Recipe Based Anomaly Detection with Adaptable Learning: Implications on Sustainable Smart Manufacturing

Junhee Lee , Jaeseok Jang , Qing Tang , Hail Jung *

Posted Date: 2 January 2025

doi: 10.20944/preprints202411.1209.v4

Keywords: Industry 4.0; artificial intelligence; injection molding; data-driven AI; anomaly detection; Recipe-Based Learning; Adaptable Learning; sustainable manufacturing

Article

Recipe Based Anomaly Detection with Adaptable Learning: Implications on Sustainable Smart Manufacturing

Junhee Lee [1,‡], Jaeseok Jang [1,‡], Qing Tang [1,] and Hail Jung [2,*]

[1] Data Science Group, INTERX, Ulsan 44542, Republic of Korea; Junhee.Lee@interxlab.com (J.L.); Jay.jang@interxlab.com (J.J.), tangqing@interxlab.com (Q.T.)
[2] Department of Business Administration, Seoul National University of Science and Technology, Seoul 01811, Republic of Korea
* Correspondence: hail95@seoultech.ac.kr; Tel.: +82-10-2994-7527
‡ These authors contributed equally to this work.

Abstract: The advent of Industry 4.0 has significantly transformed the manufacturing sector, bringing advancements in quality control efficiency, environmental sustainability, and production development. These changes have led to development of intelligent technologies such as artificial intelligence(AI). However, implementing AI solutions in manufacturing processes still faces challenges in many aspects, particularly in handling irregular datasets influenced by diverse manufacturing settings. Especially in the field of injection molding, quality inspection often occurs at the batch level rather than per individual data, providing only the overall defective ratio of the batch production instead of each labeling result. These issues limit the general application of AI and data-driven decision making. To address these limitations and promote sustainable manufacturing, this study proposes a novel anomaly detection framework for a specific manufacturing process. For Recipe-Based Learning, we first apply K-Means to separate injection molding data into specific-setting based recipes to secure data normality. Then, autoencoders for anomaly detection are trained with normal data from each recipe. With taking control of the nature of the dataset, this data-driven AI approach resulted in predicting 61 defective products compared to the existing 41 defects. Meanwhile, the integrated model without considering the change of settings only predicted 2 defects which imply a poor and distorted quality inspection. For Adaptable Learning focused on new inputs with unseen settings, we apply KL-Divergence to find the closest trained recipe data and its model. In this case, it also exceeded the predictivity compared to the integrated and additionally trained models. This leads to continuous prediction without further training, which successfully expands the data-driven AI approach considering the change of settings. In the aspect of smart factory in the injection molding field, such improvement of continuous quality inspection with the proper application of AI can first enhance the productivity throughout the advancement of data-driven decision making. Furthermore, it is expected to efficiently reduce such environmental waste and computational cost, ensuring the overall sustainable manufacturing.

Keywords: Industry 4.0; artificial intelligence; injection molding; data-driven AI; anomaly detection; Recipe-Based Learning; Adaptable Learning; sustainable manufacturing

1. Introduction

1.1. Background

Industrialization plays a critical role in solving key challenges for maintaining viability by introducing cutting-edge technologies in smart manufacturing. Recently, many industries are accelerating the application of Industry 4.0 to improve quality control and customization efficiency, and expect benefits such as increased production speed, reduced manufacturing costs, and reduced environmental impact. However, many difficulties remain in effectively adopting intelligent technologies due

 doi:10.20944/preprints202411.1209.v4

to financial, regulatory, and organizational management issues. Nevertheless, the future impact of Industry 4.0 on the manufacturing ecosystem is inevitable, leading to active research and experiments on the possibility of manufacturing innovation[1,2].

In particular, integration of new technologies such as artificial intelligence (AI) in Industry 4.0 has risen as a key element in the innovation of manufacturing process. The main objective of applying AI is to improve the productivity and flexibility of the overall process. The techniques consisting of machine learning, deep learning, and reinforcement learning help to advance data-driven decision-making [3]. Since each manufacturing process has different characteristics of product development, it becomes very important to explore the characteristics of the data in a concrete way and apply the optimal algorithm to enhance management. Furthermore, results of related research also show that the development of AI in the manufacturing sector can solve environmental problems[4,5]. In other words, if manufacturing-AI is continuously developed, it is expected to give contribution to society efficiently in various ways.

Many studies have shown that the application of manufacturing-AI can lead to the advancement of the industry and this is a task that should to be continuously developed in the future[6–8]. However, despite the technological advancement of hardware and software, the usage of AI solutions in the manufacturing sector confront many limitations such as scalability, and it is known that much research is yet to be done [9].

1.2. Issue

In the field of injection molding, the datasets used in research often do not sufficiently reflect the realistic characteristics of the entire product development process. These datasets generally have a balanced ratio of results for normal and defect qualities per each product in order to mainly focus on improving predictive measures of the applied algorithms. So it can be said that labeling the status of data on a 1:1 product basis is the key element in applying AI[10]. But in actual industrial sites, there is a significant lack of defective data, making it difficult to build an AI model which properly takes this into account[11]. Especially in the injection molding process, it is considerably difficult to perform real-time quality inspection throughout the process of every product. Rather, products in this circumstance are generally collected in batches, leading to quality evaluation which can only be done by the count of defective products per each batch. This process makes it difficult to determine whether each data indicate a normal or defect quality. In a worse condition where there is a lack of defective data itself, application of AI to measure quality inspection becomes even more limited.

In addition, there are circumstances in the field of injection molding that the setting values during product development process are frequently changed. This makes training the data more complicated[12]. If the setting values during manufacturing process change, the data itself loses its normality, and the more frequent the settings change, the more complex the data becomes. Even for a single product, the data might consist of mixture distributions which have different characteristics per each manufacturing setting. If analysis is performed without considering the diverse settings, there is a high possibility of causing distorted results. This issue also makes it difficult to apply AI and perform data-driven decision making.

1.3. Our Idea

This paper proposes a new framework for applying artificial intelligence in a data-driven approach specialized in the injection molding process. By utilizing the process characteristics with few defects, we aim to build an Anomaly Detection model that learns only data of good products in a batch process and detect actual defects throughout the remaining data. Also, in order to reflect the frequent changes of setting values during product development, we introduce the 'Recipe-based learning' approach. A 'Recipe' is defined as a unique set of setting values that is manipulated throughout the process. If even one value is changed from a set of multiple settings, it is considered as a new recipe. With this concept, Recipe-based learning means a method of classifying the dataset by each recipe and training its model. Here, we aim to present an approach that can optimize the setting values and anomaly detection

 doi:10.20944/preprints202411.1209.v4

models of the injection molding process. In other words, this approach first helps to effectively control the whole data depending on diverse change of settings, which develops a cornerstone in data-driven decision making when it comes to quality inspection. With further extension, we introduce the 'Adaptable-learning' approach. Here, we aim to directly predict new data with unseen settings by using trained models of the closest recipe. With the extensive framework of 'Recipe-Based' and 'Adaptable' learning, this paper emphasizes the capability of taking control of the data and applying AI in a specific manufacturing process. Furthermore, in the aspect of smart factory [13], achievement of sustainable manufacturing in the field of injection molding is to be expected. For the experiment, we used an injection molding dataset with labeling and setting issues collected from a private company in South Korea.

