

Article

Combining CFD and AI/ML Modeling to Improve the Performance of Polypropylene Fluidized Bed Reactors

Nayef Ghasem 

Department of Chemical and Petroleum Engineering, United Arab Emirates University, Al-Ain 15551, United Arab Emirates; nayef@uaeu.ac.ae

Abstract: Polypropylene is one of the most widely used polymers in various applications, ranging from packaging materials to automotive components. This paper proposes the Computational Fluid Dynamics (CFD) and AI/ML simulation of a polypropylene fluidized bed reactor to reduce reactor loss and facilitate process understanding. COMSOL Multiphysics 6.2[®] solves a 2D multiphase CFD model for the reactor's complex gas–solid interactions and fluid flows. The model is compared to experimental results and shows excellent predictions of gas distribution, fluid velocity, and temperature gradients. Critical operating parameters like feed temperature, catalyst feed rate, and propylene inlet concentration are all tested to determine their impact on the single-pass conversion of the reactor. The simulation simulates their effects on polypropylene yield and reactor efficiency. It also combines CFD with artificial intelligence and machine learning (AI/ML) algorithms, like artificial neural networks (ANN), resulting in a powerful predictive tool for accurately predicting reactor metrics based on operating conditions. The multifaceted CFD-AI/ML tool provides deep insight into improving reactor design, and it also helps save computing time and resources, giving industrial polypropylene plant growth a considerable lift.

Keywords: computational fluid dynamics (CFD); artificial intelligence (AI); machine learning (ML); polypropylene; fluidized bed reactors



Citation: Ghasem, N. Combining CFD and AI/ML Modeling to Improve the Performance of Polypropylene Fluidized Bed Reactors. *Fluids* **2024**, *9*, 298. <https://doi.org/10.3390/fluids9120298>

Academic Editors: Germán Mazza, César Venier, Andrés Reyes Urrutia and Jun Yao

Received: 6 November 2024

Revised: 5 December 2024

Accepted: 14 December 2024

Published: 16 December 2024



Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Polypropylene is one of the world's most popular plastics, used for various products, from packaging to auto parts [1,2]. The increasing pressure to increase production efficiency and product quality drove the development of optimized reactor designs, particularly fluidized bed reactors. These reactors are preferred due to their mixing, heat transfer, and scalability, making them ideal for polypropylene plants on a large scale. However, the intricate interactions of gases and solids, heat and mass transfers, and turbulence can be very challenging in optimizing the reaction and ensuring operational efficiency [3]. Computational Fluid Dynamics (CFD) is now a valuable tool to investigate and model complex systems in fluidized bed reactors. CFD simulations enable engineers to study how different operating parameters (feed temperature, gas composition, catalyst feed rate) affect reactor performance to maximize [4]. However, CFD calculations can be computationally prohibitive, particularly for larger-scale multiphase polypropylene reactor systems. The computational power and time required to run extensive simulations at many different operating modes are often impractical [5]. To solve these problems, artificial intelligence (AI) and machine learning (ML) combined with CFD modeling are promising approaches [1,2,6,7]. Machine learning and artificial neural networks (ANN) can be used to learn from simulations and experiments and thus make faster and more accurate predictions about reactor behavior [8,9]. By combining CFD's detailed physical modeling with AI/ML's predictive capabilities, this hybrid approach enhances the efficiency of reactor design and optimization [1,2,10–13].

In this paper, we present an integrated approach that combines CFD and AI/ML modeling. This approach has the potential to significantly improve the performance of

polypropylene fluidized bed reactors, offering a promising solution for the challenges in our field [13]. We develop a 2D multiphase CFD model using COMSOL Multiphysics 6.2[®], simulating the reactor's gas-solid interactions and fluid flow. The model is rigorously validated against experimental data, demonstrating its high accuracy in predicting gas distribution, particle velocity profiles, and temperature gradients [14–17]. We comprehensively analyze critical operating parameters, including feed temperature, catalyst feed rate, and propylene inlet concentration. This thorough evaluation provides valuable insights into their impact on single-pass conversion efficiency [18]. Additionally, we enhance the CFD model's predictive efficiency by incorporating AI/ML algorithms. By training ANN models on the CFD simulation and experimental data, we create a predictive framework capable of efficiently forecasting reactor performance metrics based on varying operational inputs. This integrated CFD-AI/ML approach reduces computational time. It provides a deeper understanding of the interdependencies between key reactor parameters, offering valuable insights for optimizing reactor design and process efficiency [19,20].

2. Architecture of ANN for Olefin Polymerization

ANNs are powerful computation models; they can learn independently and improve over time as more data are processed [21]. In this work, the ANN model was trained with datasets produced by a computational fluid dynamics (CFD) simulation of olefin polymerization in a fluidized bed reactor. Olefin polymerization is a complicated reaction process used to produce polyethylene and polypropylene at a pilot or an industrial scale [22]. This process is challenging to model because of its complex fluid dynamics, particle interactions, and heat and mass flow; hence, it is a perfect candidate for an ANN [23]. ANNs are ideal for the nonlinear dynamics of the fluidized bed reactor for olefin polymerization. Polymerization reactions involve not only chemical reactions but also complex physical processes like particle–fluid interactions, catalyst behavior, and the heating up of the reactor. Traditional mechanistic models do not usually get all the details right about such systems, especially in the case of large data sets or dynamic process states. However, ANNs can also pick up patterns in the data and learn from the interaction of many variables, thus giving us correct predictions that conventional models cannot; however, this may change depending on the polymerization rate, heat output, or catalyst distribution in fluidized bed reactors. ANNs are particularly good at such unpredictable scenarios because they handle uncertainties and react quickly to new data. When using ANNs in conjunction with CFD, the entire predictive power of the simulation is boosted. The ANN learns from the CFD data and improves with each successive run, providing helpful information about reactor performance under various conditions.

3. CFD Mathematical Modeling

The model comprehensively describes the homo-polymerization of propylene over a Ziegler–Natta catalyst within a fluidized bed reactor. It incorporates key aspects of the Union Carbide polypropylene production process, a well-established industrial-scale methodology using fluidized bed technology for efficient polymerization. Additionally, it simulates the pilot-scale UNIPOL process, a widely studied and commercially relevant method for polypropylene production. It is recognized for its operational simplicity and ability to produce various polymer grades. The model also references the pilot-scale experiments, emphasizing its basis in both industrial and academic studies to ensure robustness and relevance to real-world applications (Figure 1) [24]. In developing the UNIPOL model, fundamental principles of material balance, momentum balance, and heat transfer were employed. The fluid flow patterns of the gas and solids in the reactor are assumed to be plug flows considering radial gradients and axial dispersion [22,25].

