mathematics

MDPI

Article

Enhancing Real-Time Processing in Industry 4.0 Through the Paradigm of Edge Computing

Nerea Gómez Larrakoetxea *, Borja Sánz Uquijo, Iker Pastor López, Jon García Barruetabeña and Pablo García Bringas

Faculty of Engineering, University of Deusto, Unibertsitate Etorb., 24, Deusto, 48007 Bilbo, Spain; borja.sanz@deusto.es (B.S.U.); iker.pastor@deusto.es (I.P.L.); jgarcia.barruetabena@deusto.es (J.G.B.); pablo.garcia.bringas@deusto.es (P.G.B.)
* Correspondence: ngomez@deusto.es

Abstract: The industrial sector has undergone significant digital transformation, driven by advancements in technology and the Internet of Things (IoT). These developments have facilitated the collection of vast quantities of data, which, in turn, pose significant challenges for real-time data processing. This study seeks to validate the efficacy and accuracy of edge computing models designed to represent subprocesses within industrial environments and to compare their performance with that of traditional cloud computing models. By processing data locally at the point of collection, edge computing models provide substantial benefits in minimizing latency and enhancing processing efficiency, which are crucial for real-time decision-making in industrial operations. This research demonstrates that models derived from distinct subprocesses yield superior accuracy compared to comprehensive models encompassing multiple subprocesses. The findings indicate that an increase in data volume does not necessarily translate to improved model performance, particularly in datasets that capture data from production processes, as combining independent process data can introduce extraneous 'noise'. By subdividing datasets into smaller, specialized edge models, this study offers a viable approach to mitigating the latency challenges inherent in cloud computing, thereby enhancing real-time data processing capabilities, scalability, and adaptability for modern industrial applications.

Keywords: edge computing; real-time data processing; data modeling; industrial applications

MSC: 68-11; 68-00

Academic Editor: Xiang Li

Received: 2 December 2024
Revised: 19 December 2024
Accepted: 23 December 2024
Published: 26 December 2024

Citation: Gómez Larrakoetxea, N.; Sánz Uquijo, B.; López, I.P.; Barruetabeña, J.G.; Bringas, P.G. Enhancing Real-Time Processing in Industry 4.0 Through the Paradigm of Edge Computing. *Mathematics* **2025**, *13*, 29. https://doi.org/10.3390/math13010029

1. Introduction

In recent years, the industrial sector has experienced a substantial digital transformation, leading to the collection of vast amounts of data from various production lines [1]. This digitization, driven by advancements in technology and the integration of the Internet of Things (IoT), has facilitated the development of digital twins that represent industrial processes through extensive datasets. The volume of data collected in these environments is immense [2], often capturing thousands of variables per second from numerous sensors embedded throughout the production lines. This influx of data presents both opportunities and challenges for real-time data processing and decision-making.

The primary objective of this research is to validate the efficacy and accuracy of edge computing models that represent subprocesses within industrial settings and to compare these models with the traditional Cloud Computing paradigm. The rationale behind this study stems from the observation that, generally, smaller datasets are processed

more rapidly by models during both training and testing phases. However, as industries continue to digitize, the sheer volume of data collected poses significant challenges to efficient processing. Traditional cloud computing architectures, despite their extensive computational resources, suffer from inherent latencies due to data transmission over networks, which can impede real-time responsiveness [3].

Edge computing directly supports the goals of Industry 4.0 by decentralizing processing tasks, reducing reliance on centralized infrastructure, and significantly minimizing latency. This decentralized approach enables real-time responsiveness, a critical requirement for smart manufacturing systems.

This challenge is particularly pronounced in sectors that demand immediate responses from their systems. For instance, the automotive industry, among others, requires real-time monitoring and control to maintain high-quality standards and operational efficiency [4,5]. In such environments, digital twins are utilized to replicate and monitor the production processes, but the high data volumes can overwhelm traditional data processing systems, leading to delays and inefficiencies [6].

To address these issues, this study proposes an alternative approach by subdividing the dataset representing an entire industrial process into smaller, more manageable sets, referred to as subprocesses. Instead of developing a single comprehensive model that encompasses the entire production chain, the focus is on creating smaller, specialized "Edge" models that represent individual subprocesses within the production line. These edge models are designed to operate at or near the source of data collection, thereby reducing latency and improving the speed of data processing and anomaly detection.

The effectiveness of these smaller edge models will be validated through a series of experiments, ensuring that their performance is not inferior to that of a unified model representing the complete process. By leveraging edge computing, this research aims to enhance real-time data processing capabilities, thus providing a viable solution to the latency issues associated with cloud computing. This approach not only promises faster response times but also scalability and flexibility, which are crucial for modern industrial operations that continuously evolve and expand.