1.4. Contributions

1. Optimization in Defect Detection: This paper advances the field of injection molding by focusing on data-driven AI modeling to enhance the accuracy and efficiency of defect detection, aiming to optimize manufacturing quality and process reliability.
2. Reduced training time: Since additional learning for new data becomes unnecessary, the required learning time is reduced and the maintenance period of the product life cycle can be extended.
3. Securing corporate competitiveness: By reducing the cost of quality inspection due to advanced defect detection, the company can secure its own quality competitiveness.
4. Reduction of resource waste: When modeling is performed for each individual setting, optimizing model parameters during training process becomes more simple due to distinguishable data characteristics. This reduces the cost of inference. This is also held with additionally collected data which reduces cost of retraining models.

The paper is structured to provide a clear progression from foundational concepts through to experimental validation and future implications. Section 2 provides context for our approach by reviewing the evolution of injection molding technology and recent AI advancement in this area, emphasizing how our method builds upon and improves existing techniques. Section 3 details the algorithms and methods we use to enhance defect detection accuracy, demonstrating how our approach impacts production efficiency. In Section 4, application of the experimental data showcases our method's real-world effectiveness, significantly outperforming traditional approaches in defect detection accuracy. The Discussion section interprets these results, highlighting the potential for data-driven AI based defect detection to improve sustainability and resource efficiency in manufacturing. Finally, the Conclusion advocates for the targeted use of AI in manufacturing automation and proposes future research paths that align with sustainable and resource-efficient practices.

2. Literature Review

The injection molding industry has made great strides over time in leveraging AI.

2.1. Initial Experimentation and Predictive Modeling (Early 2000s)

Early AI implementations in injection molding aimed at predictive maintenance and process optimization, often using basic machine learning algorithms to forecast equipment failures and adjust parameters based on historical data. However, these systems struggled with limitations of data, as molding processes are highly complex and sensitive to subtle fluctuations in conditions. This made prediction accuracy inconsistent and dependent on operator expertise to validate predictions, limiting scalability in complex production environments[14].

2.2. Integration with Industry 4.0 and Real-Time Monitoring (Mid-2010s)

With the onset of Industry 4.0, AI's role in injection molding expanded to include real-time data monitoring and adaptive process controls. By using IoT sensors and more advanced machine learning models, manufacturers began to adjust molding parameters on the fly, significantly enhancing

 doi:10.20944/preprints202411.1209.v4

efficiency and reducing defect rates. Yet, challenges remained, particularly with the compatibility of data across different machine systems and the difficulty in achieving "zero-defect" manufacturing due to high variability in material and environmental factors[15].

2.3. Advanced Quality Prediction and Zero-Defect Ambitions (2020s)

More recent advancements focus on achieving high-quality and zero-defect production through complex AI models like neural networks and human-in-the-loop systems. These allow AI to refine parameters in real-time, reducing waste and maintaining consistency. However, these systems demand substantial computational resources and skilled data scientists to manage and interpret results, creating a high entry barrier for smaller manufacturers. In addition, the effectiveness of these systems is often limited by variability in data quality and completeness, hindering universal application[10].

Over time, the importance of AI in the injection molding field is increasing day by day, and its accuracy is also improving. However, due to frequent changes in the set values, it is very difficult to utilize AI[12]. In addition, deploying AI for defect detection and quality prediction in the injection molding industry has difficulty in learning effective models due to the very small number of defect data. This lack of defect samples limits the ability of AI models to generalize and accurately predict rare defects[16].

2.4. Similar Studies

In fact, in the injection process with the characteristics of a batch-unit, Autoencoder has been utilized for monitoring, quality prediction, etc[12,17]. However, it is said that many difficulties still exist in practical application, citing limitations such as a decrease in accuracy due to changes in new recipes (setting values)[12].

2.5. This Work

As explained above, due to the nature of the injection molding process, there is no actual quality labeling result per individual data and only the defect ratio in a simple cavity unit production can be known. So as a data-driven AI approach, we used the existing Autoencoder method that learns the distribution of good products and detects defects by utilizing only the data of all good products in the cavity or batch production unit[12,17].

In this paper, we propose a method to secure data normality for more sophisticated modeling and a methodology to properly predict new types of data. First, we propose a method to learn by separating the settings. If the setting values are classified, the regularity of the training data set can be secured as an example in Figure 1. If modeling is performed based on this regularity, higher performance of quality inspection can be maintained. As such, applying AI via taking full control of the nature of the dataset becomes possible.

In addition, to efficiently verify when a new type of data is input, we propose a method of making predictions for new data by utilizing a training model most similar to the new data from the existing Recipe training unit model by using an algorithm that identifies the distribution of the new data.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 January 2025 doi:10.20944/preprints202411.1209.v4

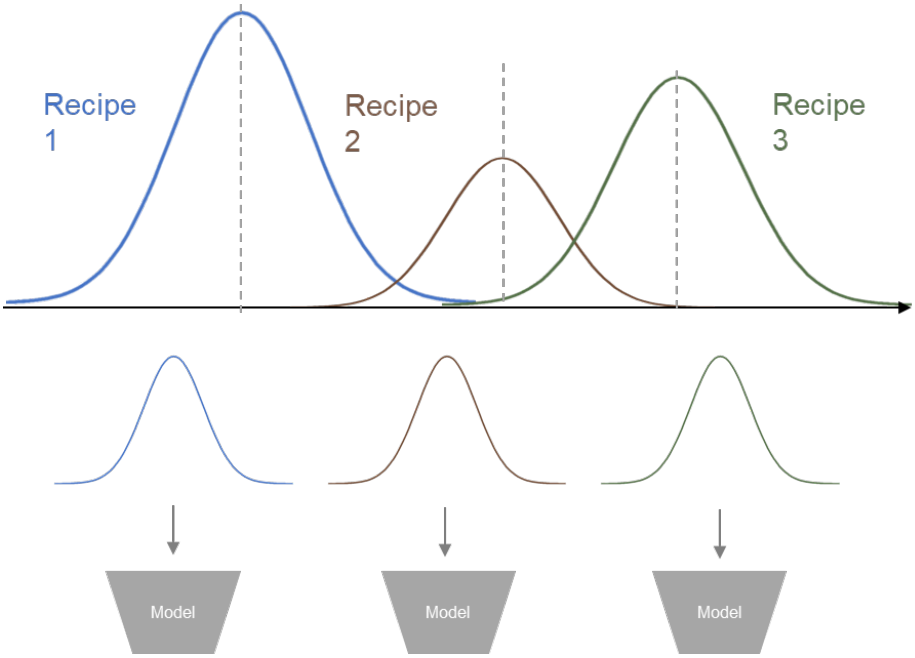

Figure 1. Distribution of Injection Molding data by Settings

3. Methodology

In this section, we introduce our proposed architecture for implementing anomaly detection focusing on data with a lack of individual labeling results consisting of setting values. The general architecture is described in Figure 2. For a brief introduction, the overall description of the collected dataset for experimental analysis is first explained. Second, the K-Means clustering algorithm [18] is described focused on securing data normality with setting-based separation. Third, the train and test data are defined for anomaly detection. Only normal data are trained, and the rest are used for prediction. Data regularization with min-max scaling is further done before the modeling process. Fourth, predictive modeling is based on the Autoencoder[19] model. Results beyond the optimized threshold of reconstruction errors are treated as anomalies. Finally, for adaptable learning, the KL-Divergence[20] is used to find the closest data among trained recipes, compared to new data based on unseen settings. Then, with the trained AutoEncoder based on the closest trained recipe, new data becomes directly predictable.