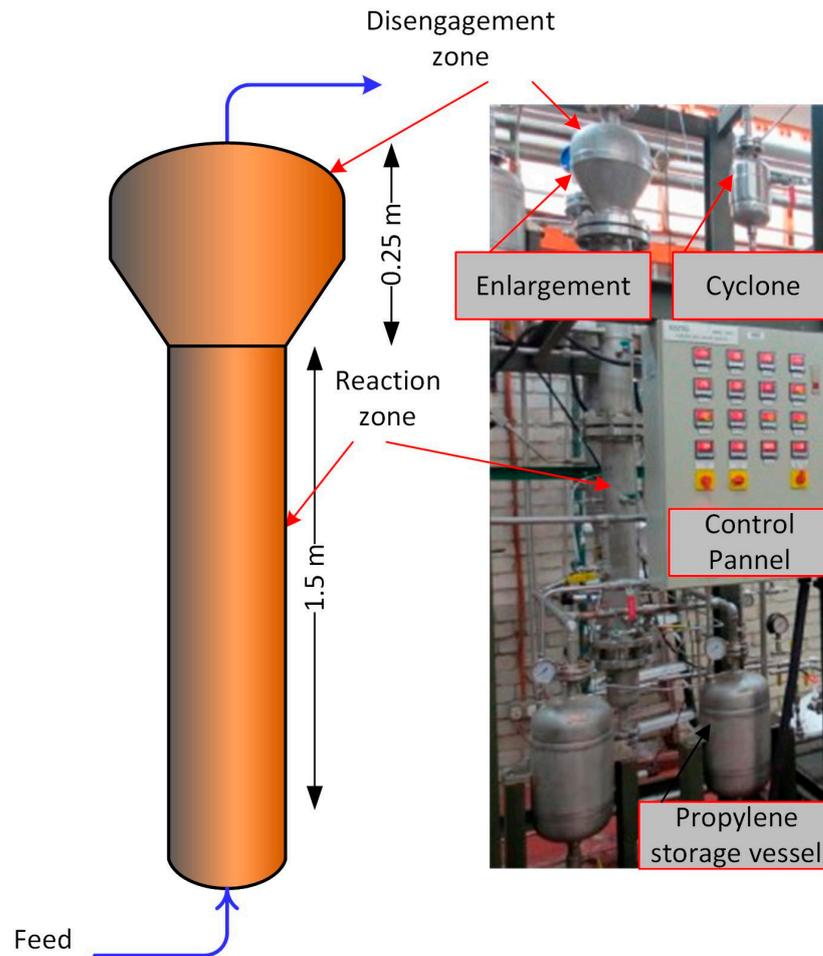


Figure 1. Pilot plant for polypropylene production in a fluidized bed reactor (UNIPOL process) [24].

3.1. Material Balance

The material balance equation describes the conservation of mass within the reactor, accounting for the inflow and outflow of monomers, catalysts, inert, and other components. The overall molar monomer balance is in cartesian coordinates, neglecting concentration change in the angular direction.

$$\frac{\partial C_i}{\partial t} = \nabla \cdot D_i \nabla C_i - u \cdot \nabla C_i + R_i \tag{1}$$

In these equations, C_i represents the concentration of the dilute species (monomer), ∇ is the del operator representing the gradient in space ($\nabla = (\partial/\partial x, \partial/\partial y, \partial/\partial z)$), D_i is the diffusion coefficient of the species, and u is the flow velocity vector. R_i represents the rate of generation or consumption of species “ i ” due to chemical reactions or other processes. The general material balance equation can be represented in a more concise and comprehensive form. The monomer concentration (C_m):

$$\frac{\partial C_m}{\partial t} = \frac{\partial}{\partial x} \left(D_m \frac{\partial C_m}{\partial x} \right) + \frac{\partial}{\partial y} \left(D_m \frac{\partial C_m}{\partial y} \right) + \frac{\partial}{\partial z} \left(D_m \frac{\partial C_m}{\partial z} \right) - u_x \frac{\partial C_m}{\partial x} - u_y \frac{\partial C_m}{\partial y} - u_z \frac{\partial C_m}{\partial z} + R_p \tag{2}$$

Initial conditions: at $t = 0$ $C_m = C_{m,0}$.

Boundary conditions: at the reactor walls of the reactor

$$\frac{\partial C_m}{\partial x} = 0$$

The propylene consumption (R_p) considering a single type of catalyst and well-mixed reactor:

$$R_p = -k_p C_m y (1 - \epsilon)$$

$$k_p = k_{p0} \exp\left(-\frac{E_a}{RT}\right)$$

where k_p is the propagation reaction rate constant, C_m is the monomer concentration, y is the catalyst concentration, k_{p0} is the pre-exponential factor, E_a is the activation energy, and ϵ is the porosity.

3.2. Fluid Flow

The Navier–Stokes equations govern the movement of fluids and can be seen as an extension of Newton’s second law of motion designed explicitly for liquids. In the case of a compressible Newtonian fluid, the equations yield the following outcomes:

$$\rho \left(\frac{\partial u}{\partial t} + u \cdot \nabla u \right) = -\nabla p + \nabla \cdot \left(\mu \left(\nabla u + (\nabla u)^T \right) - \frac{2}{3} \mu (\nabla \cdot u) I \right) + F \quad (3)$$

where u represents the velocity of the fluid, p represents the pressure of the fluid, ρ represents the density of the fluid, and μ represents the fluid’s dynamic viscosity. In a simplified form, laminar flow and constant physical properties are assumed. $(\nabla u)^T$ is the transpose of the velocity gradient tensor. The terms on the right-hand side represent the pressure gradient force, viscous forces, and the body force term (F). The velocity profile z -direction (axial direction) can be represented as follows:

$$\rho \left(\frac{\partial v_z}{\partial t} + v_x \frac{\partial v_z}{\partial x} + v_y \frac{\partial v_z}{\partial y} + v_z \frac{\partial v_z}{\partial z} \right) = -\frac{\partial p}{\partial z} + \mu \left(\frac{\partial^2 v_z}{\partial x^2} + \frac{\partial^2 v_z}{\partial y^2} + \frac{\partial^2 v_z}{\partial z^2} \right) + \rho g_z \quad (4)$$

The initial conditions ($at z = 0$): $v_z = v_0$.

The boundary condition at reactor walls (no-slip conditions): $v_x = v_y = 0$.

3.3. Heat Transfer

The following expression describes the heat diffusion governing the cartesian temperature distribution for three-dimensional unsteady heat transfer involving the heat generation term (Q_v).

$$\rho_i C_{pi} \frac{\partial T}{\partial t} = k_i \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) - \rho_i C_{pi} \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z} \right) + Q_v \quad (5)$$

Initial conditions: $at t = 0, T = T_0$.

Boundary conditions (the walls are at a fixed temperature controlled by the cooling water).

$$at x = w, T = T_w$$

where k is the conductivity of the material ($W \cdot m^{-1} \cdot K^{-1}$), ρ_i is the density ($kg \cdot m^{-3}$) of component i , C_{pi} is the specific heat capacity ($J \cdot kg^{-1} \cdot K^{-1}$), and Q_v is the rate at which energy is generated per unit volume of the medium ($W \cdot m^{-3}$). Solving the equation requires six boundary conditions, two for each coordinate. The physical properties used in the simulation are listed in Table 1.

Table 1. Physical properties and operating parameters used in the CFD simulation [22,24].

Parameter	Value	Operating Ranges
Inlet temperature (K)	333.15	320–350
Reference temperature (K)	342.15	330–360
Cooling temperature (K)	350.15	340–370

Table 1. Cont.