In conclusion, this study offers a unique contribution to the growing body of research on edge computing by focusing on the development of specialized models tailored to specific industrial subprocesses. Unlike existing works that often emphasize generalized models or cloud-based approaches, this research addresses the critical need for localized processing in real-time applications, where latency and immediate responsiveness are paramount.

A key differentiator of this work lies in its use of real-world datasets from industrial processes, such as vehicle painting and multi-stage manufacturing, to validate the efficacy of edge computing models. While prior studies have highlighted the advantages of edge computing in theoretical or simulated environments [7,8], this study provides empirical evidence from real manufacturing scenarios, showcasing the practical applicability of these models.

2. Background

Edge computing has gained traction in industrial applications due to its ability to process data locally, reducing latency and improving response times. Previous studies have highlighted the advantages of edge computing in handling smaller-scale datasets effectively. Already in 2018, a study concluded [9] that the architecture of edge computing is beneficial for applications involving smaller-scale datasets by enabling more efficient data processing and quicker decision-making without the need to transmit large amounts of data to a central cloud. Building on this foundation, further research has explored how

to optimize the performance of edge computing systems. For instance, Aral et al. [10] proposed a scheduling algorithm that assesses edge node capabilities to optimize service quality and reduce latency, demonstrating significant improvements in network delay and service time. This algorithm enhances the efficiency of edge computing by ensuring that computational tasks are allocated to the most suitable edge nodes, thereby minimizing latency and maximizing overall system performance.

Moreover, edge computing is particularly effective in industrial settings due to its ability to process data at the edge of the network, which significantly decreases latency and boosts response times for real-time applications. This capability is crucial in industrial automation systems where real-time responses are paramount. Zhang et al. [11] emphasize that the reduction in latency and improvement in response times achieved through edge computing are essential for maintaining the operational efficiency and reliability of industrial automation systems. By processing data closer to the source, edge computing enables faster decision-making and enhances the performance of real-time industrial applications.

Further emphasizing the importance of low-latency processing, Bacchiani et al. [12] discussed the SEAWALL platform, which is designed for low-latency anomaly detection in Industry 4.0 environments. This platform highlights the benefits of edge computing in reducing alert service latency, underscoring its critical role in maintaining timely and effective responses in industrial applications. By leveraging edge computing, SEAWALL enhances the capability of industrial systems to promptly detect and respond to anomalies, thereby ensuring smoother and more reliable operations. Complementing this, Abouaomar et al. [13] describe how dynamic resource allocation at edge devices enhances the low-latency response for IoT and other latency-sensitive applications. Their work illustrates the significant improvements in service quality and response times that can be achieved through adaptive resource provisioning, which is particularly beneficial for maintaining efficiency and reliability in industrial settings.

The scope of edge computing in managing varying data volumes is primarily centered on localized, real-time processing of moderate-scale datasets. The performance and capacity of edge devices depend on factors such as the number of inputs (e.g., sensors, IoT devices) and the computational resources available. Typically, edge devices are capable of handling hundreds to thousands of sensor inputs within a localized process. However, when dealing with larger-scale data streams, such as those aggregating inputs from multiple production lines or enterprise systems, a hybrid approach that incorporates cloud computing may be required.

The behavior and limitations of edge computing under varying data loads have been analyzed in a previously published study by the authors [14]. This study includes stress tests and performance evaluations for edge devices subjected to increasing data volumes. The results indicate that edge devices can effectively process up to 900 MB before experiencing performance degradation, at which point, cloud-based solutions may become necessary to maintain system efficiency.

This study aligns with the recent advancements in edge computing and Industry 4.0, which emphasize the need for real-time processing and localized decision-making in industrial settings. Recent works have explored the integration of edge computing with cutting-edge technologies such as artificial intelligence, blockchain, and 5G to address challenges in smart factory applications [15–17].

In particular, the potential of edge computing for improving latency and scalability in industrial systems has been highlighted, along with its role in supporting cyber–physical systems in manufacturing environments [7,8]. These insights underline the growing importance of edge computing in handling the increasing complexity and data volumes inherent in Industry 4.0.

3. Materials and Methods

For this experiment, we utilized proprietary industry datasets to enhance the realism and applicability of our findings. Addressing network monitoring issues, such as anomaly detection through sophisticated machine learning methods, we prioritized datasets that closely reflect real-world scenarios encountered in the industry. By leveraging these datasets, our study aims to provide a more accurate assessment of the performance and efficacy of various machine learning models in detecting anomalies within network traffic. This approach ensures that our experimental results are not only theoretically sound but also practically relevant, thereby contributing valuable insights to the field of network monitoring and anomaly detection.