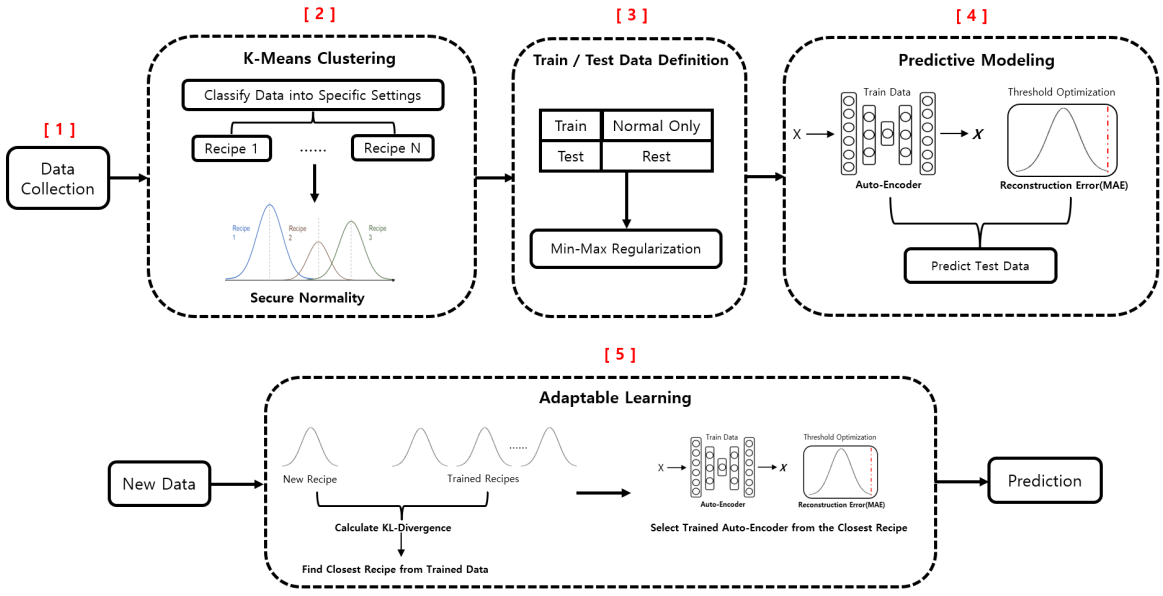

Figure 2. General Architecture of The Proposed Method

 doi:10.20944/preprints202411.1209.v4

3.1. Data Collection

The injection molding dataset for the experiment is based on a procedure measured in a batch described in Figure 3. First, a product consists of a single or multi- cavity mold. For each process, a single-cavity mold yields one product, and a multi-cavity mold yields sub-products. For example, if a 4 cavity mold product consists of cavities which 3 are normal and 1 is defect, the defect ratio of this product is 25 percent. Now, these cavity-based products are assigned to a specific batch. In such process, quality inspection is usually not performed for each product. Instead, the overall defect ratio is known based on the count of good and bad cavities. Since individual labeling is absent, it is difficult to directly apply AI methods of classification or anomaly detection. Furthermore, if settings are changed multiple times throughout the process, even a dataset with a single item-code consists of multiple distributions. In this case, data preprocessing based on a domain-specific approach is required.

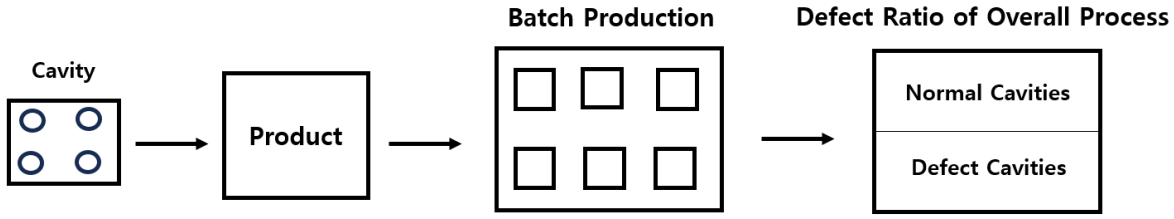

Figure 3. Process Management Based on Batch Production

The description of the experiment dataset is as follows.

1. Collected data size is $\boldsymbol{D}$ with $\boldsymbol{N}$ features.
2. The dataset consists of $\boldsymbol{F}$ facilities and $\boldsymbol{P}$ products.
3. Each product consists of $\boldsymbol{C}$ cavities.
4. $\boldsymbol{S}$ setting features are used for classifying data into recipes.
5. $\boldsymbol{I}$ input features are used for modeling Autoencoders.

The rest of the features include information such as facility and product code, process date, etc. And the target feature for prediction is the overall defect ratio of good and bad cavities in the batch process.

3.2. Recipe Separation

The prior objective of our work is comparing the prediction results of trained models based on setting-based classified data which secures normality and the original data. Considering the change of distributions throughout different setting values, our work aims to improve the prediction accuracy of defect cavities with data-driven AI approach.

In order to classify data into recipes based on specific settings, the K-Means clustering algorithm [18] is applied to the original product data.

The K-Means Clustering is an unsupervised algorithm for grouping data of similar characteristics based on distance measure. The process is as follows.

- First, the number of clusters K is decided for how many clusters the data should be divided into.
- Initial centroids for each cluster are given. Data close to each centroid are assigned to each cluster with distance-based metrics such as Euclidean-Distance.
- Initial centroids move to the center of the data points of each assigned cluster. This process is repeated until every data are assigned to clusters and centroids are finally adjusted.

