Mixture of Experts of Neural Networks and Kalman Filters for Optical Belt Sorting

Jakob Thumm, Marcel Reith-Braun, Florian Pfaff, Uwe D. Hanebeck, Merle Flitter, Georg Maier, Robin Gruna, Thomas Längle, Albert Bauer, and Harald Kruggel-Emden

***Abstract*—In optical sorting of bulk material, the composition of particles may frequently change. State-of-the-art sorting approaches rely on tuning physical models of the particle motion. The aim of this work is to increase the prediction accuracy in complex, fast-changing sorting scenarios with data-driven approaches. We propose two neural network (NN) experts for accurate prediction of a priori known particle types. To handle the large variety of particle types that can occur in real-world sorting scenarios, we introduce a simple but effective mixture of experts approach that combines NNs with hand-crafted motion models. Our new method not only improves the prediction accuracy for bulk material consisting of many particle classes, but also proves to be very adaptive and robust to new particle types.**

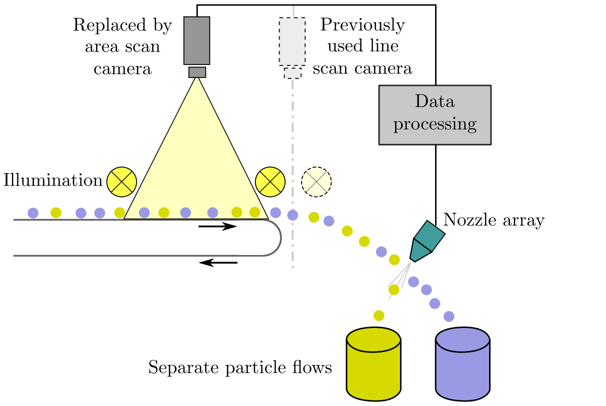


Fig. 1: Schematic setup of an optical belt sorter with an area scan camera, based on [4].

I. Introduction

THE automatic sorting of bulk material is a key technology in many industrial sectors such as civil engineering [1], recycling [2], and agriculture [3]. Optical sorters are of special importance as their sorting decision can be made solely according to visual properties, allowing the sorting of almost any material type as long as the particles that should be separated from the stream of bulk material can be distinguished visually from the other particles. This prevents significant harm to the sorting material like water or heat damage.

A standard layout of an optical sorter consists of a transport unit, a line scan camera, and a separation mechanism, as shown in Fig. 1. Often, a conveyor belt is used as a transport unit to achieve a relatively homogeneous particle motion pattern. The particles are detected and classified using a line scan camera at the end of the belt. The separation mechanism then ejects unwanted particles into a secondary stream or container with short bursts of high-pressure air. The air nozzles are selectively activated with an empirically determined time delay after the detection of a particle in the line scan camera. A certain delay is required to account for the detection, classification, and nozzle activation time. Since a single image of the line scan camera yields little to no information about the velocity of a particle, it is assumed that all particles move in a straight line to the nozzle array, and no lateral motion is considered.

Our previous work [5], [4] showed that this setup can be significantly improved by replacing the line scan camera with an area scan camera, as displayed in Fig. 1. In our *predictive tracking* approach, motion information about the individual particles can be derived from the images by tracking the particles over the course of the observable area with the help of a Kalman filter (KF). The additional motion information is then used in combination with linear motion models to predict the time and position of a particle at the nozzle array (*separation prediction*). However, so far, our approach relied on manual fine-tuning of the tracking model parameters. Additionally, we observed that certain particles tend to have a nonlinear motion behavior for which hand-crafted models are difficult to derive. In scenarios with varying particle types and complex motion behaviors, a separation prediction with a single motion model may therefore be insufficient. To mitigate these shortcomings, this article presents an advanced separation prediction with a new data-driven mixture of experts (ME) concept[1].

In the first step, we present two new neural network (NN) experts for precise particle tpye-specific predictions, a multilayer perceptron (MLP) and a long short-term memory (LSTM). These experts perform best when trained for a very specific task. Therefore, we utilize the natural separation of our data sets to generate particle type-specific

Jakob Thumm was with the Department of Intelligent Sensor Actor Systems, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, e-mail: jakob.thumm@tum.de.

M. Reith-Braun, F. Pfaff, and U. D. Hanebeck are with KIT.

Manuscript received Month dd, yyyy; revised Month dd, yyyy.

[1] Our source code and data are available at https://github.com/KIT-ISAS/TrackSort_Neural_Public/tree/TII2021 and https://doi.org/10.5281/zenodo.5506551.

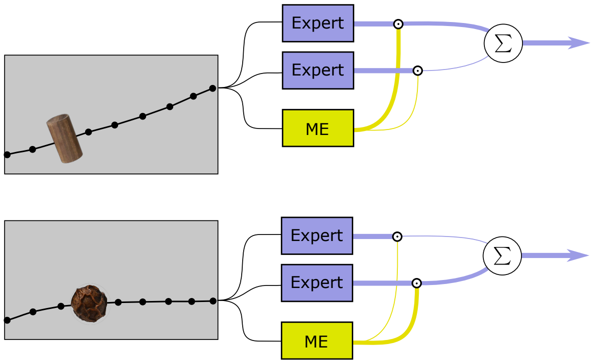


Fig. 2: Concept of generating weighted combinations with an ME gating network. The best experts for each individual particle are selected based on their motion behavior.

experts. However, the NNs tend to lack generalization in new, unseen scenarios. Hence, in the second step, we present our ME approach that selects the best fitting experts in every situation without the need of any manual calibration. In ME, a gating network, usually an MLP, assigns weights to the experts depending on their suitability to explain the current situation. The weights are then used to compute a weighted sum of experts' predictions, as sketched in Fig. 2. Previous work based on the ME approach mostly focused on combining several similar experts, such as KFs with other KFs [6], [7], [8] or NNs with other NNs [9], [10], [11], [12]. Using such homogeneous sets of experts involves the risk that the experts have common weaknesses, which cannot be compensated for. In addition, most ME models train the experts and the gating network simultaneously, thus requiring a complex optimization that is reported to prevent the ME model from reaching its full potential [13]. Our novel approach combines both the previously developed motion models and the pre-trained NN experts, thus leading to a high prediction accuracy as well as a good generalization capability. Our approach is particularly useful for engineering tasks in which useful models are already known. With our easy-to-add model combination method, we ensure a high interpretability with a new level of prediction accuracy.

A. Contributions

Our major contributions are threefold. First, we present two data-driven NNs that increase the sorting accuracy for a given particle type in comparison to the previously used linear motion models. Second, we propose to use an ME gating network to combine motion models and NN experts, which leads to a very high prediction accuracy. Third, and most significant, we show that combining motion models and previously trained NNs with our ME approach leads to high accuracy in new, previously untrained cases like new particle types. This mitigates the lack of generalization capability typical of NN experts while still maintaining validity and interpretability. To our best knowledge, the proposed method is the first to combine physical motion models with NNs trained in advance, as previous approaches only investigated the combination of multiple NNs or multiple motion models, or trained all models simultaneously.