3.1. Dataset

The datasets utilized in the experimentation are fourfold, offering a more comprehensive evaluation of real-time processing in Industry 4.0 applications. One dataset was directly obtained from an automobile manufacturing plant, while the other three are publicly available datasets focused on condition monitoring and continuous-flow manufacturing processes.

- **Vehicle Painting dataset:** This dataset was obtained from the painting process of a real vehicle manufacturing plant, with which I collaborated during my doctoral thesis to validate experiments in real environments with real data. The data were collected by various sensors throughout the process. The dataset includes 9 temperature sensors in the enamel oven, 2 humidity sensors inside the paint booth, and 2 humidity sensors at the exit of the paint booth. Additionally, it features an external humidity sensor, pressure sensors in the paint booth, and temperature sensors at various points within the paint and preparation areas. This diverse array of sensors ensures comprehensive monitoring and control of the painting environment.
  Overall, each vehicle collects a series of variables, and if it does not pass the quality control, it goes through the process again, potentially up to three times. In addition to the sensor data, the dataset also includes the characteristics of each vehicle: body color, number of painting rounds, vehicle model, and length. Additionally, as an output label, the type of defect detected by the final quality control has been recorded, with 50 possible defects.
- **Bosch dataset:** This dataset [18] consists of data acquired during the manufacturing process, which is divided into four production lines (L0, L1, L2, L3). Each line has generated a separate dataset, and the label used for all is whether it passes quality control or not. The dataset provides detailed sensor data for each stage of the production, allowing for an in-depth analysis of quality assurance processes.
- **Condition Monitoring of Hydraulic Systems dataset (ZeMA):** This dataset focuses on condition assessment of a hydraulic test rig based on multi-sensor data. It captures the behavior of four hydraulic components—cooler, valve, pump, and accumulator—under various operational conditions. The rig operates in cycles (60 s each), and data are collected on pressures, volume flows, and temperatures.The dataset [19] contains raw process data, structured as matrices, where each row represents a cycle, and columns represent sensor readings. Key sensors include six pressure sensors, two volume flow sensors, motor power, temperature sensors, and a vibration sensor, among others. The condition of the hydraulic components is annotated with different severity levels for cooler efficiency, valve switching behavior, internal pump leakage, and accumulator pressure. The dataset has been further divided into four subprocesses based on the components monitored: cooler, valve, pump, and accumulator.
- **Multi-Stage Continuous-Flow Manufacturing Process dataset:** This dataset [20] originates from a high-speed, continuous manufacturing process with parallel and

series stages. The data were collected from a production line in Michigan. The first stage involves three machines operating in parallel, whose outputs are combined, and measurements are taken at 15 locations around the material exiting the combiner. These measurements form the primary prediction target. In the second stage, the combined output is processed by two additional machines in series, with the same 15 locations measured again for the secondary prediction target. The data reflect a continuous flow process, offering insights into multi-stage manufacturing, and they have been divided into two distinct stages: the first stage (parallel machines) and the second stage (series machines).

The datasets utilized in this study were processed using a structured pipeline tailored to optimize real-time data handling within edge computing environments. The pipeline consists of the following steps:

- Normalization: Sensor readings, such as temperature, pressure, and humidity, were normalized to a range of [0, 1] to ensure comparability across variables and to improve the stability of machine learning algorithms.
- Pivoting: For time-series data, pivoting was applied to structure the data such that each row represents a specific time interval or production cycle, and columns represent sensor readings or process characteristics.
- Handling Missing Values: Missing data points were imputed using interpolation methods for continuous variables and mode imputation for categorical variables, ensuring data consistency.
- Outlier Detection: Outliers were identified using z-scores ($|z| > 3$) and removed or capped to reduce noise in the dataset.

3.2. Data Processing

In this subsection, a brief explanation is provided on how the data were prepared. This includes pivoting the data to reorganize and structure the information in a way that is more useful and easier to analyze. Additionally, the data were normalized to standardize the variables and ensure that all data points are within a comparable range, thereby eliminating any disparities that could affect the analysis results.

- For the vehicle painting data set, as observed, the number of variables collected per vehicle is substantial, and it can triple if the vehicle requires rework twice. Within the framework of this experimentation, the dataset has been divided into two parts:
  - **Enamel subprocess:** This includes the variables related to the enameling process, along with the characteristics of each vehicle.
  - **Primer coat subprocess:** This includes the variables related to the primer coat process, along with the characteristics of each vehicle.

