innovation covariance matrix. This algorithm leverages the structural computation of Kalman gain within the standard KF framework, enabling the DNN-trained Kalman gain to better align with the model-based Kalman gain in practice. To address the inflexibility of neural network architectures and the scarcity of training data, Ref. [72] proposed a model-agnostic meta-learning–based neural network-assisted Kalman filter, which can quickly adapt to model changes with only a small amount of data, thus avoiding retraining.

Inspired by recent advancements in the fine-tuning paradigm of large language models, Revach [73] proposed Adaptive KalmanNet, which employs a compact hypernetwork to generate context-dependent modulation weights. This approach allows the model to adapt to changes in the SSM without the need for retraining. Ref. [74] proposed a new method called Bayesian KalmanNet, which combines Bayesian deep learning techniques with KalmanNet. By using sampling techniques, it predicts the error covariance without requiring additional domain knowledge, thereby maintaining accurate and reliable tracking performance in partially known dynamic systems.

KalmanNet is one of the most successful learning-based filtering methods to date. It has already been applied in practical scenarios such as integrated navigation [75] and autonomous driving [76].

4. Conclusions and Outlook

The integration of deep learning and Bayesian filtering, as a frontier direction in modern artificial intelligence and signal processing, has been applied in areas such as noise suppression [77], target tracking [78], robot localization [79], and workload prediction [80].

4.1. Conclusions

This paper explored the current state of research on the integration of Bayesian filtering and deep learning, emphasizing the complementary advantages of both approaches. Bayesian filtering, as a classical state estimation method, offers the ability to quantify uncertainty through probabilistic reasoning, while deep learning, with its powerful nonlinear modeling capabilities and automatic feature extraction abilities, effectively addresses unknown dynamics in complex systems. In particular, the fusion of KF's powerful estimation capabilities and deep learning's adaptive learning abilities provides new perspectives for tasks such as modeling [81], control, and prediction [82] of complex systems. As system models are becoming increasingly large and complex [83], the future development of integrating deep learning with Bayesian filtering should focus on the interpretability of deep learning [84] models and include some degree of fault-tolerant estimation [85] capabilities.

In the comparison of optimality and application differences, the optimality of the Kalman filter refers to minimizing the mean squared error and covariance matrix under the conditions of a known linear model and Gaussian noise, providing a theoretically optimal solution. In deep learning, optimality is achieved by optimizing a loss function in a data-driven manner, without relying on an exact model or assumptions. Its optimality is more empirical. Kalman filtering provides an optimal solution based on theoretical guarantees, while deep learning achieves robust solutions through data-driven learning. Currently, data-driven approaches cannot guarantee optimality or convergence and are seen as an alternative for filtering in complex systems, optimizing the weights of the Kalman gain network via a loss function (estimation error). Although neural networks are powerful, many common deep learning methods (such as standard feedforward neural networks) do not automatically provide uncertainty quantification like covariance matrices. These methods typically output a deterministic prediction without considering the uncertainty of the output. Therefore, when deep learning models are directly used as alternatives or supplements to Bayesian filtering, they may lack the inherent uncertainty quantification of Bayesian filtering and cannot guarantee an optimal analytical solution. Between hybrid estimation and pure deep learning estimation, there are differences in applicable scenarios and computational complexity, and the optimality cannot guarantee which is superior.

4.2. Outlook

4.2.1. Complex System Filtering

As systems become increasingly complex (high-dimensional and strongly nonlinear) [86] and exhibit high coupling of internal parameters, state estimation faces new challenges [87]. In high-dimensional systems, traditional filtering methods often struggle to effectively handle the complexity of the state space. Therefore, using Autoencoders [88] for dimensionality reduction and feature extraction has emerged as an effective solution. Autoencoders can automatically learn low-dimensional latent representations from high-dimensional data, which capture the core information of the system more effectively and provide useful inputs for Bayesian filtering. This approach not only reduces computational load but also enhances the performance of Bayesian filtering in high-dimensional systems.

On the other hand, for the high coupling of internal parameters, Decoupling Representation Learning [89] offers a potential solution. This technique aims to decompose complex coupled systems into multiple independent subsystems or factors through learning, thereby reducing the interactions between parameters and simplifying system modeling.