B. Article structure

The following subsection summarizes the theoretical background behind ME. Sec. II presents the latest advances of ME and explains the predictive tracking approach in detail. Sec. III discusses the new separation prediction experts and the ME approach. We compare the performance of the new experts with that of linear motion models on our publicly available data set of real-world sorting scenarios in Sec. IV. Sec. V summarizes our main contributions and gives a brief outlook to further research goals.

C. Background

In this section, we briefly review the related background in combining models, which leads to so-called ensembles or committees and their extensions, the ME. To be consistent with the ME literature, we refer to the models as experts and to their outputs as predictions.

Assuming a set of L experts, where $\hat{y}_i(\underline{x})$ denotes the prediction of expert i for an input $\underline{x}$, a combined prediction can be obtained by the weighted sum of expert predictions

$$\hat{y}(\underline{x}) = \sum_{i=1}^{L} w_i \, \hat{y}_i(\underline{x}), \quad \sum_{i=1}^{L} w_i = 1. \tag{1}$$

Usually, the weights are restricted to be positive, guaranteeing a convex combination. The weighting can be chosen based on two fundamentally different strategies for combining the experts: fusion and selection. While the final output $\hat{y}(\underline{x})$ is constructed considering the predictions of all experts in the first strategy, it is produced based on the predictions of only one or a few experts that seem to be most appropriate for the current situation in the second strategy.

A simple approach that combines several models based solely on the fusion strategy assigns equal weights $w_i = 1/L$ to all experts. This is motivated by the fact that even for this very simple combination method, one can show that the combined models' mean squared error (MSE) is generally smaller than the averaged squared error of all experts [14]. We will refer to this as a *simple ensemble* (SE) and will use it as a benchmark in our evaluations.

It is inherently desirable to assign higher weights to experts that make better predictions. To assess the expert performance, we can calculate a measure of error on the training data set called *symmetric mean absolute percentage error* (SMAPE) [15], which is defined as

$$\hat{e}_i^{\text{SMAPE}} = \begin{cases} 0, & y^n, \hat{y}_i^n = 0 \\ \dfrac{200}{N} \cdot \displaystyle\sum_{n=1}^{N} \frac{|\hat{y}_i^n - y^n|}{|\hat{y}_i^n| + |y^n|}, & \text{otherwise,} \end{cases}$$

where $\hat{y}_i^n$ is the prediction for sample n, y^n is the corresponding ground truth, and N is the number of

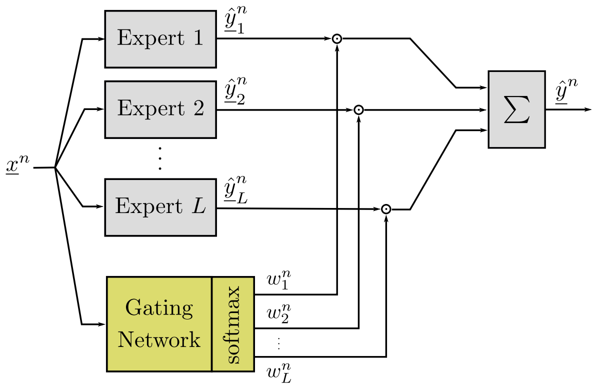


Fig. 3: ME gating network and experts structure, based on [17].

training samples. Based on the SMAPE, the weights can be calculated according to [16]

$$w_i^{\mathrm{sqr}} = \frac{\left(\frac{1}{\hat{e}_i^{\mathrm{SMAPE}}+\epsilon}\right)^2}{\sum_{j=1}^{L}\left(\frac{1}{\hat{e}_j^{\mathrm{SMAPE}}+\epsilon}\right)^2},$$

using a small constant ϵ to prevent a division by zero. This guarantees that (1) is a convex combination and that experts with comparatively low SMAPE are given high weights. One limitation of the aforementioned approaches is that they assign weights independently of the current input, and thus, are not able to adapt to changes in the sorting scenario.

One way to overcome this limitation is with an ME gating network, an NN that is trained to assign weights to the experts based on its input. Hereby, the weighted sum in (1) changes to

$$\hat{y}^n(\underline{x}) = \sum_{i=1}^{L} w_i^n(\underline{x})\, \hat{y}_i^n(\underline{x}). \tag{2}$$

The exemplary structure of an ME, as it was originally designed in [17], is displayed in Fig. 3. The ME gating network uses a *softmax* activation function in its output layer to guarantee that (2) is a convex combination. The original ME method trains the experts and the gating network in parallel using maximum likelihood via the expectation–maximization (EM) algorithm. In the “E”-step, the expected likelihood functions depending on the outputs of the gating network are built so that these likelihoods can then be maximized in the “M”-step.

Figuratively, the ME separates the input space into multiple sub-spaces and trains the experts to specialize in these sub-spaces, thus utilizing a divide-and-conquer strategy. Note that in this framework, the gating network essentially learns to cluster the input space based on input-output relationships rather than on the input space alone. The design of the gating network determines whether the ME tends towards fusion or specialization. Because the softmax function can be seen as a smooth version of a “winner-takes-all” model, the original ME strongly encourages highly specialized experts, which may be suboptimal for certain applications. In order to achieve a more balanced relationship between fusion and selection, multiple modifications of the gating network and its loss function have been proposed [13], for example by including a regularization term for the gating network [18] or using hierarchical ME structures [19] with multiple levels of gating.

As opposed to the original ME and its extensions, which are often referred to as mixture of implicitly localized experts (MILE) models, there exist other approaches that first explicitly divide the input space into sub-spaces, e.g., using clustering algorithms or prior knowledge, and then train the experts separately on each subspace. These models are often called mixture of explicitly localized experts (MELE) models. Although they do not take the input–label relationship of the data in the clustering step into account, MELE algorithms are often reported to have a better performance compared with MILE algorithms. This is due to a clearer distinction between the experts' responsibilities, which can lead to a better generalization ability [13].

II. State of the art

A. Mixture of experts

Over the last 30 years, the ME approach has continuously been developed and widely applied in many research fields, including supervised machine learning, control, and reinforcement learning. In this section, we briefly review the latest advances, cite recent applications, and present the works that are closest to ours.

On the theoretical side, various alternative learning methods and ME structures have been proposed, which are summarized in detail in [20]. Recently, a globally consistent algorithm for MILE training was proposed by [21], which does not suffer from the risk of getting stuck in local optima for synthetic data generated by a wide class of mixture models. To remove the need for the inner optimization loop required when performing the ”M”-step of standard MILE training, [22] proposed a novel inference algorithm with closed-form parameter updates.