An example of the K-Means result is shown in Figure 4.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 January 2025 doi:10.20944/preprints202411.1209.v4

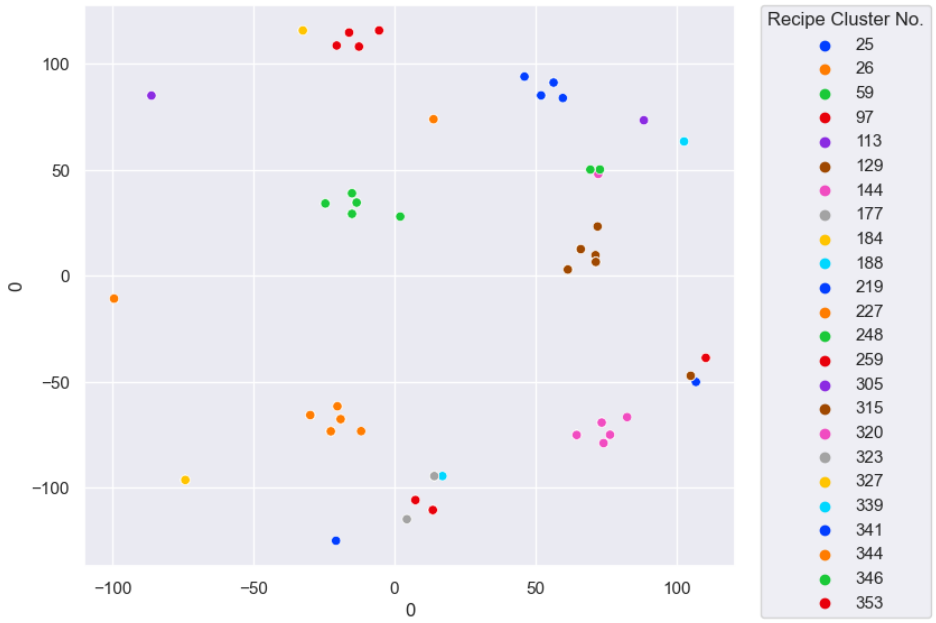

Figure 4. K-Means Clustering Example

Next, The process of applying K-Means is as follows.

- Among dataset size of $\boldsymbol{D}$, select data with only $\boldsymbol{N}$ setting features.
- From the selected data, find $\boldsymbol{C}$ unique combinations of setting features.
- Set the parameter *n clusters* as $\boldsymbol{K}$ which is the same number of $\boldsymbol{C}$.
- Apply Standard-Scaling method to unify different measures of setting features.
- Train K-Means Clustering with final data size of $\boldsymbol{d}$ rows and $\boldsymbol{N}$ features
- Return to the original data size $\boldsymbol{D}$ and predict each data based on the setting features.
- Throughout prediction, designate the cluster number from 1 to $\boldsymbol{K}$ for each data.
- Define the final numbers of each cluster as setting parameter-based recipes.
- When new data is collected, the cluster(recipe) number of setting features are predicted with the trained K-Means.

The outline of K-Means application is shown in Figure 5.

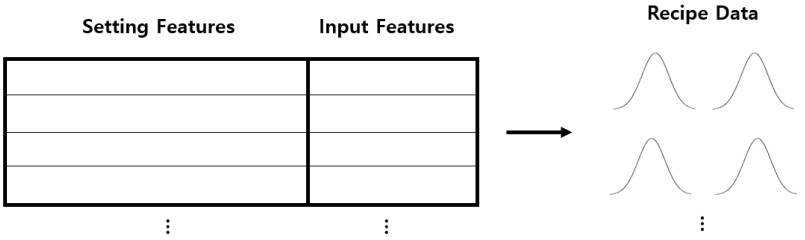

Figure 5. Classify Data into Recipe Settings

3.3. Train/Test Data Organization

As mentioned in 3.1, the data from such process lack individual labeling results. Rather, only the overall defect ratios comprising the count of good and bad cavities are known. For anomaly detection, batch process which have no defects(normal quality) can first be defined as the train dataset. To prevent overfitting issues of prediction models, 90% are actually used for training and 10% for validation. Finally, the normal data itself(integrated data) and the classified recipes are then used for training models.

 doi:10.20944/preprints202411.1209.v4

Then, batch process with existing defects (defect ratio > 0.0%) are defined as the test dataset. For appropriate prediction, additional preprocessing is done in two situations.

1. Suppose a batch process consists of multiple setting-based recipes which also exist in the train data. Although it seems appropriate for prediction, there is a setback. Since only the overall defect ratio per batch process is known, it is inappropriate to compare predicted defect ratios of multiple recipes and the existing ratio. In other words, in a batch process, the defect ratio per each recipe is untrackable. Accordingly, batch process which have one unique setting recipe are left in the test data.
2. Suppose the test data consists of an unseen recipe information. In this case, only partial prediction can be done leading to incomplete quality measurement of the batch process.

Finally, the preprocessed test data itself(integrated data) and the classified recipes are then used for prediction. By defining the integrated and recipe-based datasets, a comparison of the predictive analysis can be performed considering the settings.

3.4. Min-Max Regularization

Before building prediction models, a Min-Max scaling method is applied that transforms the input range from 0 to 1 to the input features[21]. In deep-learning methods, regularization with scaling is vital for properly training data. Especially, it can mitigate input bias[22], gradient vanishing or exploding[23], covariance shift[24], and exponential convergence of training loss [25]. Scaling is first done based on the train dataset. Then, features of validation and test data are regularized based on the train dataset.

3.5. Predictive Modeling

3.5.1. Autoencoder Application

To perform anomaly detection with train and test data defined in 3.3 , the Autoencoder method is applied as the main prediction model in our work.

The main function of Autoencoder is reconstructing the inputs into similar outputs throughout a symmetric architecture. [19,30]

- Inputs are compressed through the encoder layer into the latent vector.
- In the latent vector, non-linear correlations between features are captured which can effectively learn important components of inputs.
- Throughout the latent vector, reconstructed outputs are predicted with the decoder layer.
- Inputs for training Autoencoders of the recipes and the entire normal data(integrated data) are the $\boldsymbol{I}$ input features.

Since the main objective of our work is applying AI in such manufacturing process which have barely been used for analysis in other related works, it is closer to a data-driven approach [26]. This is why Autoencoder of a fundamental structure like Figure 6 is used rather than applying Autoencoder variants[27] which focus on improving model performance which is more of a model-driven approach [28,29].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 January 2025 doi:10.20944/preprints202411.1209.v4

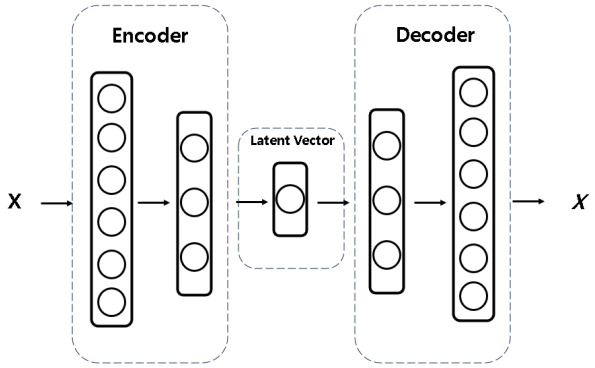

Figure 6. Autoencoder Structure

3.5.2. Threshold Optimization with Evaluation Metrics

After training the Autoencoder models, the loss value of reconstruction errors between the inputs and predicted outputs is calculated using mean absolute error. Among several metrics, The Mean Absolute Error (MAE) is selected to see a more intuitive difference between the inputs and outputs. The MAE function calculates the mean value of absolute errors between the given inputs and the predicted output values [31]. From the equation , where n is the train data size, Y_i is the input , and $\mathbf{Y}_i$ is the reconstructed output [32] .

$$MAE = \frac{1}{n}\sum|Y_i - \mathbf{Y}_i| \tag{1}$$

Furthermore, thresholds for anomaly detection are based on the maximum MAE value of the reconstruction error of the validation normal data. If the MAE value of an input exceeds the threshold, it is predicted to be anomalous. The application is to be performed in Section 4.5.2.