  Therefore, from an initial dataset containing variables for both enamel and primer coat processes, two distinct datasets have been created, one for each subprocess. Additionally, extensive preprocessing was required to pivot the data so that each row represents a vehicle, and each column represents a sensor value for that vehicle. Characteristics of each vehicle were then added to these data sets.
- Bosch dataset: The main preprocessing involved dividing the dataset into separate production lines (L0, L1, L2, and L3). Each line's data were treated as an individual dataset, with relevant variables grouped per production stage. The dataset was pivoted so that each row corresponds to a product, and each column reflects sensor measurements or production characteristics. The label for quality control, indicating whether a product passed or failed, was retained as the output variable for each production line.

- Condition Monitoring of Hydraulic Systems dataset (ZeMA): The preprocessing for this dataset focused on the four hydraulic components monitored: cooler, valve, pump, and accumulator. The raw sensor data, initially structured in cycles, were pivoted so that each row represents a cycle, and each column contains sensor readings from that cycle. Each hydraulic component's condition (cooler efficiency, valve behavior, pump leakage, accumulator pressure) was used as the target variable. Data normalization was applied to standardize pressure, temperature, volume flow, and vibration readings, ensuring consistent scales across all sensors. Additionally, any incomplete cycles were discarded to maintain data integrity.
- Multi-Stage Continuous-Flow Manufacturing Process dataset: This dataset was divided into two stages:
  - **First stage**: The output from Machines 1, 2, and 3 was measured at 15 locations around the material exiting the combiner. Data preprocessing involved restructuring the dataset so that each row represents a run of the process, with columns corresponding to sensor measurements at the 15 output locations.
  - **Second stage**: Similarly, the output from Machines 4 and 5 was processed, and the measurements at the same 15 locations were used as the primary features for this stage.

  The data from both stages were normalized to ensure comparability across the different machine outputs and sensor readings. The dataset was pivoted to ensure that each row corresponds to a production run, and sensor readings from both stages were aligned for predictive analysis of the final output properties.

4. Results

In this section, the results obtained from each experiment with each of the datasets are explained.

4.1. Bosch Dataset Experimentation

Once the original dataset was divided into subsets, each corresponding to a production line, experimentation was conducted. These subsets were processed using different families of algorithms to evaluate their effectiveness and compare it to the effectiveness of the complete dataset. The algorithms used were random forest, logistic regression, K nearest neighbors, neural network, and Gaussian naive Bayes. Tables 1 and 2 show the effectiveness results for each algorithm based on accuracy.

The results shown in Tables 1 and 2 are explained as follows: The first column, labeled L0 + L1 + L2 + L3, refers to the complete original dataset, as it contains data from all four production lines. The subsequent columns, L0, L1, L2, and L3, refer to the subsets created for each production line.

Table 1. Model comparison.

	L0 + L1 + L2 + L3 (%)	L0 (%)	L1 (%)
Random Forest	98.30	98.39	98.32
Logistic Regression	98.16	98.26	98.26
K Nearest Neighbor	98.17	97.94	97.98
Neural Network	91.11	95.66	98.24
Gaussian Naive Bayes	61.82	75.83	98.30

Table 2. Model Comparison.

Model	L0 + L1 + L2 + L3 (%)	L2 (%)	L3 (%)
Random Forest	98.30	98.33	98.68
Logistic Regression	98.16	98.26	98.23
K Nearest Neighbor	98.17	97.97	98.15
Neural Network	91.11	91.52	98.30
Gaussian Naive Bayes	61.82	98.30	95.70

The following chart clearly shows the comparison of results (Figure 1):