4.2.2. Interpretable Hybrid Filtering Methods

Deep learning models, especially those based on neural networks, are often considered black-box models due to their lack of interpretability [90]. Although deep neural networks perform exceptionally well in many tasks, their internal mechanisms are difficult to understand and analyze [91]. On the other hand, Bayesian filtering provides a probability-based reasoning framework that allows each step of the state estimation to be clearly interpreted. In complex systems, deep learning can significantly improve filtering accuracy by constructing intricate nonlinear models, but this complexity often results in a loss of interpretability. Deep neural networks fit data by utilizing a large number of parameters and layers, which makes their predictions highly accurate, but these internal processes are difficult for humans to comprehend.

Traditional Bayesian filtering methods and similar techniques typically have better interpretability, but their expressiveness and accuracy may not be sufficient to handle highly complex and nonlinear systems. To improve accuracy, more complex models such as deep learning may need to be introduced, but this could come at the cost of losing some interpretability.

It is necessary to develop explainable hybrid filtering methods. Currently, Kolmogorov–Arnold Networks (KAN) [92,93] emerge as a powerful deep learning architecture with high interpretability, which can be integrated with Bayesian filtering to provide enhanced interpretability and overcome the black-box problem commonly associated with deep learning. KAN, by their design, offer a structured way to represent complex nonlinear systems while maintaining transparency in their decision-making process, making them an attractive option for improving the interpretability of hybrid filtering models. When combined with Bayesian filtering, which inherently supports probabilistic reasoning and model transparency, this approach allows for better interpretability of the overall system, ensuring that both the predictive power of deep learning and the clear reasoning of Bayesian methods are preserved.

4.2.3. Fault-Tolerant Filtering

In real-world applications, system [94] or sensor failures [95] are inevitable. Ensuring the stability and accuracy of the filter when a system fault or anomaly occurs is a key research direction. Bayesian filtering, with its probabilistic reasoning and adaptive mechanisms, can effectively handle situations where some sensors fail or the model is inaccurate. However, deep learning typically requires large amounts of high-quality data for training and is sensitive to noise and anomalies. Therefore, integrating deep learning with Bayesian filtering to enhance the system's fault tolerance is an important research challenge.

Deep learning models can automatically adjust parameters or gains through training without the need for a completely accurate system model. This approach is particularly suitable for scenarios where the system model is unknown or uncertain. By learning the relationship between input data and system output, deep learning can effectively identify and estimate the system's state, providing more accurate results even in cases where traditional methods fail to deliver precise estimates due to model errors or mismatches.

In the field of fault-tolerant filtering for distributed state estimation, deep learning models can analyze sensor data to determine whether the data follows normal patterns. If a significant deviation is detected between the output of one sensor and those of others, the sensor is deemed to have a potential fault, and its data can be isolated.

Author Contributions: M.L.: conceptualization, methodology, writing—original draft preparation; X.H.: visualization, investigation, supervision.

Funding: This work was supported in part by National Natural Science Foundation of China under grants 62525308, 62473223, and 52172323, in part by Beijing Natural Science Foundation under grant L241016, in part by National Natural Science Foundation of China under grants 62163012.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Use of AI and AI-Assisted Technologies: No AI tools were utilized for this paper.