3.6. Adaptable Learning

When new data of untrained recipe settings are collected, additional training is usually required. However, if it is available to define the model of trained recipe with the closest distribution compared to the new settings, direct prediction becomes available. This indicates the extension of controlling the data beyond the current timestamp. In our work, this is defined as adaptable learning.

First, KL-Divergence is applied for calculating the difference of distributions among setting values. The Kullback-Leibler (KL) divergence is a statistical measure which quantifies the distance between two probability distributions. From the equation below, [20], the definition of distribution P and Q in our work is explained as follows.

- P is defined as each setting feature of the trained recipe.
- Q is defined as each setting feature of the new recipe.
- The range of KL-Divergence is from zero to infinity. If the divergence is lower, the distributions are interpreted to be more similar.
- As a result, the goal is to make sure to find Q which can properly infer P.

$$D_{KL}(P||Q) = \int P(x) log \frac{P(x)}{Q(x)} dx \tag{2}$$

Next, The process of applying KL-Divergence is as follows.

- Select recipes ***R,...., R+K*** of the trained dataset.
- Define recipes ***r,...., r+k*** of the new dataset. (Recipe numbers are predicted by the trained K-Means model in 3.2.)

 doi:10.20944/preprints202411.1209.v4

- Select N setting features per each trained recipe and new recipe.
- Calculate the KL-Divergence value of each setting feature between the new and trained recipes.
- Calculate the sum of the calculated KL-Divergence values.
- Select the trained recipe with the lowest KL-Divergence value compared to the new recipe.
- Select model of trained recipe and predict data of new recipe.
- Define I input features used for prediction.

The outline of KL-Divergence is shown in Figure 7.

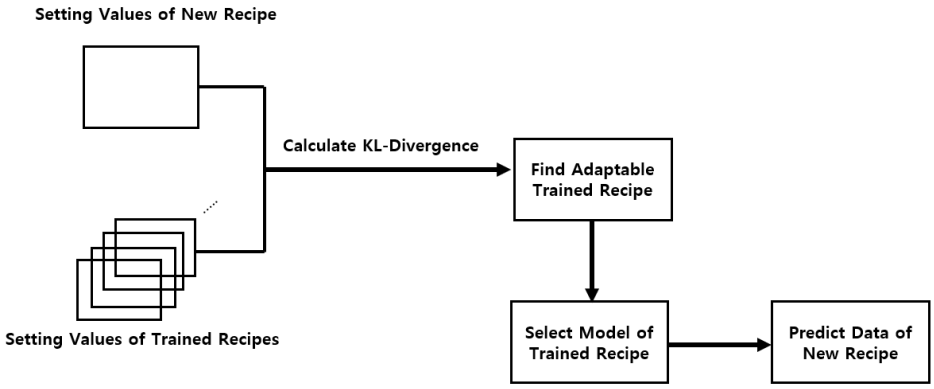

Figure 7. KL-Divergence Application

4. Experiment Setup

4.1. Data Collection

The injection molding dataset used in our work is collected from a private manufacturing company in South Korea. Referred to Section 3.1, inputs are defined as D = 432089 , N = 132, F = 4, P = 14, C = 1, S = 76, I = 6. For further information, data with the size of 432098 with 132 features consist of 14 products manufactured in 4 facilities. Every product is based on a single-cavity. 76 setting features are used for numbering recipes and 6 input features are finally used for anomaly detection.

4.2. Recipe Separation

The dataset is first classified via unique setting parameters. Then, inputs following the procedure of 3.2 are defined as D=432089 , N=76, C=366, K =366, d=366.

An example of understanding the dataset is described in Table 1. First, batch number 20231214 is the date when the manufacturing process was held. The batch process with specific settings defined as recipe number 10 results in approximately 1.37% defect ratio of cavity-based products.

Table 1. Information of Batch process Dataset

Data Information of a Specific Recipe	
Batch Number	20231214
Data Shape	438 Data Size and 132 Features
Setting Features	76 Setting Features
Input Features	6 Input Features
Number of Products	438
Predicted Recipe(Cluster) Number	10
Good Cavity Counts(Total)	432
Defect Cavity Counts(Total)	6
Overall Defect Ratio	Approximately 1.37 %

4.3. Select Experiment Data

From the complete dataset, we chose the most appropriate subset to perform our proposed method based on the following order.

 doi:10.20944/preprints202411.1209.v4

1. Remove facility data which does not have any defect.
2. Select data from a specific facility where most defect ratios exist.
3. Select data of a specific product with the largest size.

The specified dataset now has data size of 26577 and 132 features. 36 setting-based recipes with secured normality exist throughout 57 batch process from 2023-12-14 to 2024-07-17. And the descriptive statistics of 6 input features is shown in Table 2.

Table 2. Descriptive Statistics of Input Features

	Mean	Std	Min	25%	50%	75%	Max
Injection Time	2.33	0.08	2.19	2.30	2.340	2.35	6.39
Switch Position	13.46	2.83	7.99	12.00	13.50	15.99	21.0
Cushion Distance	11.29	2.72	5.88	9.81	11.39	13.69	18.56
Weight Time	24.10	2.76	17.33	23.85	24.26	24.52	172.03
Max Injection Press	151.52	2.52	118.73	149.91	151.40	153.17	171.26
Peak Pressure	13321.01	12.31	13260.5	13312.6	13323.0	13330.4	13358.9

4.4. Train / Test Data Organization

Referred to Section 3.3, 6 recipes remain from the experiment dataset. The description of the train and test data is shown in Tables 3 and 4. In case of recipe 4,5,6, only training is available since there are no test data with identical recipes. In this case, prediction can only be done if new data of same recipes are actually collected. This gives an information that from the original 8190 data, 3 setting-specific subsets are controllable for data-driven AI approach. Validation data are then defined. Finally, Min-Max regularization is performed as referred to 3.4. With the organization of the train and test data, the prediction of the integrated and recipe-based models follows the next step in 4.5 .