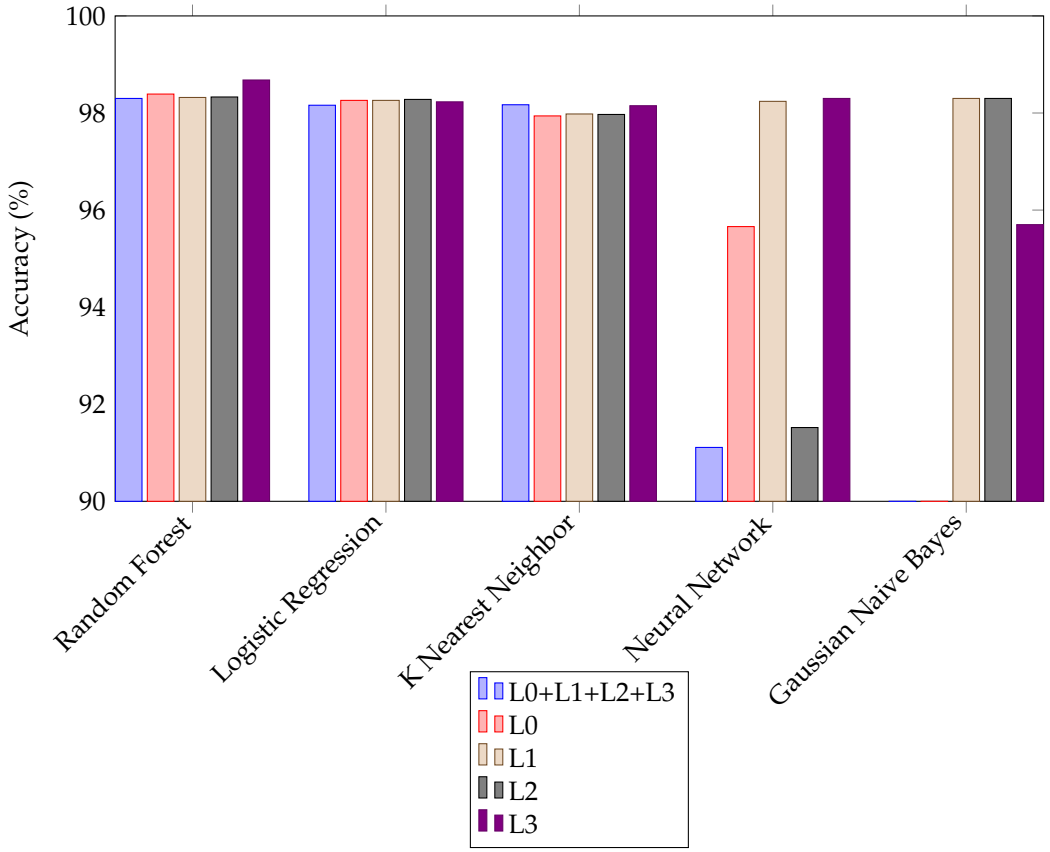

Figure 1. Model performance accuracy across comparison.

4.2. Vehicle Painting Dataset Experimentation

During the experimentation in this case, due to the large volume of variables, only neural networks provided promising accuracy results. The rest did not even reach 40% accuracy, so the experimentation focused on analyzing the results of the models with different neural network architectures. In addition to the different architectures, comparisons were also made with different optimizers, as their function is to reduce the network error by optimizing the values [21].

Following this, experimentation was conducted using the unified data from the entire painting process, encompassing the subprocesses of enameling and primer coat.

As can be observed in Table 3, the experimentation was conducted with neural networks of different configurations. Therefore, for each combination of hidden layers, both the Adam optimizer and the SGD optimizer were used.

As can be seen, in the case of the unified model, the accuracy is quite low. The configuration that yields the best results has (288, 240) layers and uses the Adam optimizer.

Table 3. Accuracy results for different configurations of layers and optimizers.

Hidden Layers	Adam (%)	SGD (%)	Hidden Layers	Adam (%)	SGD (%)
(288, 64)	26.04	26.04	(256, 256)	25.52	25.87
(288, 80)	28.07	27.43	(272, 256)	25.56	26.39
(288, 96)	28.30	26.91	(272, 272)	25.57	26.91
(288, 112)	28.71	23.61	(304, 256)	26.07	27.08
(288, 128)	27.67	25.69	(256, 272)	26.57	27.03
(288, 144)	25.80	26.49	(272, 272)	26.45	26.91
(288, 160)	27.19	26.47	(288, 272)	26.55	27.01
(288, 176)	27.95	27.53	(304, 272)	26.63	26.74
(288, 192)	27.60	25.35	(272, 288)	26.14	26.99
(288, 208)	28.60	27.17	(304, 288)	26.95	26.97
(288, 224)	28.37	26.78	(288, 288)	27.13	27.03
(288, 240)	27.97	27.09	(304, 288)	27.62	27.17
(288, 256)	27.95	26.91	(304, 304)	28.28	28.30

4.2.1. Enamel Subprocess Results

Next, experimentation was conducted using only the data from the enameling subprocess.

With the results in view in Table 4, it is important to highlight that having more layers does not necessarily lead to better outcomes. The configuration that yielded the best results consists of (256, 192) layers and uses the Adam optimizer. This configuration achieved a precision of 78.10%, the highest among all configurations tested.

Table 4. Model Comparison with Adam and SGD Optimizers.