References

1. Silva, P.R.; Bruno, M.G.; Moraes, A.O. Cooperative-Based Augmentation System Using Bayesian Filtering. *IEEE Trans. Veh. Technol.* **2024**, *73*, 10502–10515.
2. Wu, G.; Lindquist, A. A Non-Gaussian Bayesian Filter Using Power and Generalized Logarithmic Moments. *Automatica* **2024**, *166*, 111748.
3. Yue, H.; He, H.; Han, M.; et al. Active Disturbance Rejection Control Strategy for PEMFC Oxygen Excess Ratio Based on Adaptive Internal State Estimation Using Unscented Kalman Filter. *Fuel* **2024**, *356*, 129619.
4. Lv, H.; Liao, Y.; Zhao, C.; et al. State of Charge Estimation of Lithium-Titanate Battery Based on Multi-Model Extended Kalman Filter Considering Temperature and Current Rate. *J. Energy Storage* **2024**, *77*, 109890.
5. Fry, C.; Sage, A. Identification of Aircraft Stability and Control Parameters Using Hierarchical State Estimation. *IEEE Trans. Aerosp. Electron. Syst.* **1974**, *AES-10*, 255–273.
6. Jin, F.; Ma, L.; Zhao, C.; et al. State Estimation in Networked Control Systems with a Real-Time Transport Protocol. *Syst. Sci. Control Eng.* **2024**, *12*, 2347885.
7. Crassidis, J.L. Sigma-Point Kalman Filtering for Integrated GPS and Inertial Navigation. *IEEE Trans. Aerosp. Electron. Syst.* **2006**, *42*, 750–756.
8. Chen, S.Y. Kalman Filter for Robot Vision: A Survey. *IEEE Trans. Ind. Electron.* **2011**, *59*, 4409–4420.
9. Chen, X.; Feng, Z.; Zhang, J.A.; et al. Kalman Filter-Based Sensing in Communication Systems with Clock Asynchronism. *IEEE Trans. Commun.* **2023**, *72*, 403–417.
10. Efron, B. Bayes' Theorem in the 21st Century. *Science* **2013**, *340*, 1177–1178.
11. Shlezinger, N.; Revach, G.; Ghosh, A.; et al. AI-Aided Kalman Filters. *arXiv* **2024**, arXiv:2410.12289.
12. Feng, S.; Li, X.; Zhang, S.; et al. A Review: State Estimation Based on Hybrid Models of Kalman Filter and Neural Network. *Syst. Sci. Control Eng.* **2023**, *11*, 2173682.
13. Cui, Z.; Dai, J.; Sun, J.; et al. Hybrid Methods Using Neural Network and Kalman Filter for the State of Charge Estimation of Lithium-Ion Battery. *Math. Probl. Eng.* **2022**, *2022*, 9616124.
14. Li, P.; Zhang, Z.; Grosu, R.; et al. An End-to-End Neural Network Framework for State-of-Health Estimation and Remaining Useful Life Prediction of Electric Vehicle Lithium Batteries. *Renew. Sustain. Energy Rev.* **2022**, *156*, 111843.
15. Tian, J.; Xiong, R.; Shen, W.; et al. State-of-Charge Estimation of LiFePO4 Batteries in Electric Vehicles: A Deep-Learning Enabled Approach. *Appl. Energy* **2021**, *291*, 116812.