Table 3. Train Data Description

Dataset Information	Data Shape (Existing Defect Ratio=0.0%)	Test Appliable
Integrated Data	8190 Data Size and 6 Features	O
Recipe1 Data	2654 Data Size and 6 Features	O
Recipe2 Data	1996 Data Size and 6 Features	O
Recipe3 Data	3073 Data Size and 6 Features	O
Recipe4 Data	435 Data Size and 6 Features	X
Recipe5 Data	31 Data Size and 6 Features	X
Recipe6 Data	1 Data Size and 6 Features	X

Table 4. Test Data Description

Dataset Information	Data Shape (Existing Defect Ratio>0.0%)
Integrated Data	5842 Data Size and 6 Features
Recipe1 Data	4396 Data Size and 6 Features
Recipe2 Data	475 Data Size and 6 Features
Recipe3 Data	971 Data Size and 6 Features

4.5. Results

4.5.1. Autoencoder Configuration

The information of the trained Autoencoders is described in Table 5. Stacked Autoencoders [33] with the best weights [34] throughout the epochs of the training process are defined as the basic structure. To maximize performance, we optimally selected the parameters for each model.

 doi:10.20944/preprints202411.1209.v4

Table 5. Parameter Descriptions of Trained Autoencoders

Applied Parameters	Integrated Data	Recipe 1	Recipe 2	Recipe 3
Loss	Mean Absolute Error			
Activation Function	Tanh			
optimizer	Adam			
Learning Rate	0.01			
Number of Epochs	200			
Early Stopping	200			
Output Layer Size	128	128	64	64
Inner Layer Size	64	64	32	32
Latent Vector Size	16	4	16	16
Dropout Ratio	0.15	0.1	0.15	0.15
Batch Size	100	10	10	10
Validation Split Ratio	0.15	0.1	0.15	0.15

4.5.2. Threshold Optimization

To define optimal thresholds for anomaly detection, the validation data mentioned in 3.3 and 4.4 are first predicted with each Autoencoder. The distribution of reconstruction errors are shown in Figure 8. The optimal thresholds are selected based on the maximum MAE values referred to 3.5.2. Instead of calculating thresholds via statistical approach, the maximum MAE value is selected to ensure the highest objectivity. The calculated thresholds are described in Table 6.

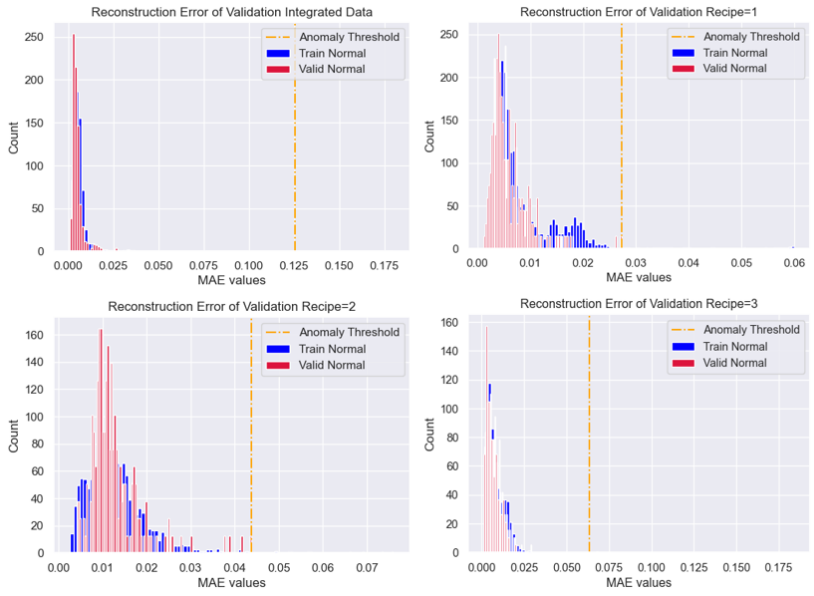

Figure 8. Validation Thresholds of Reconstruction Errors

Table 6. Thresholds for Anomaly Detection

Autoencoder	Validation Maximum Threshold
Integrated	0.125
Recipe1	0.0273
Recipe2	0.0437
Recipe3	0.0635

4.5.3. Prediction Comparisons

With the defined thresholds, the prediction results based on the integrated and recipe-based models are described in Table 7. From the table, each information includes the processed date of the batch with the existing and predicted defects. For recipe-based prediction, the specific number of the

 doi:10.20944/preprints202411.1209.v4

recipe is also shown. For example, in the first information, there are 5 existing defects among 971 cavities, 1 predicted defect from the integrated model, and 6 predicted defects from the recipe-based model. The recipe number based on the specific settings of the trained model and the test data is 3.

Compared to the total number of 41 existing defects in 5842 cavities(0.701%), the integrated model without considering the settings predicted only 2(0.034%) defects, while the recipe-based models predicted 61 defects(1.04%). In terms of accuracy, recipe-based prediction shows approximately more than twice of the integrated-based prediction. Especially in the second and eighth information, the recipe-based models predicted the same number of defects compared to the existing ones. However, none were predicted with the integrated model. This implies the importance of building appropriate prediction models considering the difference of data distributions due to change of settings.

Table 7. Prediction Comparison of Defect Ratios

Batch No	Existing Defect Ratio	Integrated pred	Recipe Pred	Recipe No.
20240322	0.514(5/971)	0.102(1/971)	0.617(6/971)	3
20240402	0.319(1/313)	0.00(0/313)	0.319(1/313)	1
20240429	0.879(6/682)	0.00(0/682)	1.173(8/682)	1
20240430	0.833(7/835)	0.00(0/835)	1.556(13/835)	1
20240502	0.611(5/817)	0.00(0/817)	1.22(10/817)	1
20240507	1.920(7/368)	0.00(0/368)	0.271(1/368)	1
20240614	1.312(5/381)	0.262(1/381)	2.099(8/381)	1
20240617	0.647(2/309)	0.00(0/309)	0.647(2/309)	1
20240623	0.289(2/691)	0.00(0/691)	0.723(5/691)	1
20240717	0.210(1/475)	0.00(0/475)	1.473(7/475)	2
Total	0.701%(41/5842)	0.034%(2/5842)	1.04%(61/5842)	

4.6. Adaptable Learning

Referring to Section 3.2 and 3.6, we used new inputs predicted as recipe numbers 6 and 7 . The characteristics of the new data are as follows.

- For recipe 6, there exist two unique batches each with no defects(defect ratio=0.0%) and existing defects(defect ratio>0.0%). In other words, the first batch consists of only good cavities, and the second consists of both good and bad cavities.
- For Recipe 7, there exists a process of one unique batch with existing defects.

For both recipes, data which contain defects are predicted with the following approaches.

1. Find each nearest trained recipe data and its prediction model(AutoEncoder) of recipe numbers 6 and 7 using the KL-Divergence calculation.
2. Optimize thresholds with validation data.
3. Predict new data with each selected Autoencoder and the integrated AutoEncoder referred to Table 5 .
4. Compare prediction results of integrated and recipe-based models.

For the batch process data with no existing defects in recipe 6, it is also used to train a new model. Then, the batch process data with existing defects are predicted. This is held to compare the anomaly detection results of additional training and adaptable learning.

4.6.1. KL-Divergence Calculation

With defined parameters $\boldsymbol{R}$=1,$\boldsymbol{K}$=2, r=$\boldsymbol{6}$, k=$\boldsymbol{1}$, N= $\boldsymbol{76}$, $\boldsymbol{I}$=6 according to 3.6, the calculation results are described in Table 8. As a result, the closest trained recipe dataset of the new inputs can be defined as both recipe 1.

 doi:10.20944/preprints202411.1209.v4

Table 8. Calculation Result of KL-Divergence Values

New Recipes	Trained Recipe1	Trained Recipe2	Trained Recipe3
Recipe6	40.1	41.63	41.12
Recipe7	11.13	12.66	12.15

4.6.2. Data Organization

As 3.5.2 and 4.5.2, threshold optimization is held based on the validation dataset of normal cavities. The validation data for the new inputs are defined in Table 9. Data are defined by whether batch process from a new data consists only of normal cavities can be known or not.