Parameter	Value	Operating Ranges
Reaction zone height (m)	1.5	
Reactor diameter (m)	0.1	
Pressure (atm)	22	15–30
Feed monomer concentration (mol/m ³)	0.75	0.5–1.0
Pre-exponential factor, k_{po} (m ³ /mol/s)	1.24×10^4	
Activation energy, E (J/mol)	7.04×10^4	
Catalyst active site (mol/m ³)	1.88×10^{-4}	

4. Structure of the ANN

An ANN usually has three layers: an input layer, hidden layers, and an output layer. In the present work, the ANN consists of an input layer, two hidden layers, and one output layer (Figure 2).

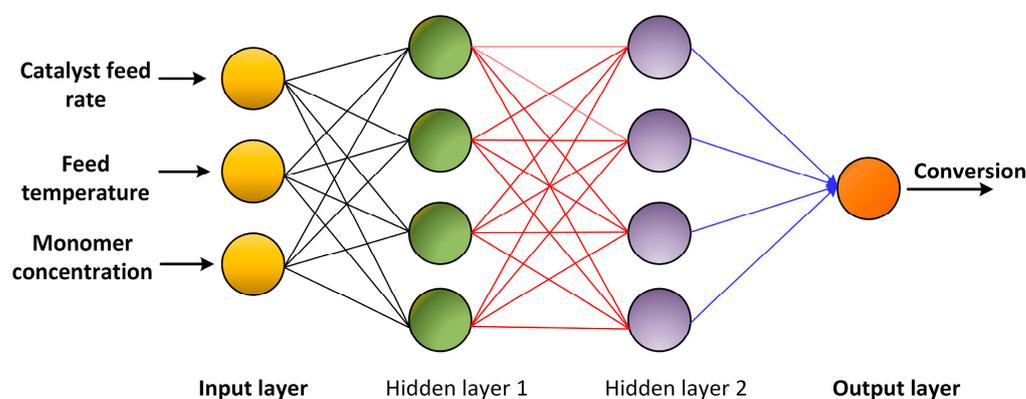


Figure 2. The architecture of an artificial neural network has three inputs, two hidden layers, and one output layer.

The input layer encodes input from outside the application, for example, process parameters (temperature, pressure, catalyst feed rates) during olefin polymerization [26]. These inputs are fed through one or more hiding layers, and neurons (units of computation) make complex mathematical transformations to extract patterns from the data. In every hidden layer, the neuron is given an activation function that introduces nonlinearity, allowing the network to simulate highly nonlinear behavior like reaction dynamics or flow in the fluidized bed. The activation function we applied in this work was Exponential Linear Unit (ELU) because it can stabilize the training cycle by decreasing the probability of gradient vanishing. The last layer, the output layer, returns predictions or classifications on the computations of the hidden layers. The result could be output with the reactor's performance data, such as product yield, reaction speed, or temperature curves. The ANN structure with neurons is connected between the input, hidden, and output layers. This is due to the network depth and the number of neurons per layer. The structure of the ANN is crucial in the context of olefin polymerization modeling as it allows for the efficient processing of complex data and the extraction of meaningful patterns, leading to accurate predictions of reactor performance [27,28].

5. Training and Optimization of the ANN Model

The ANN has four phases: model selection, training, testing, and validation. In training, the network extracts information from the data and adjusts its internal parameters (weights and biases) to make the difference between predicted and accurate outputs as small as possible. For this work, the ANN model was trained with the Adam optimization

algorithm. Adam (Adaptive Moment Estimation) is a fast, popular, and high-performing deep learning algorithm, mainly used on large and rich data like CFD simulations [29]. Adam takes the benefits of two other widely used optimization algorithms, AdaGrad and RMSProp, and it can change the learning rate in real-time for each parameter depending on the first and second moments of the gradient. This flexibility renders Adam robust and tolerant of noise and variance that are part of process modeling (e.g., olefin polymerization). It was a suitable candidate for this research because Adam can work with sparse gradients and leverage momentum to converge the optimization process faster. Unlike other optimization approaches that need a lot of fine-tuning, Adam does not need to adjust the learning rate for the individual parameters. Still, it automatically takes on its learning rate, making it easier to obtain good results across all datasets.

The Adam algorithm offers several advantages, including its adaptive learning rate, which efficiently handles sparse gradients and varying parameter scales, and its faster convergence due to momentum. It works well with default hyperparameters, making it easy to use and widely applicable across different tasks, including non-stationary objectives. However, Adam has some limitations, such as potential generalization issues compared to simpler optimizers like SGD with momentum, sensitivity to hyperparameters, and a higher risk of overfitting in models with many parameters. Additionally, it may struggle with convex problems and complex non-convex optimization landscapes, occasionally converging to suboptimal solutions. The Adam algorithm is implemented in several Python libraries, including TensorFlow, PyTorch, and Keras, where it is readily available as a built-in optimizer for training machine learning models.

6. Adam Optimization Algorithm

Adam, short for Adaptive Moment Estimation, is an efficient algorithm in machine learning for training neural networks. It is the best of two other optimization algorithms, RMSprop and Momentum. The model works as follows:

Initialize the model's parameters (weights and biases) to small random values.

Initialize the first and second moment estimates, m_t and v_t , to zeros for each parameter.

Set the learning rate, α , and the hyperparameters β_1 and β_2 (typically 0.9 and 0.999, respectively).

Calculate the gradient (g_t) of the loss function $J(\theta)$ with respect to each parameter (θ).

$$g_t = \nabla_{\theta} J(\theta_t) \quad (6)$$

In practical implementations, frameworks like TensorFlow, PyTorch, or JAX handle gradient computation using automatic differentiation.

The loss function quantifies the difference between the model's predictions and the true values. For example, for a mean squared error (MSE) loss:

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (7)$$

where y_i is the true value, \hat{y}_i is the predicted value, and n is the number of data points.

Update the first moment estimate (momentum):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (8)$$

Update the second moment estimate (RMSprop):

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (9)$$

Bias-correct the first (\hat{m}_t) and second moment (\hat{v}_t) estimates:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (10)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (11)$$

Update the parameters (θ_t):

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{v_t + \epsilon}} \hat{m}_t \quad (12)$$

where:

θ_t : the parameter vector at time step t .

α : the learning rate (10^{-3} , adjustable depending on the task).

ϵ : a small number (e.g., 10^{-8}) to prevent division by zero.

Adam is a widely used optimizer known for its ability to adjust learning rates for each parameter individually, enabling faster and more reliable convergence. By incorporating momentum, it accelerates optimization and helps overcome local minima. It is robust to noisy and sparse gradients, requires minimal hyperparameter tuning, and is effective in diverse applications. Adam is commonly applied in Natural Language Processing (NLP) tasks, such as language modeling and machine translation, and in computer vision tasks like training convolutional neural networks for image classification and object detection, where its adaptability and speed are advantageous. While Adam generally performs well, testing with different hyperparameters or alternative algorithms may be required to achieve optimal performance in some scenarios. In this analysis, Adam was chosen for its flexibility and efficiency, particularly for optimizing models with sparse gradients and complex loss landscapes. Although generalization can be a limitation, the selected hyperparameters and datasets mitigate this concern.