Since the development of deep MEs by [23], the ME approach has been attracting increasing interest in the deep learning community. In deep ME, multiple ME models, each consisting of several MLP experts, are applied sequentially. This allows the model to split complex tasks into sequential subtasks, each of which can be solved by the current best expert. The idea of stacked MEs was reused by [9] and [10] to increase the number of network parameters without increasing computational effort in extremely large NNs for language modeling and image recognition, respectively. Both report improved results at lower computational costs. More recently, [24] even created a trillion parameter model using an ME-based switch transformer approach. However, as a result of the latest developments of huge NNs, most researchers can no longer afford to train state-of-the-art NNs for certain applications. To address this problem via crowdsourcing, [25] proposed an NN architecture based on a decentralized ME that distributes the training to loosely connected experts running on distributed hardware.

In addition to deep learning, ME has also been used to enhance control and reinforcement learning algorithms. For example, [11] used a MELE model with MLP experts in nonlinear optimal control to account for discontinuities caused by non-convexity or control switching in the mapping from problem parameters to optimal solutions. In reinforcement learning, ME models are utilized to represent the actor in model-free reinforcement learning [12] and the environment model in model-based reinforcement learning [26], respectively. In the latter work, multiple physical contact models are combined with data-driven MLPs in a two-stage hierarchical ME for robot arm control.

Whereas most of the cited work relies on data-driven experts, often in combination with a MILE model, also KFs and their variants have been used in the processing of sequential data [6], [7]. For example, [6] used an ME gating network with adaptive KF experts for state estimation and tracking. In this work, the ME approach outperformed the well-known Magill filter bank in fast-changing scenarios by constantly selecting the best filter in the bank and continuously changing the KF experts' parameters using an EM algorithm.

Recent applications of ME are in the field of representing light field images to generate photorealistic VR camera-captured scenes [27] and acidity prediction in a polymerization batch process [18]. With regard to MELE models, [28] developed an ME model based on kriging and radial basis function experts for forecasting aircraft fuel burn, and [8] used extended and unscented KF experts for resident space object tracking.

Almost all previous ME approaches are based on the combination of rather similar experts, e.g., solely MLPs or filter algorithms. The only work combining physical motion models and NNs using ME is [26]. However, since they are using ME in reinforcement learning, they rely on a MEME approach. We try to avoid this for our supervised learning problem, because of the discussed disadvantages, such as poorer accuracy, low interpretability, and high complexity in comparison to MELE approaches. To the best of our knowledge, there is no approach that combines the advantages of well-generalizing but often less accurate physical models with highly specialized data-driven experts using a MELE model.

B. Predictive tracking with Kalman filters

Our previous work [5] focused on applying KFs on the centroid data for predictive tracking. In the tracking of particles along the observable area (*tracking phase*), linear, physically motivated models such as the constant velocity (CV) and constant acceleration (CA) model are used. Since multiple particles are closely spaced in the field of view, techniques from multitarget tracking need to be used. In each time step, the most likely association between particle tracks and measurements is determined using the uncertainties of the predicted particle positions and new measurements. In order to account for the prediction calculation time and nozzle activation delays, the end of the tracking phase needs to be located at a sufficient distance to the separation mechanism. We refer to the phase after the tracking phase and before the actual separation as *prediction phase*. The separation prediction bridges the entire prediction phase. The separation prediction can be done individually for each particle. Therefore, we do not address the tracking phase and refer the interested reader to [5].

As we proposed in [4], the last updated KF state of a particle before the beginning of the prediction phase $\underline{x}_{\mathrm{Last}} = [\mathsf{x}_{\mathrm{L}}, \dot{\mathsf{x}}_{\mathrm{L}}, \mathsf{y}_{\mathrm{L}}, \dot{\mathsf{y}}_{\mathrm{L}}]^\top$ (CV) or $\underline{x}_{\mathrm{Last}} = [\mathsf{x}_{\mathrm{L}}, \dot{\mathsf{x}}_{\mathrm{L}}, \ddot{\mathsf{x}}_{\mathrm{L}}, \mathsf{y}_{\mathrm{L}}, \dot{\mathsf{y}}_{\mathrm{L}}, \ddot{\mathsf{y}}_{\mathrm{L}}]^\top$ (CA) can be used to predict the time $\hat{t}_{\mathrm{N}}$ and y-position $\hat{\mathsf{y}}_{\mathrm{N}}$ of the particle at the nozzle array[2]. For the CV approach, this leads to

$$\hat{t}_{\mathrm{N}} = t_{\mathrm{L}} + \frac{1}{\dot{\mathsf{x}}_{\mathrm{L}}} (\mathsf{x}_{\mathrm{N}} - \mathsf{x}_{\mathrm{L}}), \quad \hat{\mathsf{y}}_{\mathrm{N}} = \mathsf{y}_{\mathrm{L}} + \dot{\mathsf{y}}_{\mathrm{L}} \left(\hat{t}_{\mathrm{N}} - t_{\mathrm{L}}\right), \tag{3}$$

where the x-position of the nozzle array x_{N} is assumed to be known and t_{L} is the time of the last measurement before the prediction phase. For the CA model, $\hat{t}_{\mathrm{N}}$ is calculated by solving

$$\mathsf{x}_{\mathrm{N}} = \mathsf{x}_{\mathrm{L}} + \left(\hat{t}_{\mathrm{N}} - t_{\mathrm{L}}\right) \cdot \dot{\mathsf{x}}_{\mathrm{L}} + \frac{1}{2} \left(\hat{t}_{\mathrm{N}} - t_{\mathrm{L}}\right)^2 \cdot \ddot{\mathsf{x}}_{\mathrm{L}} \tag{4}$$

for $\hat{t}_{\mathrm{N}}$. The y-position is then calculated by inserting $\hat{t}_{\mathrm{N}}$ into

$$\hat{\mathsf{y}}_{\mathrm{N}} = \mathsf{y}_{\mathrm{L}} + \left(\hat{t}_{\mathrm{N}} - t_{\mathrm{L}}\right) \cdot \dot{\mathsf{y}}_{\mathrm{L}} + \frac{1}{2} \left(\hat{t}_{\mathrm{N}} - t_{\mathrm{L}}\right)^2 \cdot \ddot{\mathsf{y}}_{\mathrm{L}}. \tag{5}$$

To further improve the predicted times and positions at which the particles reach the nozzle array, several adjustments to the motion models in (3), (4), and (5) were introduced in [4]. These models account for nonlinear motion behavior due to interaction with the belt or free-flight phase by analyzing how previous particles behaved in the prediction phase. Here, we briefly explain the model adaptions that led to the best prediction results in our real-world tests. The *constant velocity with bias correction* (CVBC) model corrects the bias of the temporal prediction of the CV model by subtracting the average prediction error over a training set from the predictions in the test data. The *ratio*-based approach can additionally be used to improve the spatial CV model prediction. Hereby, it is assumed that the ratio r of the remaining velocity when reaching the separation mechanism $\dot{\mathsf{y}}_{\mathrm{N}}$ and the velocity at the start of the prediction phase $\dot{\mathsf{y}}_{\mathrm{L}}$ ($r = \dot{\mathsf{y}}_{\mathrm{N}}/\dot{\mathsf{y}}_{\mathrm{L}}$) is identical for all particles. Based on the above ratio, an acceleration is calculated that ensures that only a share of r of the velocity remains when the particle reaches the separation mechanism.