First, for recipe 6, the first batch(defect ratio=0.0%) is used for validation and the second batch(defect ratio>0.0%) is used for prediction. Second, for recipe 7, normal the validation data is defined as follows.

1. Based on recipe 1, select the closest data among the remaining recipe 2 and 3.
2. Parameters according to 3.6 are defined as $\boldsymbol{R}$=2, $\boldsymbol{K}$=1, $\boldsymbol{r}$=1, $\boldsymbol{k}$=0, $\boldsymbol{N}$=76 , $\boldsymbol{I}$=6.
3. With KL-Divergence, the closest data of recipe 1 result in recipe 3.
4. The existing train data of recipe 3 is decided as the validation dataset.

Table 9. Data Organization for Adaptable Learning

Train Data	Validation Data	Prediction Data
Recipe1	Recipe6(Normal)	Recipe6(Defect)
Recipe1	Recipe3	Recipe 7

4.6.3. Prediction Comparisons

In Table 9, thresholds for anomaly detection are optimized via the maximum MAE values of the reconstruction error distributions. Then the adaptable learning procedure is held with the prediction data. The results compared to the integrated model are described in Table 10. Especially for recipe number 6, the prediction result of additional training is also compared.

While the integrated model without considering unique settings still fails to predict defects as 4.5.3, results based on the adaptable learning process predicted relatively acceptable amount of defects. In recipe number 6, the prediction except for the integrated model resulted in noticably more defects compared to the existing ones. Particularly, adaptable learning led to a better result than additional training with 10 less predicted anomalies. And in recipe number 7, predicted anomalies with adaptable learning are almost equal to the existing ones. This implies that applying adaptable learning for new data is more effective in the perspective of prediction accuracy and the continuous control of quality inspection.

Table 10. Prediction Comparisons of new recipes

Recipe No.	Batch No	Defect Ratio	Integrated	Adaptable	Additional
6	20240903	0.293(2/682)	0.00(0/682)	4.692(32/682)	6.158(42/682)
7	20240902	0.546(4/728)	0.0(0/728)	0.683(5/728)	X

5. Discussion

For comparison, both the integrated model and setting-specific recipe models used stacked autoencoders of a simple structure. And recognizing the importance of hyper-parameter tuning [36–38], this was performed for each model.

When using the trained results of the integrated model and the models classified by recipes, the thresholds of the recipe-based models can be set more precisely than that of the integrated model. When this is utilized to classify good and defective products, it can be seen that the defective classification

 doi:10.20944/preprints202411.1209.v4

rate for each of the recipe data is significantly predicted better than that of the integrated model. This underscores the importance of categorizing the controllable data that vary with settings, ensuring the normality of each dataset during training. In other words, it shows that it is very important to train, ensuring the normality of each data by classifying the data distributions that change according to the setting for production efficiency.

Also, when predicting data that have not been previously trained, a KL-Divergence approach to identify the closest recipe distribution further demonstrated higher accuracy in distinguishing defective from non-defective products compared to the integrated model and additionally trained model. This method expands the control of identifiable settings and reduces the need for frequent retraining while maintaining high classification performance. The targeted nature of recipe-specific models streamlines data processing, minimizes noise, and improves accuracy by addressing setting-specific variations that the integrated model might overlook. Therefore, since these models avoid the broader generalizations required by the integrated model, both the improvement of defect detection rates and optimization of computational efficiency are enhanced. This adaptability ensures scalability and robustness, particularly for large-scale industrial applications.

An additional approach leveraging fine-tuning thresholds without extensive retraining further emphasizes the potential of setting-specific modeling. By continuous inference, this method achieves high prediction accuracy while optimizing computational resources. Such advancements highlight the transformative potential of adaptive modeling in dynamically changing industrial environments, enabling leaner, more efficient AI applications.

In sum, the findings demonstrate the efficacy of AI modeling specialized in a data-driven approach over traditional methods in the injection molding process. These methods address the variability inherent in operational settings, ensuring improved prediction quality and reduced resource demands. The move toward context-driven methodologies represents a significant step forward in the development of robust and efficient AI systems for sustainable manufacturing.

6. Conclusions

As AI becomes integral to manufacturing, its role in fostering efficiency and sustainability gains prominence. AI-driven systems can optimize data process, enabling precise defect identification and waste reduction. This focus on quality control not only enhances competitiveness, but also supports environmental goals by minimizing resource waste in the aspect of smart factory.

When utilizing AI in actual manufacturing sites, environmental benefits must also be considered. The benefits of AI for environmental sustainability have been clearly demonstrated in small-scale project initiatives and hypothetical or qualitative tests[39–41]. However, to achieve a meaningful impact from the perspective of AI and sustainability, it is essential not only to identify and improve environmental factors but also to prioritize factors that are competitive and beneficial to business[42,43].

Compared to existing systems, this will present a new recipe unit-based paradigm of advanced data-driven decision making focused on reducing waste, compressing the weight of computed models and improving prediction accuracy. This thereby leads to saving the cost of real-time inference. In addition, it will promote research in new directions and engage in continuous development. By utilizing these AI advantages, this study can emphasize that in the area of reducing environmental costs, which is a factor that has recently emerged, resource waste can also be reduced.

The development of optimized, setting-specific datasets allows the deployment of simpler, less resource-intensive models without compromising accuracy. This streamlined data-driven AI approach lowers computational costs and fosters broader adoption of AI in sustainable manufacturing. Incorporating domain-specific knowledge further refines these models, enhancing their interpretability and predictive capabilities while promoting proactive quality control.

However, in reality of the manufacturing process environment, only a few batches of product data can be accurately identified for good products, so semi-supervised learning must also be utilized. And

 doi:10.20944/preprints202411.1209.v4

due to the characteristics of semi-supervised learning, the classification accuracy is not maintained at a very high level.

Future research should focus on further integration of domain expertise, advanced feature selection techniques, and optimized data practices based on the recipe-level analysis outlined in this paper. Specifically, utilizing domain knowledge to identify key variables that influence the distinction between high- and low-quality products, combined with targeted feature engineering, can enhance predictive accuracy. These advancements will ensure that AI systems remain efficient, precise and sustainable in driving innovation across various industries while effectively addressing critical environmental challenges.