7. Performance Evaluation Measures

The datasets were divided into three categories, i.e., 75% of the datasets were for training, 15% for testing, and 10% for validation. The model validation is performed by utilizing the mean squared (MS) error, root mean square (RMS) error, and correlation coefficient (R). MS, RMS, and R^2 were calculated using the following equations, respectively [1,2,23].

$$MS = \frac{1}{n} \sum_i^n (X_{i,p} - X_{i,s})^2 \quad (13)$$

$$RMS = \sqrt{\frac{1}{n} \sum_i^n (X_{i,p} - X_{i,s})^2} \quad (14)$$

$$R^2 = 1 - \frac{\sum_i^n (X_{i,p} - X_{i,s})}{\sum_i^n (X_{i,p} - X_{avg})} \quad (15)$$

where n is the total number of samples, $X_{i,p}$ is the predicted ANN value, $X_{i,s}$ is the CFD simulation values, and X_{avg} is average values.

8. Results and Discussion

To initiate the study, the CFD model was rigorously validated against experimental data obtained from a pilot-scale plant, specifically designed to replicate the propylene polymerization process as detailed in reference [24]. This validation was essential to confirm the model's accuracy and predictive capability for critical process parameters, thereby ensuring its applicability for the further analysis of reactor performance. Figure 3a is the physics-controlled mesh. A normal mesh density is applied in the bulk flow region where properties are relatively uniform, optimizing computational efficiency. A denser mesh is used near the boundaries to capture steep gradients in velocity, temperature, and concentration critical for resolving boundary-layer effects. Irregularly shaped elements conform to the complex geometry of the reactor, aligning with flow directions and adapting to regions of interest like reaction zones. This mesh strategy balances accuracy and efficiency, making it well-suited for modeling the intricate dynamics of polypropylene production. In

Figure 3a, the temperature profile within the fluidized bed reactor is illustrated. The figure reveals that the reactor's wall cooling system effectively maintains a uniform temperature throughout the reaction zone, which is critical for controlling reaction kinetics and optimizing polymer yield. However, in the disengagement zone, where neither reaction nor cooling occurs, a noticeable temperature gradient emerges, highlighting the natural heat dissipation that occurs in the absence of active thermal control. Figure 3c provides insight into the velocity profile along the reactor length. The velocity reaches a fully developed profile shortly after the reactor entrance, indicating stabilized flow behavior as the gas progresses through the reactor. This characteristic is essential for achieving consistent mixing and optimal particle–fluid contact throughout the reactor bed, which are crucial factors for effective polymerization. Finally, Figure 3d depicts the monomer (propylene) concentration profile along the reactor length. As expected, the concentration of propylene decreases progressively from the reactor inlet to the outlet due to its consumption in the polymerization reaction. This decline in concentration aligns with the model's representation of the chemical kinetics involved, validating its capacity to simulate the conversion of monomers within the reactor environment accurately. Together, these profiles, temperature, velocity, and monomer concentration demonstrate the model's robustness in capturing key aspects of the fluidized bed polymerization process.

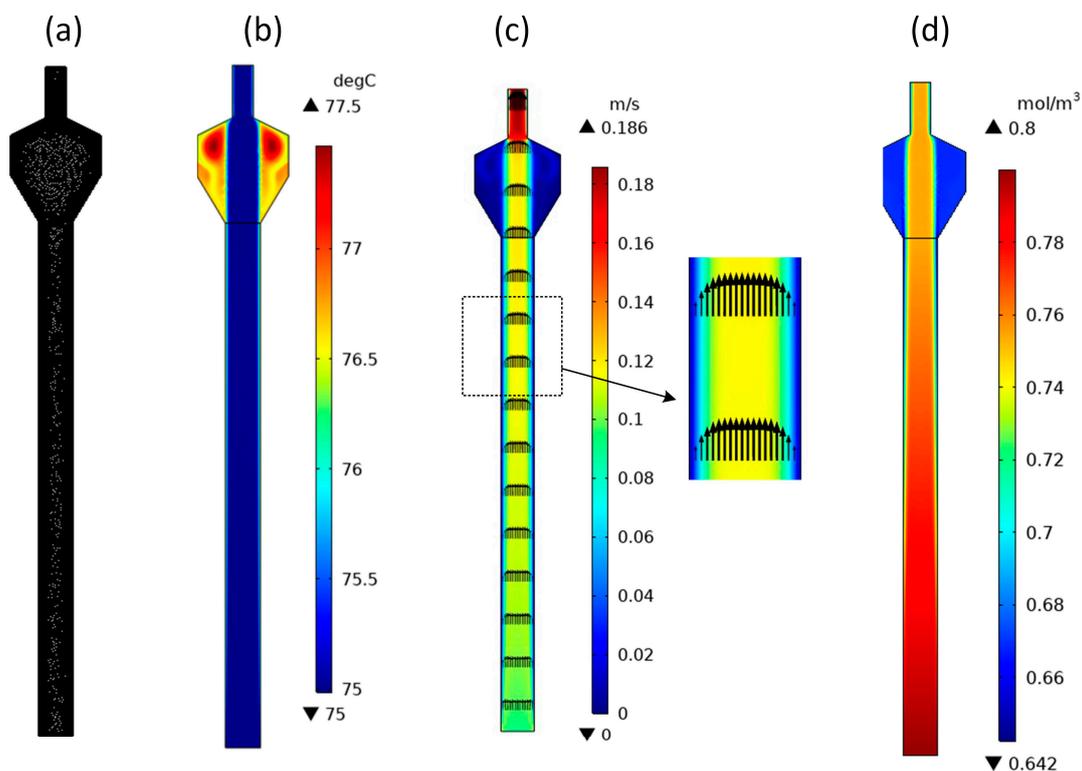


Figure 3. CFD simulation of the polypropylene pilot fluidized bed reactor, (a) physics-controlled mesh, (b) temperature profile, (c) velocity profile, and (d) monomer concentration profile.

Figure 4 compares CFD-predicted temperature profiles against the experimental data, while Figure 5 illustrates the catalyst feed rate comparisons. The close alignment between the CFD predictions and experimental measurements demonstrates the model's robustness in simulating real-world conditions within the reactor. The results indicate that increasing monomer concentration and reactor temperature substantially influences the reactor's single-pass conversion rate. Higher monomer concentrations provide more reactant molecules, promoting a greater degree of polymerization. Similarly, elevated reactor temperatures enhance the reaction kinetics, accelerating the polymerization rate. Consequently, monomer concentration and temperature contribute to an increased poly-

merization rate, leading to higher conversion efficiencies within the reactor. This insight underscores the importance of controlling these variables to optimize process efficiency and product yield in industrial-scale polymerization processes.