The comparison of the tracking-based motion models with the previously used line scan camera-based approach in [4] on noise-free simulation data shows that the new models were able to make almost perfect temporal predictions, while 50% of the predictions of the line scan camera-based approach had an error larger than 1 ms. Since the separation prediction is always based on the last updated

[2] We refer to the coordinate in transport direction as x and orthogonally to it as y.

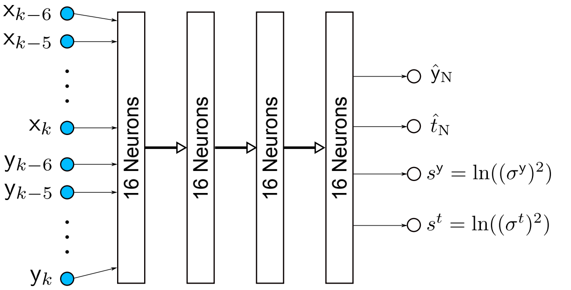


Fig. 4: Structure of the new MLP expert.

KF state, the prediction highly varies with the chosen KF parameters. Therefore, the tremendous increase in the prediction accuracy of the KF-based predictive tracking approach comes with the disadvantage of manual parameter fine-tuning.

C. Advanced tracking with long short-term memory

To overcome the need for manually setting the parameters of the KF, we proposed a tracking approach with a recurrent NN in [29]. The KF in the predictive tracking approach is replaced by an LSTM network that is trained to predict the particle position in the next time step. To obtain uncertainties for the particle–track association in the tracking phase, we estimated the aleatoric uncertainties by not only predicting the x- and y-positions, but also the log-variances $s_k^n = \ln\big((\sigma_k^n)^2\big)$ in every time step k. The network was trained by maximizing the prediction likelihood via the negative log likelihood (NLL) loss function

$$L \propto \frac{1}{N}\sum_{n=1}^{N}\frac{1}{T_n-1}\sum_{k=2}^{T_n}\exp(-s_k^n)\left\|\underline{\hat{y}}_k^n\left(\underline{\hat{x}}_{k-1}^n\right)-\underline{y}_k^n\right\|^2+s_k^n,$$

where N is the number of tracks with varying number of measurements T_n, $\underline{y}_k^n = [\mathsf{x}_k^n, \mathsf{y}_k^n]^\top$ is the ground truth position of the particle with index n in time step k, and $\underline{\hat{y}}_k^n\left(\underline{\hat{x}}_{k-1}^n\right)$ is the corresponding predicted position. The input to the network $\underline{\hat{x}}_{k-1}^n$ is the last measured particle position. In single-target prediction tests on simulated data, the NLL LSTM was slightly worse than the KF approach. Only when a high artificial noise was added, the NLL LSTM outperformed the KF.

III. Methodology

This section presents our approaches to improve the separation prediction of optical belt sorters. The first section covers two new data-driven experts for the separation prediction. The second section then describes how all expert predictions can be combined using an ME approach to improve the sorting accuracy in changing scenarios. Note that the algorithm for tracking the particles on the belt, as described in Sec. II-B, remains unchanged, and we only consider the prediction to the nozzle array in this paper.

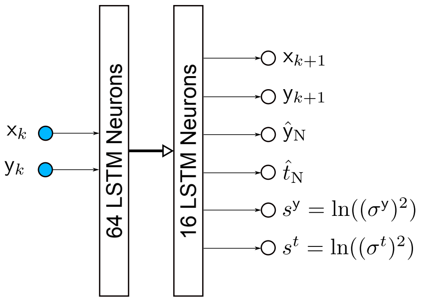


Fig. 5: Structure of the new LSTM expert.

A. Advanced prediction experts

The first new data-driven expert is an MLP that takes the last q measurements before the prediction phase as input and predicts the time t_N and position y_N of the particle at the nozzle array. Since the loss incorporates both a prediction in time and space, both t and y are normalized by constant factors k^t and k^y in the data preprocessing to assure unitless variables, so that $t_\mathrm{Norm} = k^t t\,\mathrm{s}^{-1}$ and $\mathsf{y}_\mathrm{Norm} = k^\mathsf{y}\mathsf{y}\,\mathrm{m}^{-1}$. For the sake of brevity, we will use t and y instead of t_Norm and y_Norm in this section. Our tests have shown that the prediction accuracy of the MLP can be increased by also predicting the aleatoric uncertainties. Therefore, we add two outputs for the estimation of the log variances $s_n^t = \ln\big((\sigma_n^t)^2\big)$ and $s_n^\mathsf{y} = \ln\big((\sigma_n^\mathsf{y})^2\big)$ of t_N and y_N. The resulting network is displayed in Fig. 4. The loss function of the MLP is defined proportionally and with constant offset to the NLL so that minimizing the loss maximizes the data log likelihood

$$\begin{aligned}\mathrm{L} = \frac{1}{N}\sum_{n=1}^{N}&\exp\big(-s_n^t\big)\left\|t_\mathrm{N}^n-\hat{t}_\mathrm{N}^n\right\|^2+s_n^t\\&+\exp(-s_n^\mathsf{y})\left\|\mathsf{y}_\mathrm{N}^n-\hat{\mathsf{y}}_\mathrm{N}^n\right\|^2+s_n^\mathsf{y}.\end{aligned}$$

However, we are constrained by the requirement of the MLP to have fixed-sized inputs. Therefore, the network is not able to take the whole track into account and is not applicable if less than q measurements are available. In our case, a good trade-off between the amount of predictable tracks and a high prediction accuracy is $q = 7$.

To overcome these restrictions, we introduced a recurrent NN as our second new expert. The LSTM expert tracks a particle over the entire belt length and, when reaching the prediction phase, predicts the time t_N and position y_N of the particle at the nozzle array. For this, we add six outputs to the network, two for tracking the particle in the tracking phase (x- and y-position at the next time step), two for the $\hat{t}_\mathrm{N}$ and $\hat{\mathsf{y}}_\mathrm{N}$ prediction, and two for the log-variances in t_N and y_N. The network structure is visualized in Fig. 5.

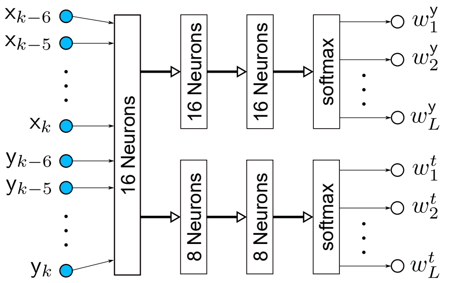


Fig. 6: Structure of the ME gating network. The network outputs one weight per expert and dimension (temporal and spatial).