16. Liu, J.; Guo, G. Vehicle Localization During GPS Outages with Extended Kalman Filter and Deep Learning. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 7503410.
17. Bai, Y.; Yan, B.; Zhou, C.; et al. State of Art on State Estimation: Kalman Filter Driven by Machine Learning. *Annu. Rev. Control* **2023**, *56*, 100909.
18. Cho, S.; Choi, M.; Gao, Z.; et al. Fault Detection and Diagnosis of a Blade Pitch System in a Floating Wind Turbine Based on Kalman Filters and Artificial Neural Networks. *Renew. Energy* **2021**, *169*, 1–13.
19. Revach, G.; Shlezinger, N.; Ni, X.; et al. KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics. *IEEE Trans. Signal Process.* **2022**, *70*, 1532–1547.
20. Chen, C.; Lu, C.X.; Wang, B.; et al. DynaNet: Neural Kalman Dynamical Model for Motion Estimation and Prediction. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 5479–5491.
21. Wang, Y.; Wen, C.; Wu, X. Fault Detection and Isolation of Floating Wind Turbine Pitch System Based on Kalman Filter and Multi-Attention 1DCNN. *Syst. Sci. Control Eng.* **2024**, *12*, 2362169.
22. Fariña, B.; Toledo, J.; Acosta, L. Sensor Fusion Algorithm Selection for an Autonomous Wheelchair Based on EKF/UKF Comparison. *Int. J. Mech. Eng. Robot. Res.* **2023**, *12*, 1–7.
23. Cui, L.; Li, W.; Liu, D.; et al. A Novel Robust Dual Unscented Particle Filter Method for Remaining Useful Life Prediction of Rolling Bearings. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 3509009.
24. Khodarahmi, M.; Maihami, V. A Review on Kalman Filter Models. *Arch. Comput. Methods Eng.* **2023**, *30*, 727–747.
25. Allotta, B.; Chisci, L.; Costanzi, R.; et al. A Comparison Between EKF-Based and UKF-Based Navigation Algorithms for AUVs Localization. In Proceedings of the OCEANS 2015-Genova, Genova, Italy, 18–21 May 2015; pp. 1–5.
26. Djuric, P.M.; Kotecha, J.H.; Zhang, J.; et al. Particle Filtering. *IEEE Signal Process. Mag.* **2003**, *20*, 19–38.
27. Zhou, D.; Frank, P. Strong Tracking Filtering of Nonlinear Time-Varying Stochastic Systems with Coloured Noise: Application to Parameter Estimation and Empirical Robustness Analysis. *Int. J. Control* **1996**, *65*, 295–307.
28. Kiani, M.; Ahmadvand, R. The Strong Tracking Innovation Filter. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 3261–3270.
29. Avzayesh, M.; Abdel-Hafez, M.; AlShabi, M.; et al. The Smooth Variable Structure Filter: A Comprehensive Review. *Digit. Signal Process.* **2021**, *110*, 102912.
30. Goodman, J.; Hilal, W.; Gadsden, S.A.; et al. Comparison of SVSF-KF Adaptive Estimation Algorithms on an Electro-Hydrostatic Actuator Subject to a Fault. *IEEE Sens. J.* **2024**, *25*, 2905–2920.