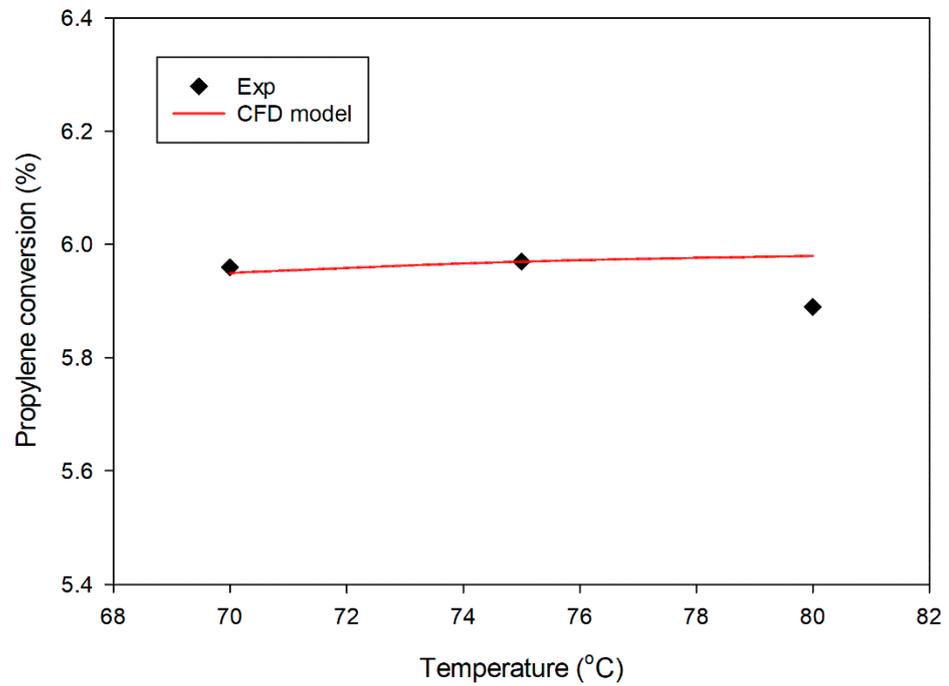


Figure 4. Comparison of pilot plant experimental data [24], with CFD model prediction for the change in reactor temperature on reactor single-pass conversion.

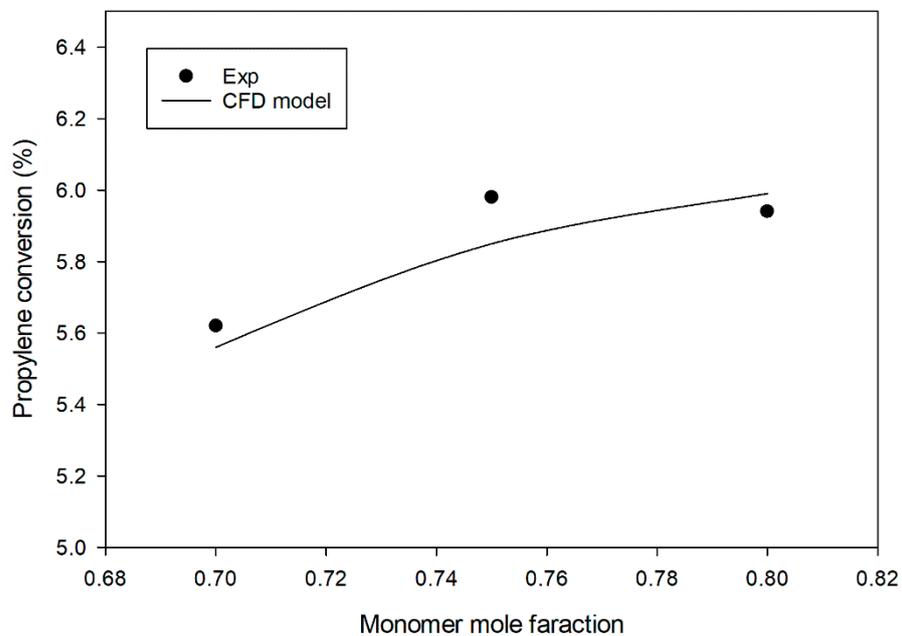


Figure 5. Comparison of experimental data from pilot plant [24] and CFD predicted results.

The ANN-generated figures illustrate the model’s effective learning and generalization capabilities across training, testing, and validation datasets. The Training and Validation Loss Over Epochs plot shows a rapid initial decrease in both training and validation losses, indicating swift learning during the early epochs. As training progresses, the losses converge to near-zero values, suggesting that the model has effectively minimized error

without overfitting. The consistent alignment of the validation loss with the training loss underscores the model's capacity to generalize well to unseen data, reinforcing its robustness (Figure 6). The ANN model is highly predictive and provides correlation coefficients ($R^2 > 0.998$) and low error ratios (MSE 0.0015) for training, testing, and validation datasets. The model contains two hidden layers (with eight neurons in each layer, activated with ReLU) and a linear output layer to make it repeatable. It was trained with the Adam optimizer at learning speed 0.001 and loss function MSE. This dataset was divided into 70% training, 15% testing, and 15% validation; data were shuffled and normalized to prevent bias. Training took 500 epochs, 32-batch size, and a 20-epoch delay to avoid overfitting. Weights were set using Xavier initialization, and random seeds were assigned to ensure consistency. These conditions were applied 10 times to the model, and it generated the same metrics over the course of each run, which showed the model's robustness and reproducibility. These details define the training and make model development transparent.

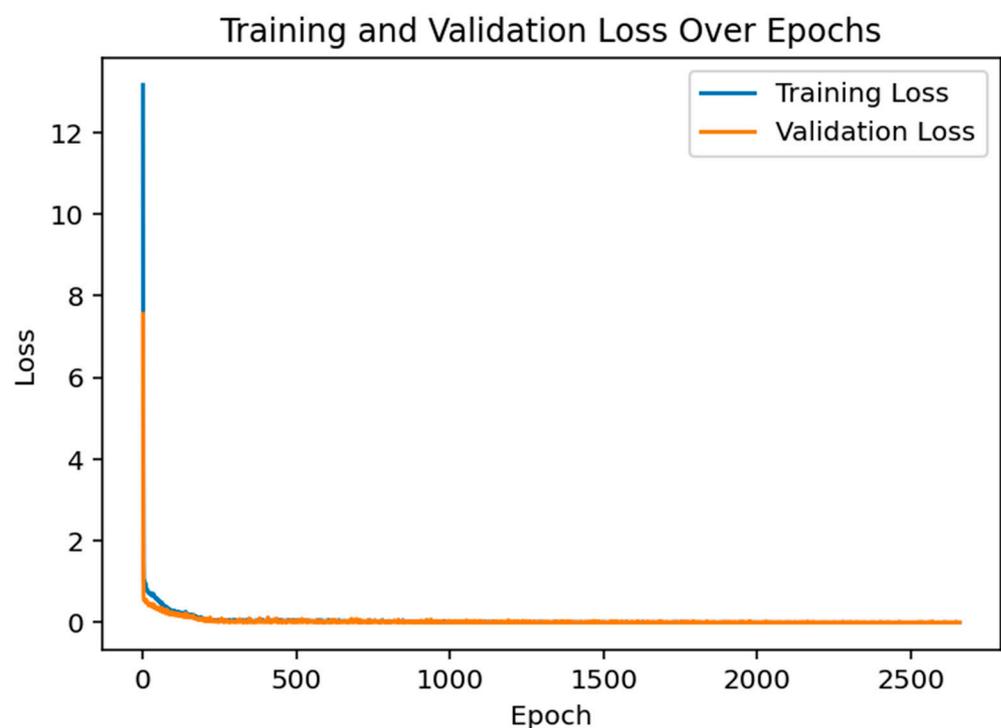


Figure 6. Training and validation loss over Epochs.