The total loss function

$$
\begin{aligned}
\mathrm{L} =& \frac{1}{N}\sum_{n=1}^{N}\big(\exp\big(-s_n^t\big)\,\big\|t_{\mathrm{N}}^n - \hat{t}_{\mathrm{N}}^n\big\|^2 + s_n^t \\
&\quad + \exp(-s_n^{\mathsf{y}})\,\|\mathsf{y}_{\mathrm{N}}^n - \hat{\mathsf{y}}_{\mathrm{N}}^n\|^2 + s_n^{\mathsf{y}} \\
&\quad + \frac{1}{T_n-1}\sum_{k=2}^{T_n}\|\mathsf{y}_k^n - \hat{\mathsf{y}}_k^n\|^2 + \|\mathsf{x}_k^n - \hat{\mathsf{x}}_k^n\|^2\big)
\end{aligned}
$$

consists of two parts, a separation prediction NLL loss, as described previously, and a tracking loss. The $\hat{t}_{\mathrm{N}}$ and $\hat{\mathsf{y}}_{\mathrm{N}}$ predictions are made in every time step, but only the prediction in the last time step before the prediction phase is taken into account in the loss function. The additional tracking loss helps the LSTM build up a useful internal state. The tracking loss can be changed to an NLL tracking loss as described in Sec. II-C if this network should be used in the tracking phase. However, since the tracking performance with a KF approach is better than with an LSTM, we stick with the KF for the tracking phase.

B. *Expert combination methods*

The idea of an expert combination approach is to achieve higher prediction accuracy and a better adaption to new scenarios with the right selection of weights. For the expert combination in the separation prediction, we use separate weights for the spatial and temporal prediction. Thus, the combined predictions are

$$
\hat{\mathsf{y}}_{\mathrm{N}} = \sum_{i=1}^{L} w_i^{\mathsf{y}}\,\hat{\mathsf{y}}_{\mathrm{N},\,i}, \quad \hat{t}_{\mathrm{N}} = \sum_{i=1}^{L} w_i^t\,\hat{t}_{\mathrm{N},\,i}. \tag{6}
$$

We propose to use an ME gating network to learn the weights based on the motion information of the particles in the training set. Since our data already have a natural separation in the form of particles types, we use an MELE model and train each expert type on every data type to create our set of experts. As discussed in Sec. I-C, the pre-definition of input clusters leads to good generalization capability of our approach, which will also be shown in the results section. Our ME gating network structure is based on an MLP, which takes the last q measurements as inputs and outputs the weights for each expert in the temporal and spatial domain. The network with the same $q = 7$ inputs as the MLP expert is displayed in Fig. 6. We noticed that a fork of the network in a spatial and a temporal branch after the first layer is slightly beneficial for the prediction accuracy. To formulate the loss function, we first calculate the combined prediction using the expert predictions and the predicted weights according to (6). Then, we calculate the MSE loss from the combined prediction with

$$
\mathrm{L} = \frac{1}{N}\sum_{n=1}^{N}\big\|t_{\mathrm{N}}^n - \hat{t}_{\mathrm{N}}^n\big\|^2 + \|\mathsf{y}_{\mathrm{N}}^n - \hat{\mathsf{y}}_{\mathrm{N}}^n\|^2\,.
$$

The expert and ME networks are trained in Python with TensorFlow and an Adam optimizer. We use an exponential learning rate decay $\eta_k = \eta_0 \cdot \lambda^{\frac{k}{\zeta}}$, where η_k is the learning rate in epoch k, and λ and ζ define the rate of the decay. For training each model, we used $\eta_0 = 0.005$, $\zeta = 500$, and $\lambda = 128$ on a batch size of 128 tracks. The number of training epochs for the MLP, LSTM, and ME gating network were 4000, 1000, and 5000, respectively. The optimal model structure and training parameters were determined using a hyperparameter search with the focus on a balance between prediction accuracy and generalization capability. In the sorting application, it is important that the selected hyperparameters work well for data sets involving arbitrary particle types. By choosing robust parameters, a new hyperparameter search for every new particle type can be avoided. We provide a more detailed description of the hyperparameters and their effects on the prediction accuracy in Appendix B.

IV. Results

In this section, we present the accuracy of our new methods in comparison with the previously used motion models in tests on real-world measurements. The first section covers the key properties of the considered data and their acquisition. The second section presents the evaluation of single-particle-type and multi-particle-type scenarios as well as a robustness test.

A. *Data description*

The experiment setup with the transport belt, area scan camera, and high-pressure nozzle array is based on the detailed descriptions in [30]. The area scan camera has a resolution of 2320 × 1726 pixels, and acquires data with 200 Hz. Its field of view has a size of approximately 130 mm × 96.7 mm. The particles are detected by filtering the images for color and size. The actual particle tracks are then obtained with the multitarget tracker described in Sec. II-B. The data include the tracks of 7712 spheres, 7170 peppercorns, 19 200 cylinders, and 8702 wheat grains. Additional information on the distribution of positions and velocities of the tracks of the four materials as a function of time step can be found in Appendix A.

The video data of the particles at the nozzle array are difficult to obtain because the nozzle array obstructs the view. Therefore, we only use video footage of the transport belt to test our methods. For this, we start the prediction phase early in the observable area and consider the prediction to a specified line in the image and act

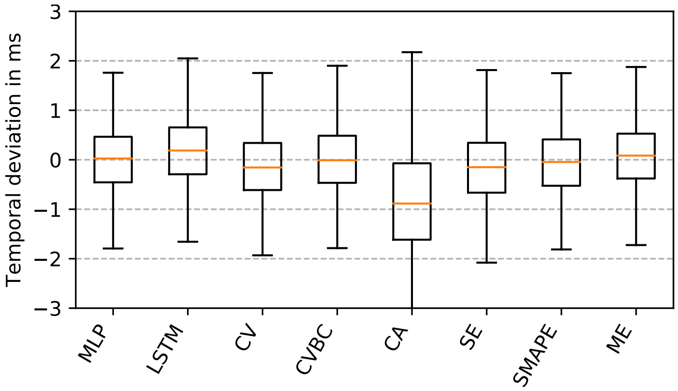


Fig. 7: Temporal prediction error in ms of all possible models on real-world peppercorn data.

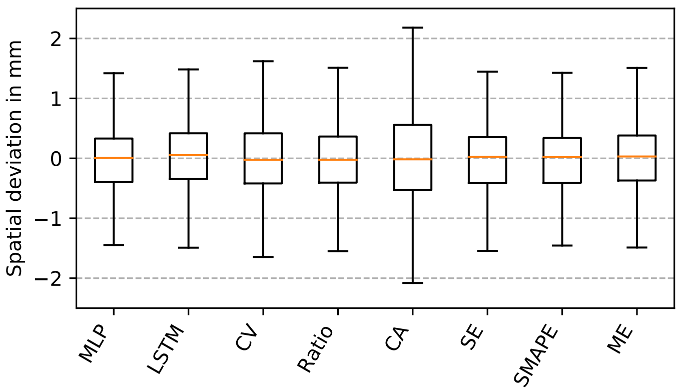


Fig. 8: Spatial prediction error in mm of all possible models on real-world peppercorn data.

as if the nozzle array was covering that line. The start of the prediction phase in our setup is at 800 pixels and the specified line is located at 1550 pixels, resulting in a corresponding prediction phase length of approximately 4.2 cm.