31. Afshari, H.H.; Gadsden, S.A.; Habibi, S.R. Robust Fault Diagnosis of an Electro-Hydrostatic Actuator Using the Novel Dynamic Second-Order SVSF and IMM Strategy. *Int. J. Fluid Power* **2014**, *15*, 181–196.
32. Oussar, A.; Ferrag, A.; Guiatni, M.; et al. SVSF-Based Robust UGV/UAV Control/Tracking Architecture in Disturbed Environment. *Intell. Autom. Soft Comput.* **2021**, *29*, 467–495.
33. Li, Y.; Li, G.; Liu, Y.; et al. A Hybrid SVSF Algorithm for Automotive Radar Tracking. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 15028–15042.
34. Rusk, N. Deep Learning. *Nat. Methods* **2016**, *13*, 35.
35. Nafea, A.A.; Alameri, S.A.; Majeed, R.R.; et al. A Short Review on Supervised Machine Learning and Deep Learning Techniques in Computer Vision. *Babylon. J. Mach. Learn.* **2024**, *2024*, 48–55.
36. Ma, C.; Cheng, P.; Cai, C. Localization and Mapping Method Based on Multimodal Information Fusion and Deep Learning for Dynamic Object Removal. *Int. J. Netw. Dyn. Intell.* **2024**, *3*, 100008.
37. Sani, O.G.; Pesaran, B.; Shanechi, M.M. Dissociative and Prioritized Modeling of Behaviorally Relevant Neural Dynamics Using Recurrent Neural Networks. *Nat. Neurosci.* **2024**, *27*, 2033–2045.
38. Al-Selwi, S.M.; Hassan, M.F.; Abdulkadir, S.J.; et al. RNN-LSTM: From Applications to Modeling Techniques and Beyond—Systematic Review. *J. King Saud Univ. Comput. Inf. Sci.* **2024**, *36*, 102068.
39. Bonassi, F.; La Bella, A.; Farina, M.; et al. Nonlinear MPC Design for Incrementally ISS Systems with Application to GRU Networks. *Automatica* **2024**, *159*, 111381.
40. Aguirre, F.; Sebastian, A.; Le Gallo, M.; et al. Hardware Implementation of Memristor-Based Artificial Neural Networks. *Nat. Commun.* **2024**, *15*, 1974.
41. Yan, K.; Ji, Z.; Shen, W. Online Fault Detection Methods for Chillers Combining Extended Kalman Filter and Recursive One-Class SVM. *Neurocomputing* **2017**, *228*, 205–212.
42. Shi, R.; Mo, Z.; Di, X. Physics-Informed Deep Learning for Traffic State Estimation: A Hybrid Paradigm Informed by Second-Order Traffic Models. In Proceedings of the AAAI Conference on Artificial Intelligence, online, 2–9 February 2021; Volume 35, pp. 540–547.
43. Wang, J.; Zhuang, Y.; Liu, Y. FSS-Net: A Fast Search Structure for 3D Point Clouds in Deep Learning. *Int. J. Netw. Dyn. Intell.* **2023**, *2*, 100005.
44. Arulananth, T.; Prakash, S.W.; Ayyasamy, R.K.; et al. Classification of Paediatric Pneumonia Using Modified DenseNet-121 Deep-Learning Model. *IEEE Access* **2024**, *12*, 35716–35727.
45. Li, J.; Wang, S.; Chen, L.; et al. Adaptive Kalman Filter and Self-Designed Early Stopping Strategy Optimized Convolutional Neural Network for State of Energy Estimation of Lithium-Ion Battery in Complex Temperature Environment. *J. Energy Storage* **2024**, *83*, 110750.