References

1. Cioffi, Raffaele, et al. "Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions." Sustainability 12.2 (2020): 492.
2. Tseng, Ming-Lang, et al. "Sustainable industrial and operation engineering trends and challenges Toward Industry 4.0: A data driven analysis." Journal of Industrial and Production Engineering 38.8 (2021): 581-598.
3. Bousdekis, Alexandros, et al. "A review of data-driven decision-making methods for industry 4.0 maintenance applications." *Electronics* 10.7 (2021): 828.
4. Waltersmann, Lara, et al. "Artificial intelligence applications for increasing resource efficiency in manufacturing companies—a comprehensive review." Sustainability 13.12 (2021): 6689.
5. Chen, Yixuan, and Shanyue Jin. "Artificial intelligence and carbon emissions in manufacturing firms: The moderating role of green innovation." Processes 11.9 (2023): 2705.
6. Rakholia, Rajnish, et al. "Advancing Manufacturing Through Artificial Intelligence: Current Landscape, Perspectives, Best Practices, Challenges and Future Direction." IEEE Access (2024).
7. Peres, Ricardo Silva, et al. "Industrial artificial intelligence in industry 4.0-systematic review, challenges and outlook." IEEE access 8 (2020): 220121-220139.
8. Arinez, Jorge F., et al. "Artificial intelligence in advanced manufacturing: Current status and future outlook." Journal of Manufacturing Science and Engineering 142.11 (2020): 110804.
9. Espina-Romero, Lorena, et al. "Challenges and Opportunities in the Implementation of AI in Manufacturing: A Bibliometric Analysis." Sci 6.4 (2024): 60.
10. Xu, Jiawen, et al. "A review on AI for smart manufacturing: Deep learning challenges and solutions." Applied Sciences 12.16 (2022): 8239.
11. Rousopoulou, Vaia, et al. "Predictive maintenance for injection molding machines enabled by cognitive analytics for industry 4.0." Frontiers in Artificial Intelligence 3 (2020): 578152.
12. Jung, Hail, et al. "Application of machine learning techniques in injection molding quality prediction: Implications on sustainable manufacturing industry." Sustainability 13.8 (2021): 4120.
13. Hozdić, Elvis. "Smart factory for industry 4.0: A review." *International Journal of Modern Manufacturing Technologies* 7.1 (2015): 28-35.
14. Charest, Meaghan, Ryan Finn, and Rickey Dubay. "Integration of artificial intelligence in an injection molding process for on-line process parameter adjustment." 2018 Annual IEEE International Systems Conference (SysCon). IEEE, 2018.
15. Silva, Bruno, et al. "Enhance the injection molding quality prediction with artificial intelligence to reach zero-defect manufacturing." Processes 11.1 (2022): 62.
16. Thabtah, Fadi, et al. "Data imbalance in classification: Experimental evaluation." Information Sciences 513 (2020): 429-441.
17. Mao, Ting, et al. "Data driven injection molding process monitoring using sparse auto encoder technique." 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM). IEEE, 2015.
18. K. P. Sinaga and M. -S. Yang. "Unsupervised K-Means Clustering Algorithm." IEEE Access, vol. 8, pp. 80716-80727, 2020.
19. Chen, Zhaomin, et al. "Autoencoder-based network anomaly detection." *2018 Wireless telecommunications symposium (WTS)*. IEEE, 2018.
20. F.Perez-Cruz, "Kullback-Leibler divergence estimation of continuous distributions." 2008 IEEE International Symposium on Information Theory, Toronto, ON, Canada, 2008, pp. 1666-1670.
21. Deepa, Bharathi and K. Ramesh. "Epileptic seizure detection using deep learning through min max scaler normalization." *International journal of health sciences* (2022).

 doi:10.20944/preprints202411.1209.v4

22. Nayak, S. C., Bijan B. Misra, and Himansu Sekhar Behera. "Impact of data normalization on stock index forecasting." *International Journal of Computer Information Systems and Industrial Management Applications* 6 (2014): 13-13.
23. S. Al-Abri, T. X. Lin, M. Tao and F. Zhang. "A Derivative-Free Optimization Method With Application to Functions With Exploding and Vanishing Gradients." IEEE Control Systems Letters, vol. 5, no. 2, pp. 587-592, April 2021.
24. Sullivan, Joe H., and William H. Woodall. "Change-point detection of mean vector or covariance matrix shifts using multivariate individual observations." *IIE transactions* 32.6 (2000): 537-549.
25. Song, Yang, Alexander Schwing, and Raquel Urtasun. "Training deep neural networks via direct loss minimization." *International conference on machine learning*. PMLR, 2016.
26. Hosen, Mohammed Shahadat, et al. "Data-Driven Decision Making: Advanced Database Systems for Business Intelligence." *Nanotechnology Perceptions* (2024): 687-704.
27. Berahmand, Kamal, et al. "Autoencoders and their applications in machine learning: a survey." *Artificial Intelligence Review* 57.2 (2024): 28.
28. Yang, Tianyi, et al. "Deep learning model-driven financial risk prediction and analysis." (2024).
29. Bariah, Lina, and Merouane Debbah. "The interplay of ai and digital twin: Bridging the gap between data-driven and model-driven approaches." *IEEE Wireless Communications* (2024).
30. Sewak, Mohit, Sanjay K. Sahay, and Hemant Rathore. "An overview of deep learning architecture of deep neural networks and autoencoders." *Journal of Computational and Theoretical Nanoscience* 17.1 (2020): 182-188.
31. Hodson, Timothy O. "Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not." *Geoscientific Model Development Discussions* 2022 (2022): 1-10.
32. Givnan, Sean, et al. "Anomaly detection using autoencoder reconstruction upon industrial motors." *Sensors* 22.9 (2022): 3166.
33. Yu, Miao, et al. "A model-based collaborate filtering algorithm based on stacked AutoEncoder." *Neural Computing and Applications* (2022): 1-9.
34. Liu, Tong, et al. "High-ratio lossy compression: Exploring the autoencoder to compress scientific data." *IEEE Transactions on Big Data* 9.1 (2021): 22-36.
35. Wang, Jianlong, et al. "Parameter selection of Touzi decomposition and a distribution improved autoencoder for PolSAR image classification." *ISPRS Journal of Photogrammetry and Remote Sensing* 186 (2022): 246-266. c
36. Awwad, Ameen, Ghaleb A. Husseini, and Lutfi Albasha. "AI-Aided Robotic Wide-Range Water Quality Monitoring System." Sustainability 16.21 (2024): 9499.
37. Tynchenko, Vadim, et al. "Predicting Tilapia Productivity in Geothermal Ponds: A Genetic Algorithm Approach for Sustainable Aquaculture Practices." Sustainability 16.21 (2024): 9276.
38. Liu, Yang, et al. "Ada-XG-CatBoost: A Combined Forecasting Model for Gross Ecosystem Product (GEP) Prediction." Sustainability 16.16 (2024): 7203.
39. Nañez Alonso, Sergio Luis, et al. "Digitalization, circular economy and environmental sustainability: The application of Artificial Intelligence in the efficient self-management of waste." Sustainability 13.4 (2021): 2092.
40. Khakurel, Jayden, et al. "The rise of artificial intelligence under the lens of sustainability." Technologies 6.4 (2018): 100.
41. Wu, Carole-Jean, et al. "Sustainable ai: Environmental implications, challenges and opportunities." Proceedings of Machine Learning and Systems 4 (2022): 795-813.
42. Tanveer, Muhammad, Shafiqul Hassan, and Amiya Bhaumik. "Academic policy regarding sustainability and artificial intelligence (AI)." Sustainability 12.22 (2020): 9435.
43. Yigitcanlar, Tan, and Federico Cugurullo. "The sustainability of artificial intelligence: An urbanistic viewpoint from the lens of smart and sustainable cities." Sustainability 12.20 (2020): 8548.