Hidden Layers	Adam (%)	SGD (%)	Hidden Layers	Adam (%)	SGD (%)
(256, 64)	77.09	76.39	(272, 256)	77.94	76.47
(256, 80)	77.24	76.24	(288, 256)	76.63	76.93
(256, 96)	76.32	75.54	(304, 256)	75.97	75.62
(256, 112)	77.00	75.88	(272, 272)	77.19	76.63
(256, 128)	75.93	77.44	(288, 272)	76.66	75.94
(256, 144)	76.55	77.14	(304, 272)	75.70	75.62
(256, 160)	76.47	76.55	(272, 288)	77.00	76.47
(256, 176)	76.76	77.03	(288, 288)	75.93	77.14
(256, 192)	77.18	76.63	(304, 288)	76.24	76.70
(256, 208)	76.86	76.93	(272, 304)	77.01	77.03
(256, 224)	77.32	76.24	(288, 304)	75.77	75.47
(256, 240)	76.80	75.94	(304, 304)	76.86	75.54
(256, 256)	77.40	75.85			

4.2.2. Primer Coat Subprocess

Table 5 presents a summary of the accuracy results from the experimentation with the primer coat data model. In this case, the configuration that achieved the highest accuracy consists of (208, 128) layers and uses the Adam optimizer, with an accuracy of 67.01%.

Table 5. Results of the models with different hidden layer configurations and Adam and SGD optimizers.

Hidden Layers	Adam (%)	SGD (%)	Hidden Layers	Adam (%)	SGD (%)
(208, 64)	65.10	65.10	(256, 256)	64.58	63.72
(208, 80)	63.54	63.02	(272, 256)	64.54	64.41
(208, 96)	63.71	63.32	(288, 256)	64.63	64.11
(208, 112)	63.19	62.19	(304, 256)	64.56	64.15
(208, 128)	63.71	64.76	(256, 272)	64.63	63.63
(208, 144)	63.71	64.11	(272, 272)	63.95	63.28
(208, 160)	63.19	63.71	(288, 272)	64.51	63.72
(208, 176)	65.62	63.32	(304, 272)	64.40	63.85
(208, 192)	65.62	63.19	(272, 288)	63.97	63.54
(208, 208)	64.76	63.89	(288, 288)	65.97	63.63
(208, 224)	63.24	63.89	(304, 288)	64.51	62.78
(240, 240)	63.02	64.41	(288, 304)	64.41	63.19
(304, 240)	64.24	62.85	(304, 304)	63.37	64.24

4.2.3. Results Comparison

Once the accuracy results for each case were obtained, Table 6 reflects the comparison of model accuracy, using the best results from the experimentation for each case (Figure 2).

Table 6. Comparison of Model Results.

Enamel Edge Model (%)	Preparation Edge Model (%)	Preparation and Enamel Model (%)
78.10	67.01	39.86

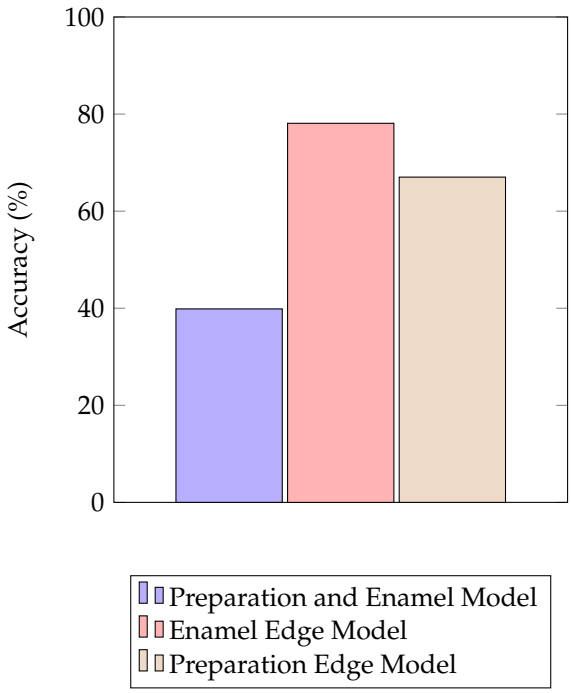

Figure 2. Comparison of accuracy results plotted.

4.3. ZEMA Dataset Experimentation

The condition monitoring of hydraulic systems dataset (ZEMA) was divided into four subprocesses: cooler, valve, pump, and accumulator. Experimentation was conducted using five machine learning algorithms: random forest, logistic regression, K nearest neighbors

(KNN), neural network, and Gaussian naive Bayes. The accuracy results for each algorithm across the four subprocesses (Figure 3), as well as the global accuracy, are presented in Tables 7 and 8.