The predicted vs. actual output plots for the training (Figure 7), testing (Figure 8), and validation (Figure 8) datasets further validate the model's accuracy. In the training data plot, the predicted values align almost perfectly with the actual values along the ideal line, indicating that the model has successfully captured the underlying relationships within the training data.

In the testing (Figure 8) and validation (Figure 9) datasets, the model's predictions demonstrate a close alignment with the ideal line, signaling its robust predictive performance across diverse datasets. This tight adherence to the ideal line indicates the model's proficiency in accurately capturing underlying patterns and relationships, even when presented with new, untrained data. Such consistency suggests that the model is not only effective in training data but also adept at applying learned insights to unfamiliar datasets, a critical measure of real-world utility. The low deviation from the ideal line across both the testing and validation datasets underscores the model's strength in generalization, a key attribute for machine learning models intended for practical applications. This minimal deviation indicates that the model is resilient to overfitting, maintaining high accuracy and predictive reliability on unseen data. The ability to generalize well is essential, as it

demonstrates that the model’s performance is not confined to specific training conditions but can extend effectively to new scenarios, increasing confidence in its application to varied datasets.

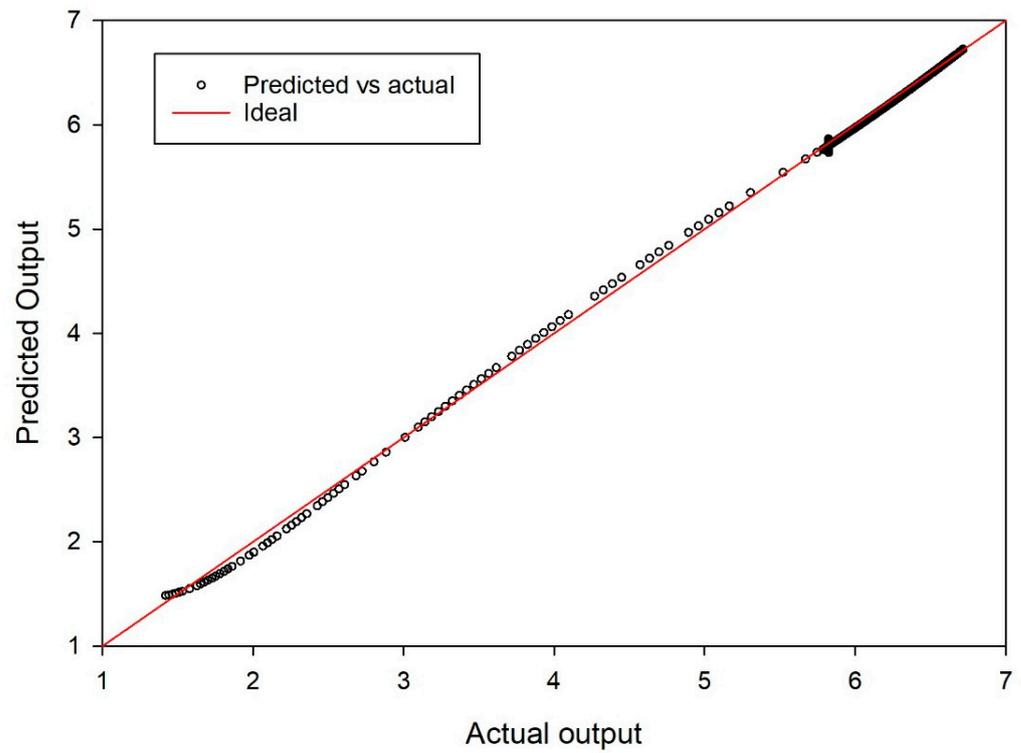


Figure 7. Training data, predicted versus actual output.

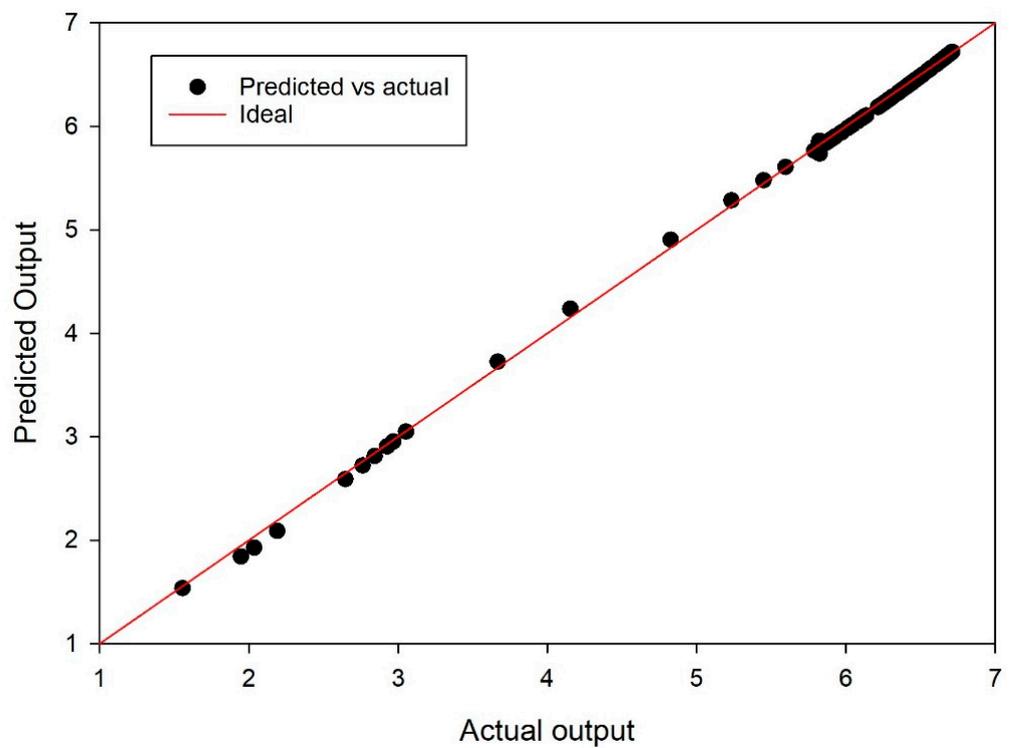


Figure 8. Testing data, predicted versus actual output.

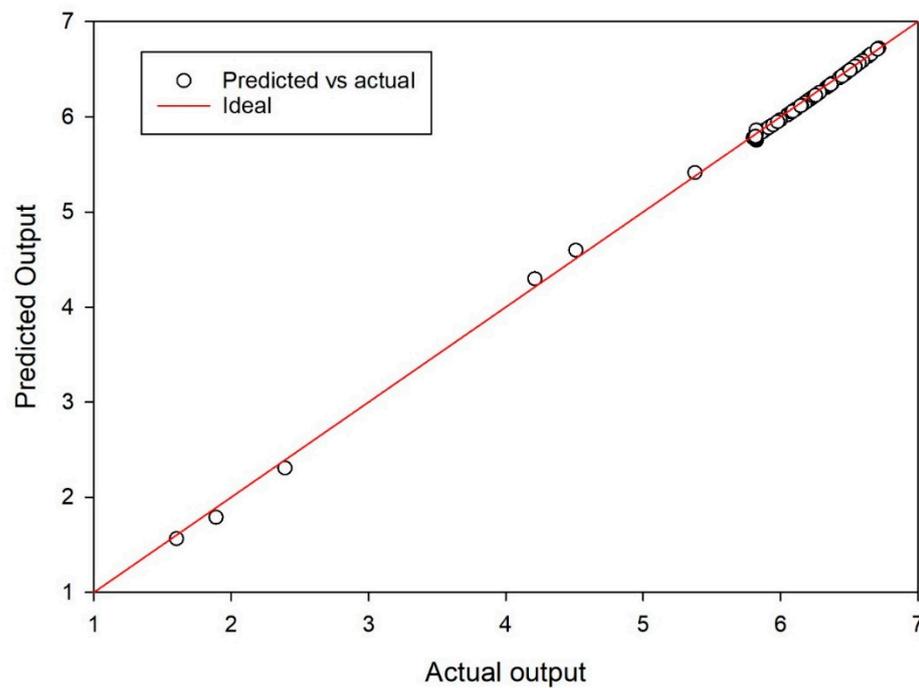


Figure 9. Validated data, predicted versus actual output.