While our ground truth data contain observations during the prediction phase and even after passing the nozzle array x_N, we have no observation at precisely x_N. Therefore, we perform a linear interpolation between the last observation before x_N and the first observation thereafter, which implies the assumption that the velocity is constantly the average velocity between the two points. With these two data points, which are approximately 5 mm apart, we obtain an approximation of the true x_N value. The accuracy of this interpolation is much higher than the prediction accuracy of the experts on a length of 42 mm.

All spatial values are normalized by the image width, so that $k^\mathsf{y} = 1/0.13$. To achieve approximately the same magnitude of prediction errors in the temporal as in the spatial domain, we set the temporal normalization factor to $k^t = 1/16$. In order to make the results more interpretable, the values t_N and y_N are denormalized and converted from pixel and frames to mm and ms for the presentation of the evaluation results.

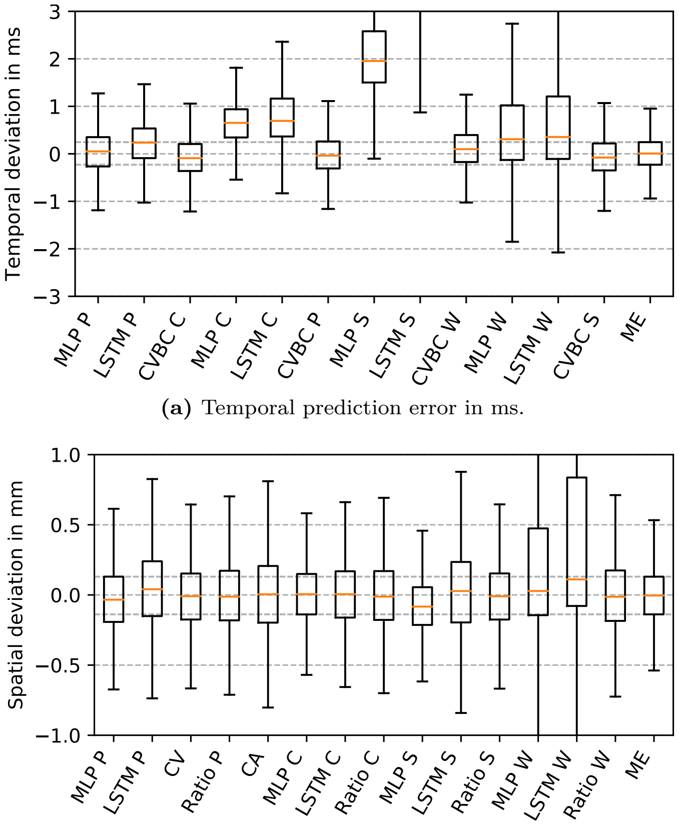
We visualize the accuracy of the predictions using boxplots showing the prediction errors $e^t = \hat{t} - t$ and $e^\mathsf{y} = \hat{\mathsf{y}} - \mathsf{y}$. The boxes range from the 25%- to the 75%-quantile. The upper and lower whiskers mark the last value in the interquartile range (which ranges from the first to the third quartile) scaled by 1.5. The use of boxplots not only allows a detailed comparison of the prediction accuracy but also helps to detect biases.

B. Results

1) Single particle type evaluation: We first evaluate the model performance in scenarios with just one particle type. Since the peppercorn data set has the most diverse motion behaviors, the results are discussed on this particle type. The evaluation of the temporal prediction error in Fig. 7 indicates that the MLP and CVBC models make the most accurate predictions without any biases. The MLP is slightly better than all other models. The LSTM has a slight temporal prediction bias, which indicates a bad generalization of the network. The CA model is overall less accurate than its CV counterparts in all tests. A combination of expert predictions with the SE approach is not beneficial since it incorporates the prediction bias of bad models. In this scenario, both the ME and SMAPE approaches are good combination methods that provide roughly equally accurate predictions as the best expert.

The evaluation of the spatial predictions in Fig. 8 shows a very similar behavior. Again, the MLP is the best model, the LSTM is slightly worse than the MLP, and the CA approach is unsatisfactory. The ratio-based model yields slightly better results than the CV model. The ME and SMAPE approach are again both good combination methods with prediction accuracies close to those of the best expert. To summarize, the NNs outperform the KF-based approaches in such scenarios without the need for any manual parameter fine-tuning or modelling. There is no disadvantage of using our new ME approach in simple scenarios, but it does not lead to a major increase in the prediction accuracy, either.

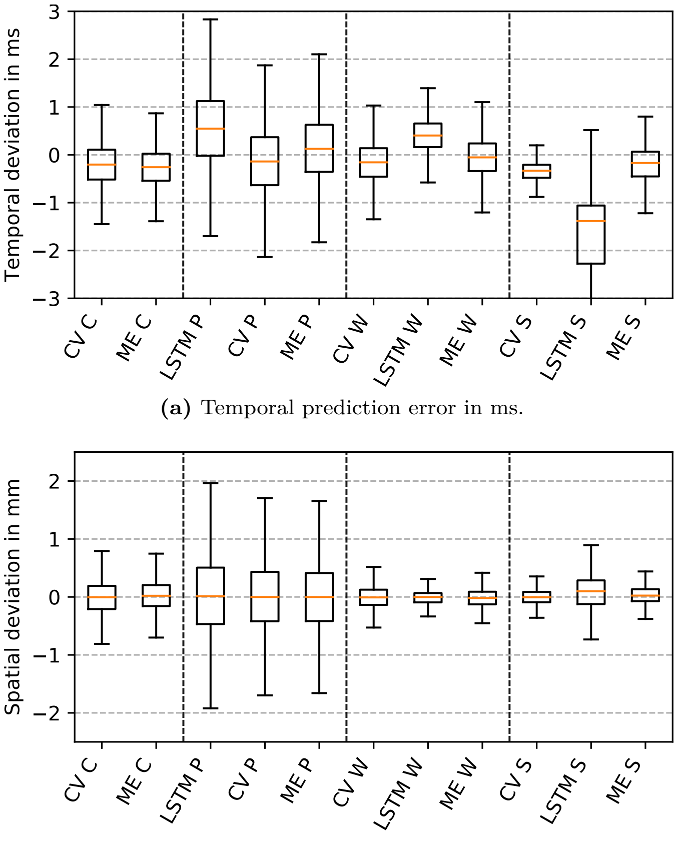
2) Mixture of particle type evaluation: To evaluate the accuracy of the ME approach in a diverse mixture of particle types, we train all types of models on all four different particle type data sets—peppercorn, cylinder, spheres, and wheat grain. Then, we train the ME gating network on the combined data set of all four particle types to weight between the peppercorn, cylinder, sphere, and wheat grain experts. All models and the ME approach are tested on the mixed data set. It should be noted that we tested all additional motion model experts presented in [4], leading to 48 experts in total (12 per particle type), which the ME has to assign weights to. However, we only show a selection of experts based on accuracy, significance, and uniqueness to improve the visualization of the results. The left-out experts are not significant for the ME performance. The results in Fig. 9 show that especially the NN approaches do not generalize well and have problems to make accurate bias-free predictions on the combined data set. The CV-based approaches are generally more robust than the MLP and LSTM. Our ME approach is



(a) Temporal prediction error in ms.