Table 7. Accuracy Results for ZEMA Dataset (Cooler, Valve, Pump).

Model	Global Accuracy	Cooler Accuracy	Valve Accuracy	Pump Accuracy
Logistic Regression	0.9502	0.9985	1.0000	0.9985
Random Forest	0.9698	1.0000	1.0000	0.9955
KNN	0.7719	0.9955	0.9094	0.9607
Neural Network	0.7477	0.9985	0.9622	0.8640
Gaussian Naive Bayes	0.2024	0.9320	0.8399	0.5574

Table 8. Accuracy Results for ZEMA Dataset (Accumulator).

Model	Global Accuracy	Accumulator Accuracy
Logistic Regression	0.9502	0.9502
Random Forest	0.9698	0.9743
KNN	0.7719	0.8731
Neural Network	0.7477	0.8686
Gaussian Naive Bayes	0.2024	0.4063

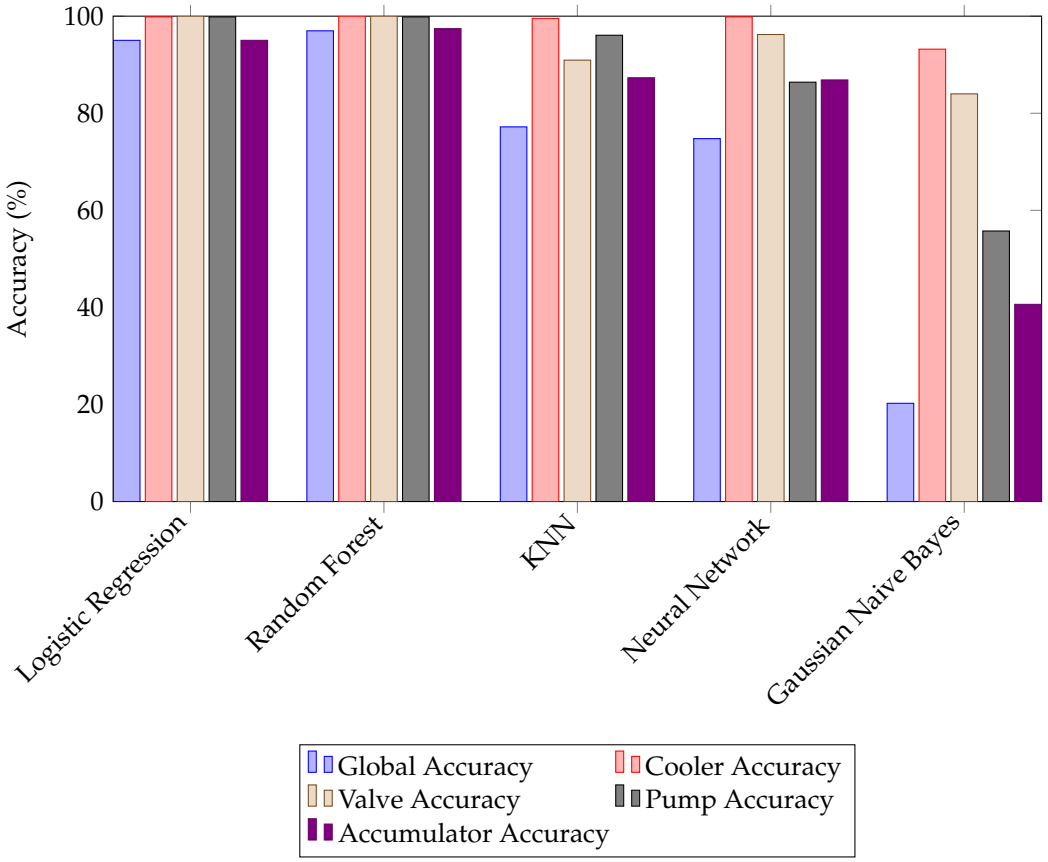

Figure 3. Comparison of accuracy results plotted for ZEMA Dataset (Cooler, Valve, Pump, Accumulator).

4.4. Multi-Stage Continuous Flow Manufacturing Process Dataset Experimentation

The multi-stage continuous flow manufacturing process dataset was divided into two stages: the first stage (parallel machines) and the second stage (series machines). Experimentation with the same five machine learning algorithms yielded the accuracy results shown in Table 9 (Figure 4).

Table 9. Accuracy Results for Multi-Stage Dataset.