The model performance metrics indicate a high level of accuracy across training, testing, and validation datasets, with low error values and high R^2 scores close to 1, Table 2, Table 3, and Table 4, respectively. Specifically, the Mean Squared (MS) error and Root Mean Squared (RMS) error are consistently low across all three datasets, suggesting minimal prediction error and high precision in the model’s output. The Mean Absolute (MA) error values further corroborate this, indicating that the average deviation between predicted and actual values is small. The R^2 values above 0.998 for all datasets demonstrate that the model explains nearly all the variance in the data, which is particularly promising for generalization. The close alignment of metrics across training, testing, and validation datasets also implies that the model generalizes well without significant overfitting or underfitting. This stability suggests the chosen architecture, training regimen, and data split effectively capture the underlying patterns in the data, making the model robust for prediction tasks.

Table 2. Summary of model evaluation metrics for training data.

Parameter	Value
Mean Squared (MS) error	0.0015
Root Mean Squared (RMS) error	0.03864
Mean Absolute (MA) error:	0.0314
R^2	0.999

Table 3. Performance metrics of the model on testing data.

Parameter	Value
Mean Squared (MS) error	0.001176
Root Mean Squared (RMS) error	0.0343
Mean Absolute (MA) error:	0.026
R^2	0.999

Table 4. Model performance evaluation on validation data.

Parameter	Value
Mean Squared (MS) error	0.00121
Root Mean Squared (RMS) error	0.0348
Mean Absolute (MA) error:	0.0293
R ²	0.9986

Table 5 displays the model weights. The weights provide insight into how each layer contributes to the final prediction. The weights in the first hidden layer show moderate variability across different nodes, with positive and negative values, suggesting the model captures diverse relationships between input features. This layer’s biases are relatively small in magnitude, centering the initial transformations around specific offsets. In the second hidden layer, the weights exhibit larger absolute values in some neurons, indicating that these neurons might play a stronger role in transforming the intermediate features produced by the first layer. Interestingly, the second layer has both high positive and negative values, which can help capture more complex non-linear relationships in the data. The biases in this layer further add flexibility by shifting the activation outputs. The output layer weights show that certain neurons in the final hidden layer (notably the first and second neurons) contribute heavily to the output prediction, as reflected by their larger negative weights. The bias term in the output layer is small, suggesting a minimal final offset. Overall, this weight distribution reflects a model that has adjusted its layers to handle specific patterns in the data, with nodes contributing variously to optimize the prediction accuracy.

Table 5. Optimized neural network weights and bias across layers.

Layer	Weights (Summary)	Biases
Hidden Layer 1	Weights Matrix (8 × 3): [0.1487, 0.3277, −0.4415] [0.6385, −0.1013, 1.3685] [0.0597, 0.4303, −0.2279] [1.0443, −0.0782, 0.9459] [0.1546, −0.1331, 0.5737] [−0.0257, 0.2440, 0.5846] [−0.3609, −0.0226, 0.8059] [0.2651, −0.0299, 0.6386]	[−0.5514, 0.9357, −0.1503, 0.1367, 0.2333, 0.5919, 0.8652, 1.1181]
Hidden Layer 2	Weights Matrix (8 × 8): [−0.1769, −1.0006, −0.1837, −1.2279, −1.0278, 0.3616, 0.9383, 0.1725] [−0.0056, −0.9868, 0.0468, −1.1911, −1.0450, −0.1968, 1.8472, 0.3350] [−0.2179, −0.3011, 0.1712, 0.0173, −0.1347, −0.0254, −0.2644, −1.2065] [0.0745, −1.0579, −0.2267, −1.0273, −0.6077, 0.1580, 0.7350, 0.1032] [0.1703, 0.5037, −0.0202, 0.6894, 0.7529, 0.3236, −0.0876, 1.2015] [0.1975, 0.7067, 0.3574, 0.7841, 0.4695, −0.2394, 0.3858, 1.0386] [−0.3320, −0.0073, 0.2647, −0.7450, −0.2386, −0.0717, −0.7683, −1.8494] [0.1988, 0.1581, 0.1848, 0.6047, 0.6061, 0.2058, 0.1252, 1.0464]	[0.9976, 0.8632, −0.0010, 0.5637, −0.2007, −0.1606, 0.1503, −0.4417]
Output Layer	Weights Vector (1 × 8): [−2.6298, −2.6707, −0.0223, −1.6126, 0.0857, 0.1449, −0.4213, 0.3016]	[−0.5402]

9. Conclusions

This study highlights the application of Computational Fluid Dynamics (CFD) combined with AI/ML to enhance the efficiency of polypropylene production in a fluidized bed catalytic reactor. The 2D multiphase CFD model, developed in COMSOL Multiphysics 6.2[®], accurately simulates gas–solid interactions and fluid flows, achieving excellent alignment with experimental data. Key parameters such as feed temperature, catalyst feed rate, and propylene inlet concentration were analyzed for their effects on reactor conversion and yield. Integrating CFD with AI/ML algorithms, particularly artificial neural networks (ANN), resulted in a predictive tool that enhances process understanding, optimizes reactor design, and reduces computational demands, offering valuable potential for improving industrial polypropylene production. In summary, the results collectively demonstrate that the model is well-trained and robust for predictive tasks. The low loss values and close alignment of predicted and actual values across all datasets indicate that the model has effectively captured the patterns in the data, making it highly reliable for practical applications.

Funding: This research received no external funding.

Data Availability Statement: The data supporting the findings of this study are available upon request.