(b) Spatial prediction error in mm.

Fig. 9: The evaluation results for the temporal (a) and spatial (b) separation prediction for the combined data set including cylinders, peppercorns, spheres, and wheat grains. The ME gating network weighs between a large set of experts and improves the overall prediction accuracy significantly.



(a) Temporal prediction error in ms.

(b) Spatial prediction error in mm.

Fig. 10: Robustness test of the ME model. The previously discussed models are trained on three data sets and tested on the fourth. The letters (P)eppercorn, (C)ylinder, (S)pheres, and (W)heat stand for the set the models are tested on. The ME is compared with the standard CV motion model and the LSTM expert trained on the cylinder data set. Displayed are the temporal (a) and spatial (b) separation prediction accuracies.

able to select the best experts out of a fairly large number of experts (48) based on the motion behavior of each particle to achieve a precise, bias-free prediction in this complex sorting scenario.

The prediction time of each MLP, LSTM, or ME gating network is approximately 5 ms and of a classical motion model 0.25 ms per particle[3]. All 48 experts and the ME gating network therefore have a combined prediction time of approximately 55 ms per particle. The calculation time can be greatly sped up by parallelizing the expert predictions, merging all NN experts into one TensorFlow graph, or upgrading the hardware.

3) Leave-one-out robustness test: Last, we tested how our ME approach performs on new, previously unseen data. As a robustness test, we train every previously discussed expert on each of three out of four data sets. Then, we train the ME gating network on the combination of these same three data sets, leaving the fourth out. Finally, we test the accuracy of our approach on the fourth data set. This is repeated four times, always leaving out a different set. Fig. 10 shows this leave-one-out cross-validation in the temporal and spatial domains. The ME approach is compared with the standard CV motion model without any adaptations and the LSTM expert trained on cylinder data. Since the LSTM is trained on cylinders, it is not included in the cylinder test shown in Fig. 10. The cylinder LSTM mostly shows very bad generalization capabilities, except for the spatial prediction on wheat data. The CV model always makes quite good predictions, since it does not rely on any training and therefore has no overfitting effects. The CV model also describes the motion behavior of the spheres very precisely. Our ME approach is better than the very robust CV model in three out of the four cases. In the spheres case, the CV model describes the motion behavior of the particles very precisely. However, our ME approach still makes reasonable predictions in this case. Even though it should occur seldomly that unknown particle types appear, it is still important and remarkable that our ME approach reliably handles these cases. When a particle type that is not in the training data occurs regularly, the ME gating network should be retrained to increase the overall accuracy.

[3] Run on a computer with an Intel Core i5-7600 processor running at 3.50Ghz, 16GB DDR3 RAM, and an Nvidia GTX 1060 6GB VRAM GPU.

V. Conclusion

We showed that our proposed MLP and LSTM experts with NLL loss outperform classical motion models but do not generalize well to varying particle types. In order to benefit from the good prediction accuracy of the NN experts, we proposed an ME approach that learns to weigh between multiple experts based on the motion behavior of the particles. We take advantage of the natural separation in our data sets to train the experts on a specific particle type and use the ME gating network to incorporate the predictions of both NN and physical motion model experts. Our ME approach achieves significantly better results than the single experts in mixed particle type scenarios. We further proved that the ME approach is sufficiently robust for unseen data. With an increasing number of models for different particle types, the ME approach integrates more and more possible motion behaviors and therefore its prediction accuracy is further improved for unseen data. The combination of highly specialized data-driven experts with robust physical motion model using our ME approach has proven to be very beneficial for optical bulk material sorting.

Our future goal is to build a diverse set of experts that covers a wide variety of particle types and combine it with our ME approach. This setup is expected to manage all kinds of real-world sorting situations and is therefore universally applicable in optical bulk material sorting.

Acknowledgment

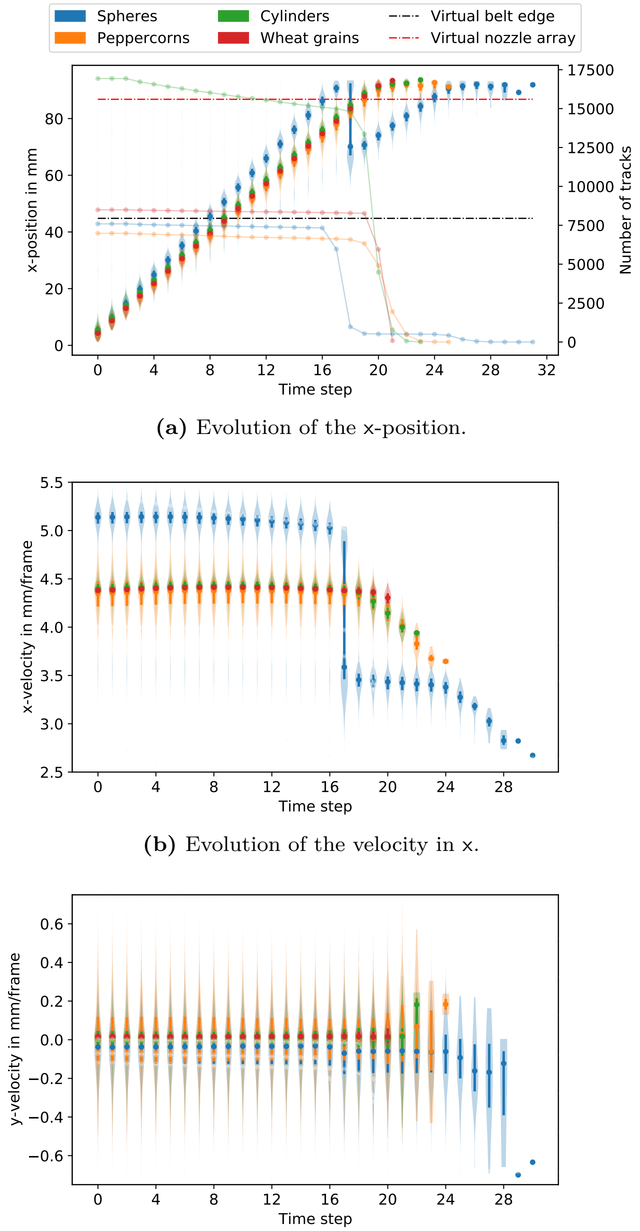
The IGF project 20354 N of the research association Forschungs-Gesellschaft Verfahrens-Technik e.V. (GVT) was supported via the AiF in a program to promote the Industrial Community Research and Development (IGF) by the Federal Ministry for Economic Affairs and Energy on the basis of a resolution of the German Bundestag.