46. Gagliardi, A.; de Gioia, F.; Saponara, S. A Real-Time Video Smoke Detection Algorithm Based on Kalman Filter and CNN. *J. Real-Time Image Process.* **2021**, *18*, 2085–2095.
47. Bao, T.; Zhao, Y.; Zaidi, S.A.R.; et al. A Deep Kalman Filter Network for Hand Kinematics Estimation Using sEMG. *Pattern Recognit. Lett.* **2021**, *143*, 88–94.
48. Zhang, H.; Yang, G.; Yu, H.; et al. Kalman Filter-Based CNN-BiLSTM-ATT Model for Traffic Flow Prediction. *Comput. Mater. Contin.* **2023**, *76*, 1047.
49. Newman, M.E. Analysis of Weighted Networks. *Phys. Rev. E* **2004**, *70*, 056131.
50. Sun, S.; Chen, W.; Wang, L.; et al. On the Depth of Deep Neural Networks: A Theoretical View. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30, pp. 2066–2072.
51. Riid, A.; Rüstern, E. Adaptability, Interpretability and Rule Weights in Fuzzy Rule-Based Systems. *Inf. Sci.* **2014**, *257*, 301–312.
52. Gao, X.; Deng, F.; Shang, W.; et al. Attack-Resilient Asynchronous State Estimation of Interval Type-2 Fuzzy Systems Under Stochastic Protocols. *Int. J. Syst. Sci.* **2024**, *55*, 2688–2700.
53. Feng, J.; Cai, F.; Wu, L.; et al. State of Charge Estimation for Lithium-Ion Battery Pack Based on Real Vehicle Data and Optimized Backpropagation Method by Adaptive Cross Mutation Sparrow Search Algorithm. *Energy Sci. Eng.* **2024**, *12*, 896–912.
54. Tang, H.; Tang, Y.; Su, Y.; et al. Feature Extraction of Multi-Sensors for Early Bearing Fault Diagnosis Using Deep Learning Based on Minimum Unscented Kalman Filter. *Eng. Appl. Artif. Intell.* **2024**, *127*, 107138.
55. Shen, C.; Zhang, Y.; Tang, J.; et al. Dual-Optimization for a MEMS-INS/GPS System During GPS Outages Based on the Cubature Kalman Filter and Neural Networks. *Mech. Syst. Signal Process.* **2019**, *133*, 106222.
56. Abbaspour, A.; Sargolzaei, A.; Forouzannezhad, P.; et al. Resilient Control Design for Load Frequency Control System Under False Data Injection Attacks. *IEEE Trans. Ind. Electron.* **2019**, *67*, 7951–7962.
57. Gaytan, A.; Begovich-Mendoza, O.; Arana-Daniel, N. Training of Convolutional Neural Networks for Image Classification with Fully Decoupled Extended Kalman Filter. *Algorithms* **2024**, *17*, 243.
58. Zhang, R.; Li, G.; Qu, S.; et al. Mamba-GIE: A Visual State Space Models-Based Generalized Image Extrapolation Method via Dual-Level Adaptive Feature Fusion. *Expert Syst. Appl.* **2025**, *264*, 125961.
59. Svensson, A.; Schön, T.B. A Flexible State-Space Model for Learning Nonlinear Dynamical Systems. *Automatica* **2017**, *80*, 189–199.
60. Gedon, D.; Wahlström, N.; Schön, T.B.; et al. Deep State Space Models for Nonlinear System Identification. *IFAC-PapersOnLine* **2021**, *54*, 481–486.
61. Zou, Z.; Huang, T.; Ye, L.; et al. CNN Based Adaptive Kalman Filter in High-Dynamic Condition for Low-Cost Navigation System on Highspeed UAV. In Proceedings of the 2020 5th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Singapore, 17–19 July 2020; pp. 103–108.
62. Yang, J.; Fang, Y.; Fei, J. Chebyshev Fuzzy Neural Finite Time Strategy Using Extended Kalman Filter. *IEEE Internet Things J.* **2024**, *11*, 28452–28464.
63. Zhao, C.; Sun, L.; Yan, Z.; et al. Learning Kalman Network: A Deep Monocular Visual Odometry for On-Road Driving. *Robot. Auton. Syst.* **2019**, *121*, 103234.
64. Coskun, H.; Achilles, F.; DiPietro, R.; et al. Long Short-Term Memory Kalman Filters: Recurrent Neural Estimators for Pose Regularization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5524–5532.
65. Hao, P.; Karakuş, O.; Achim, A. RKFNet: A Novel Neural Network Aided Robust Kalman Filter. *Signal Process.* **2025**, *230*, 109856.
66. Zhang, Y.; Oliver, D.S. Improving the Ensemble Estimate of the Kalman Gain by Bootstrap Sampling. *Math. Geosci.* **2010**, *42*, 327–345.
67. Thacker, N.; Lacey, A. Tutorial: The Kalman Filter. 1998. Available online: https://www.scribd.com/document/94341290/LaceyThackerTutorial (accessed on 15 April 2026).
68. Welch, G. *An Introduction to the Kalman Filter*; Technical Report; University of North Carolina at Chapel Hill: Chapel Hill, NC, USA, 1995.
69. Ortega, L.; Vilà-Valls, J.; Chaumette, E.; et al. Robust Tracking Under Measurement Model Mismatch via Linearly Constrained Extended Kalman Filtering. In Proceedings of the 2020 59th IEEE Conference on Decision and Control (CDC), Jeju, Korea, 14–18 December 2020; pp. 2924–2929.
70. Kullberg, A.; Skog, I.; Hendeby, G. Online Joint State Inference and Learning of Partially Unknown State-Space Models. *IEEE Trans. Signal Process.* **2021**, *69*, 4149–4161.
71. Choi, G.; Park, J.; Shlezinger, N.; et al. Split-KalmanNet: A Robust Model-Based Deep Learning Approach for State Estimation. *IEEE Trans. Veh. Technol.* **2023**, *72*, 12326–12331.
72. Chen, S.; Zheng, Y.; Lin, D.; et al. MAML-KalmanNet: A Neural Network-Assisted Kalman Filter Based on Model-Agnostic Meta-Learning. *IEEE Trans. Signal Process.* **2025**, *73*, 988–1003.
73. Ni, X.; Revach, G.; Shlezinger, N. Adaptive Kalmannet: Data-Driven Kalman Filter with Fast Adaptation. In Proceedings