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

References

1. Maddah, H.A. Polypropylene as a promising plastic: A review. *Am. J. Polym. Sci* **2016**, *6*, 1–11. [[CrossRef](#)]
2. Akbari, V.; Borhani, T.N.G.; Shamiri, A.; Shafeeyan, M.S. Computational Fluid Dynamics Modeling of Gas–Solid Fluidized Bed Reactor: Influence of Numerical and Operating Parameters. *Exp. Comput. Multiph. Flow* **2024**, *6*, 85–125. [[CrossRef](#)]
3. Gísnas, A.; Srinivasan, B.; Bonvin, D. Optimal Grade Transitions for Polyethylene Reactors. In *Process Systems Engineering 2003, 8th International Symposium on Process Systems Engineering*; Chen, B., Westerberg, A., Eds.; Elsevier: Amsterdam, The Netherlands, 2003; Volume 15, pp. 463–468. ISBN 1570-7946.
4. Soloveva, O.V.; Solovev, S.A.; Egorova, S.R.; Lamberov, A.A.; Antipin, A.V.; Shamsutdinov, E.V. CFD Modeling a Fluidized Bed Large Scale Reactor with Various Internal Elements near the Heated Particles Feeder. *Chem. Eng. Res. Des.* **2018**, *138*, 212–228. [[CrossRef](#)]
5. van Wachem, B.G.M.; Almstedt, A.E. Methods for Multiphase Computational Fluid Dynamics. *Chem. Eng. J.* **2003**, *96*, 81–98. [[CrossRef](#)]
6. Wu, K.; de Martín, L.; Mazzei, L.; Coppens, M.-O. Pattern Formation in Fluidized Beds as a Tool for Model Validation: A Two-Fluid Model Based Study. *Powder Technol.* **2016**, *295*, 35–42. [[CrossRef](#)]
7. Maleki, A.; Safdari Shadloo, M.; Rahmat, A. Application of Artificial Neural Networks for Producing an Estimation of High-Density Polyethylene. *Polymers* **2020**, *12*, 2319. [[CrossRef](#)] [[PubMed](#)]
8. Sharma, N.; Liu, Y.A. *A Hybrid Science-Guided Machine Learning Approach for Modeling and Optimizing Chemical Processes*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2021; Volume 68.
9. Sharma, N.; Liu, Y.A. A Hybrid Science-Guided Machine Learning Approach for Modeling Chemical Processes: A Review. *AIChE J.* **2022**, *68*, e17609. [[CrossRef](#)]
10. Shi, H.; Wang, S.; Wang, P. From Simulation to Reality: CFD-ML-Driven Structural Optimization and Experimental Analysis of Thermal Plasma Reactors. *J. Environ. Chem. Eng.* **2024**, *12*, 112998. [[CrossRef](#)]
11. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [[CrossRef](#)]
12. Sundaresan, S. Modeling the Hydrodynamics of Multiphase Flow Reactors: Current Status and Challenges. *AIChE J.* **2000**, *46*, 1102–1105. [[CrossRef](#)]
13. Kuipers, H.; Van Swaaij, W. *Computational Fluid Dynamics Applied to Chemical Reaction Engineering, In Scientific Computing in Chemical Engineering II*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 383–390. [[CrossRef](#)]
14. Stern, F.; Wilson, R.V.; Coleman, H.W.; Paterson, E.G. Verification and Validation of CFD Simulations Iowa Institute of Hydraulic Research (IIHR), 1999, Report No. 407, 1–58. Grace, J.R.; Taghipour, F. Verification and Validation of CFD Models and Dynamic Similarity for Fluidized Beds. *Powder Technol.* **2004**, *139*, 99–110.
15. Oberkampf, W.L.; Trucano, T.G. Verification and Validation in Computational Fluid Dynamics. *Prog. Aerosp. Sci.* **2002**, *38*, 209–272. [[CrossRef](#)]
16. Prasser, H.-M. Novel Experimental Measuring Techniques Required to Provide Data for CFD Validation. *Nucl. Eng. Des.* **2008**, *238*, 744–770. [[CrossRef](#)]

17. Aeschliman, D.P.; Oberkampf, W.L. Experimental Methodology for Computational Fluid Dynamics Code Validation. *AIAA J.* **1998**, *36*, 733–741. [[CrossRef](#)]
18. Boye, T.E.; Nwaoha, T.C.; Olusegun, S.D.; Ashiedu, F.I. A Validation Method of Computational Fluid Dynamics (CFD) Simulation against Experimental Data of Transient Flow in Pipes System. *Am. J. Eng. Res.* **2017**, *6*, 67–79.
19. Whyte, H.E.; Raillard, C.; Subrenat, A.; Héquet, V. Influence of Operating Parameters on the Single-Pass Photocatalytic Removal Efficiency of Acrylonitrile. *J. Photochem. Photobiol. A Chem.* **2019**, *382*, 111905. [[CrossRef](#)]
20. Sirat, A.; Ahmad, S.; Ahmad, I.; Ahmed, N.; Ahsan, M. Integrative CFD and AI/ML-Based Modeling for Enhanced Alkaline Water Electrolysis Cell Performance for Hydrogen Production. *Int. J. Hydrogen Energy* **2024**, *83*, 1120–1131. [[CrossRef](#)]
21. Shamiri, A.; Hussain, M.A.; Mjalli, F.S.; Shafeeyan, M.S.; Mostoufi, N. Experimental and Modeling Analysis of Propylene Polymerization in a Pilot-Scale Fluidized Bed Reactor. *Ind. Eng. Chem. Res.* **2014**, *53*, 8694–8705. [[CrossRef](#)]
22. Lira, J.O.B.; Riella, H.G.; Padoin, N.; Soares, C. Computational Fluid Dynamics (CFD), Artificial Neural Network (ANN) and Genetic Algorithm (GA) as a Hybrid Method for the Analysis and Optimization of Micro-Photocatalytic Reactors: NO_x Abatement as a Case Study. *Chem. Eng. J.* **2022**, *431*, 133771. [[CrossRef](#)]
23. Khan, M.J.H.; Hussain, M.A.; Mujtaba, I.M. Developed Hybrid Model for Propylene Polymerisation at Optimum Reaction Conditions. *Polymers* **2016**, *8*, 47. [[CrossRef](#)] [[PubMed](#)]
24. Ghasem, N. Modeling and Simulation of a Multizone Circulating Reactor for Polyethylene Production with Internal Cooling. *Polymers* **2023**, *15*, 3741. [[CrossRef](#)] [[PubMed](#)]
25. Ghasem, N.M. Design of a Fuzzy Logic Controller for Regulating the Temperature in Industrial Polyethylene Fluidized Bed Reactor. *Chem. Eng. Res. Des.* **2006**, *84*, 97–106. [[CrossRef](#)]
26. Hammoudi, M.; Henao, C.; Agbossou, K.; Dubé, Y.; Doumbia, M.L. New Multi-Physics Approach for Modelling and Design of Alkaline Electrolyzers. *Int. J. Hydrogen Energy* **2012**, *37*, 13895–13913. [[CrossRef](#)]
27. Bilgiç, G.; Bendeş, E.; Öztürk, B.; Atasever, S. Recent Advances in Artificial Neural Network Research for Modeling Hydrogen Production Processes. *Int. J. Hydrogen Energy* **2023**, *48*, 18947–18977. [[CrossRef](#)]
28. Sareen, K.; Panigrahi, B.K.; Shikhola, T.; Nagdeve, R. Deep Learning Solar Forecasting for Green Hydrogen Production in India: A Case Study. *Int. J. Hydrogen Energy* **2024**, *50*, 334–351. [[CrossRef](#)]
29. Shao, Y.; Wang, J.; Sun, H.; Yu, H.; Xing, L.; Zhao, Q.; Zhang, L. An Improved BGE-Adam Optimization Algorithm Based on Entropy Weighting and Adaptive Gradient Strategy. *Symmetry* **2024**, *16*, 623. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.