Appendix A
Statistical Data Description

The violin plots in Fig. 11 show the distribution of the particle position and velocity at each time step. The dots in the center of each violin plot depict the median, and the vertical lines represent the interquartile range. The velocities are obtained by calculating the first difference along the corresponding coordinate axis. The number of tracks that have not yet left the camera field of view is displayed in Fig. 11a.

Appendix B
Training Hyperparameters

Network structure and size: The network size should be adequate to the amount of training data. With more training samples, the NNs can be larger. In our case, two hidden layers for the MLP and LSTM experts are a solid choice. For the ME network, a third hidden layer leads to a slightly higher prediction accuracy. The number of neurons in each layer is not a very sensitive hyperparameter. However, the LSTM shows a strong tendency to overfit with 1024 instead of 64 neurons in the first layer. The MLP network structure is overall more robust to overfitting than the recurrent LSTM network structure. The proposed number of neurons per layer proved to be sufficient to learn the motion behavior of all particle types.



(a) Evolution of the x-position.

(b) Evolution of the velocity in x.

(c) Evolution of the velocity in y.

Fig. 11: The evolution of the x-position (a), the x-velocity (b), and the y-velocity (c) on the belt of the four different materials used. The start and end of the prediction phase is denoted by *virtual belt edge* and *virtual nozzle array*.

Learning rate: An exponential decay leads to faster convergence and higher accuracy than a linear decay. The decay parameters are not very sensitive and do not have a major impact on the training.

Number of epochs: The number of training epochs is most important for the LSTM structure, since it has a strong tendency to overfit. If the number of epochs is too low, the model is not fully trained and if it is too high, overfitting occurs. Here, one could also implement early stopping, but using a fixed number of epochs is sufficient in most cases.

Batch size: We tested the batch sizes 32, 64, and 128. The largest batch size eliminates small prediction biases, which can occur for smaller batch sizes.

References

[1] H. R. Thomas *et al.*, "Fundamental Principles of Site Material Management," *Journal of Construction Engineering and Management*, vol. 131, no. 7, pp. 808–815, Jul. 2005.

[2] V. W. Tam *et al.*, "A Review on the Viable Technology for Construction Waste Recycling," *Resources, Conservation and Recycling*, vol. 47, no. 3, pp. 209–221, Jun. 2006.

[3] J. H. Connell, "Almond Harvest Operations in California - Maintaining Nut Quality," *Acta Horticulturae*, no. 373, pp. 241–248, Sep. 1994.

[4] F. Pfaff *et al.*, "Predictive tracking with improved motion models for optical belt sorting," *at - Automatisierungstechnik*, vol. 68, no. 4, pp. 239–255, Apr. 2020.

[5] F. Pfaff, *Multitarget Tracking Using Orientation Estimation for Optical Belt Sorting*, ser. Karlsruhe Series on Intelligent Sensor-Actuator-Systems. Karlsruhe Institute of Technology, 2019, no. 22.

[6] W. Chaer *et al.*, "A Mixture-of-Experts Framework for Adaptive Kalman Filtering," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 27, no. 3, pp. 452–464, Jun. 1997.

[7] A. Ravet *et al.*, "Learning to combine multi-sensor information for context dependent state estimation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Tokyo: IEEE, Nov. 2013, pp. 5221–5226.

[8] J. J. LaPointe, *Adaptive Estimation Techniques for Resident Space Object Characterization.* The University of Arizona., 2016.

[9] N. Shazeer *et al.*, "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017, p. 19.

[10] X. Wang *et al.*, "Deep Mixture of Experts via Shallow Embedding," in *Uncertainty in Artificial Intelligence.* PMLR, Aug. 2020, pp. 552–562, iSSN: 2640-3498.

[11] G. Tang *et al.*, "Discontinuity-Sensitive Optimal Control Learning by Mixture of Experts," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 7892–7898, iSSN: 2577-087X.

[12] Z. Zheng *et al.*, "Self-Supervised Mixture-of-Experts by Uncertainty Estimation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 5933–5940, Jul. 2019, number: 01.

[13] S. Masoudnia *et al.*, "Mixture of experts: a literature survey," *Artificial Intelligence Review*, vol. 42, no. 2, pp. 275–293, Aug. 2014.

[14] C. M. Bishop, *Neural Networks for Pattern Recognition.* Clarendon Press, Nov. 1995.

[15] R. J. Hyndman *et al.*, "Another Look at Measures of Forecast Accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, Oct. 2006.

[16] M. Pawlikowski *et al.*, "Weighted Ensemble of Statistical Models," *International Journal of Forecasting*, vol. 36, no. 1, pp. 93–97, Jan. 2020.

[17] R. A. Jacobs *et al.*, "Adaptive Mixtures of Local Experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, Feb. 1991.

[18] F. Souza *et al.*, "A Regularized Mixture of Linear Experts for Quality Prediction in Multimode and Multiphase Industrial Processes," *Applied Sciences*, vol. 11, no. 5, p. 2040, Feb. 2021.

[19] W. Chaer *et al.*, "Hierarchical adaptive Kalman filtering for interplanetary orbit determination," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 883–896, Jul. 1998.

[20] S. E. Yuksel *et al.*, "Twenty Years of Mixture of Experts," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1177–1193, 2012.

[21] A. Makkuva *et al.*, "Breaking the gridlock in Mixture-of-Experts: Consistent and Efficient Algorithms," in *International Conference on Machine Learning.* PMLR, May 2019, pp. 4304–4313.

[22] A. Sharma *et al.*, "A flexible probabilistic framework for large-margin mixture of experts," *Machine Learning*, vol. 108, no. 8, pp. 1369–1393, Sep. 2019.

[23] D. Eigen *et al.*, "Learning Factored Representations in a Deep Mixture of Experts," in *ICLR Workshop*, 2014.

[24] W. Fedus *et al.*, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," 2021.

[25] M. Ryabinin *et al.*, "Towards Crowdsourced Training of Large Neural Networks using Decentralized Mixture-of-Experts," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 3659–3672.

[26] J. Ahn *et al.*, "Nested Mixture of Experts: Cooperative and Competitive Learning of Hybrid Dynamical System," in *Learning for Dynamics and Control.* PMLR, May 2021, pp. 779–790, iSSN: 2640-3498.

[27] R. Verhack *et al.*, "Steered Mixture-of-Experts for Light Field Images and Video: Representation and Coding," *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 579–593, Mar. 2020.

[28] R. P. Liem *et al.*, "Surrogate models and mixtures of experts in aerodynamic performance prediction for aircraft mission analysis," *Aerospace Science and Technology*, vol. 43, pp. 126 – 151, 2015.

[29] D. Pollithy *et al.*, "Estimating Uncertainties of Recurrent Neural Networks In Application to Multitarget Tracking," in *Proceedings of the 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2020)*, Sep. 2020.

[30] G. Maier *et al.*, "Experimental Evaluation of a Novel Sensor-Based Sorting Approach Featuring Predictive Real-Time Multiobject Tracking," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 2, pp. 1548–1559, Feb. 2021.