of the ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Korea, 14–19 April 2024; pp. 5970–5974.
74. Dahan, Y.; Revach, G.; Dunik, J.; et al. Bayesian KalmanNet: Quantifying Uncertainty in Deep Learning Augmented Kalman Filter. *IEEE Trans. Signal Process.* **2025**, *73*, 2558–2573.
75. Song, J.; Mei, W.; Xu, Y.; et al. Practical Implementation of KalmanNet for Accurate Data Fusion in Integrated Navigation. *IEEE Signal Process. Lett.* **2024**, *31*, 1890–1894.
76. Piperigkos, N.; Gkillas, A.; Anagnostopoulos, C.; et al. Cooperative Plug-and-Play-KalmanNet for 4D Situational Awareness in Autonomous Driving. In Proceedings of the 2024 IEEE 7th International Conference on Industrial Cyber-Physical Systems (ICPS), St. Louis, MO, USA, 12–15 May 2024; pp. 1–6.
77. Adžemović, M.; Tadić, P.; Petrović, A.; et al. Beyond Kalman Filters: Deep Learning-Based Filters for Improved Object Tracking. *Mach. Vis. Appl.* **2025**, *36*, 20.
78. Li, Y.; Yu, H.; Xiao, L.; et al. Inspection Robot GPS Outages Localization Based on Error Kalman Filter and Deep Learning. *Robot. Auton. Syst.* **2025**, *183*, 104824.
79. Naveen Kumar, M.; Annappa, B.; Yadav, V. Efficient Kalman Filter Based Deep Learning Approaches for Workload Prediction in Cloud and Edge Environments. *Computing* **2025**, *107*, 10.
80. Luo, Z.; Shi, D.; Gan, W.S.; et al. Delayless Generative Fixed-Filter Active Noise Control Based on Deep Learning and Bayesian Filter. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2023**, *32*, 1048–1060.
81. Bhalodia, R.; Elhabian, S.; Adams, J.; et al. DeepSSM: A Blueprint for Image-to-Shape Deep Learning Models. *Med. Image Anal.* **2024**, *91*, 103034.
82. Deutsch, J.; He, M.; He, D. Remaining Useful Life Prediction of Hybrid Ceramic Bearings Using an Integrated Deep Learning and Particle Filter Approach. *Appl. Sci.* **2017**, *7*, 649.
83. Schär, S.; Marelli, S.; Sudret, B. Emulating the Dynamics of Complex Systems Using Autoregressive Models on Manifolds (mNARX). *Mech. Syst. Signal Process.* **2024**, *208*, 110956.
84. Bolton, W.J.; Wilson, R.; Gilchrist, M.; et al. Personalising Intravenous to Oral Antibiotic Switch Decision Making Through Fair Interpretable Machine Learning. *Nat. Commun.* **2024**, *15*, 506.
85. Ge, Q.; Ma, Z.; Lu, Z.; et al. Fault-Tolerant Cubature Kalman Filter for Engineering Estimation Control Systems. *IEEE Trans. Cybern.* **2024**, *54*, 3943–3953.
86. Landry, N.W.; Amburg, I.; Shi, M.; et al. Filtering Higher-Order Datasets. *J. Phys. Complex.* **2024**, *5*, 015006.
87. Han, B. Distributionally Robust Kalman Filtering with Volatility Uncertainty. *IEEE Trans. Autom. Control* **2024** *70*, 4000–4007.
88. Lyu, S.; Sowlati-Hashjin, S.; Garton, M. Variational Autoencoder for Design of Synthetic Viral Vector Serotypes. *Nat. Mach. Intell.* **2024**, *6*, 147–160.
89. Stooke, A.; Lee, K.; Abbeel, P.; et al. Decoupling Representation Learning from Reinforcement Learning. In Proceedings of the International Conference on Machine Learning, online, 18–24 July 2021; pp. 9870–9879.
90. Yang, X.; Lin, J.; Zhang, H.; et al. Improving Query Efficiency of Black-Box Attacks via the Preference of Deep Learning Models. *Inf. Sci.* **2024**, *678*, 121013.
91. Fan, J.; Sui, Z.; Wang, X. Multi-Physical Interpretable Deep Learning Network for Oil Spill Identification Based on SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 4202015.
92. Liu, Z.; Wang, Y.; Vaidya, S.; et al. Kan: Kolmogorov-Arnold Networks. *arXiv* **2024**, arXiv:2404.19756.
93. Genet, R.; Inzirillo, H. Tkan: Temporal Kolmogorov-Arnold Networks. *arXiv* **2024**, arXiv:2405.07344.
94. Vaish, R.; Dwivedi, U.; Tewari, S.; et al. Machine Learning Applications in Power System Fault Diagnosis: Research Advancements and Perspectives. *Eng. Appl. Artif. Intell.* **2021**, *106*, 104504.
95. Li, H.; Gao, Y.; Shi, P.; et al. Observer-Based Fault Detection for Nonlinear Systems with Sensor Fault and Limited Communication Capacity. *IEEE Trans. Autom. Control* **2015**, *61*, 2745–2751.