Model	Global Accuracy	First Stage Accuracy	Second Stage Accuracy
Logistic Regression	0.9579	0.9734	0.9605
Random Forest	0.9757	0.9862	0.9766
KNN	0.9479	0.9665	0.9489
Neural Network	0.9283	0.9656	0.9863
Gaussian Naive Bayes	0.8254	0.8331	0.8476

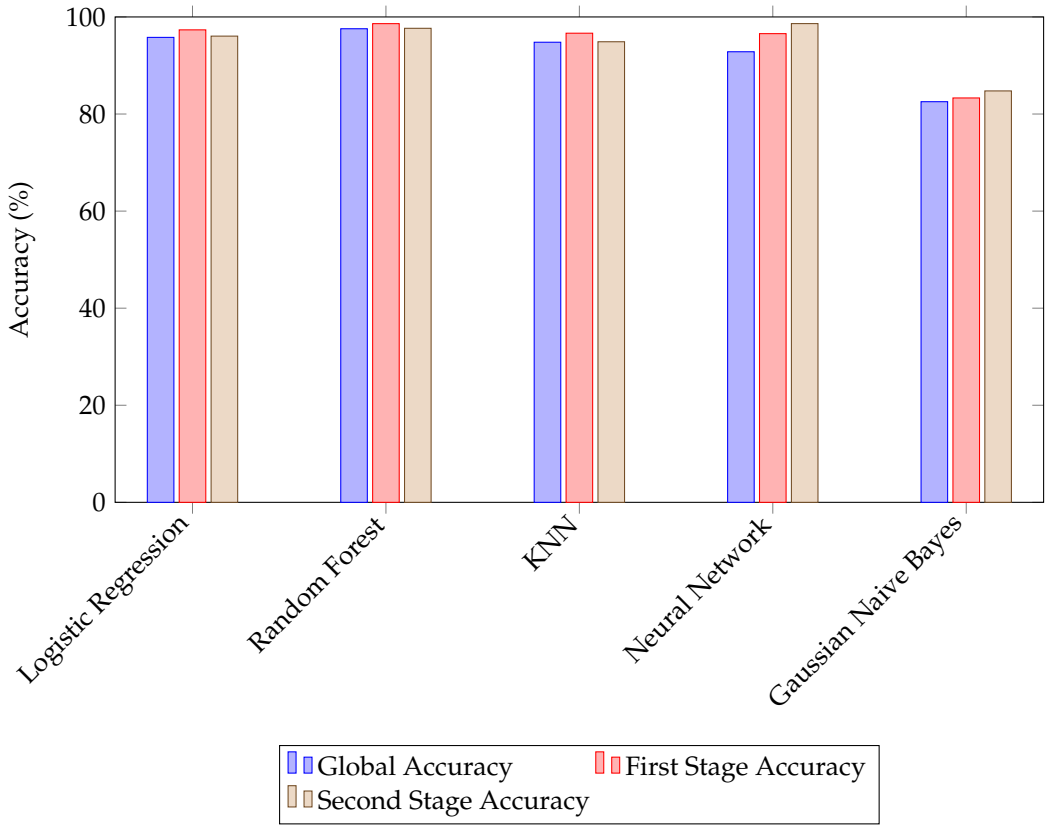

Figure 4. Comparison of accuracy results plotted for Multi-Stage Dataset (Global, First Stage, Second Stage).

5. Discussion

The results of the experiments conducted in the previous section are discussed below.

5.1. Bosch Dataset

As can be observed in Tables 1 and 2, except for the K nearest neighbors algorithm, the algorithms offer better results when the models are created with the data subsets, meaning the models achieve better accuracy when trained with the data from each production line separately. Nevertheless, for the K nearest neighbors algorithm, the largest loss in accuracy occurs in line 0 (L0) and amounts to 0.23%, which could be considered negligible. On the other hand, it is worth noting that with the Gaussian naive Bayes algorithm, the improvement in accuracy when training the models with the smaller datasets is significant, increasing from 61.82% accuracy with the complete model to 98.30% accuracy with the dataset from line 1.

5.2. Vehicle Painting Dataset

Table 6 shows a summary of the results of the experiment conducted with this dataset. As can be observed, the accuracy of the models when generated separately is higher than when a complete model containing all the data is generated, similar to what occurred in the experimentation with the previous dataset.

5.3. Condition Monitoring of Hydraulic Systems (ZEMA) Dataset

Regarding the ZEMA dataset [19], as shown in Tables 7 and 8, random forest achieved the highest global accuracy of 96.98%, performing perfectly on the cooler and valve subprocesses, while logistic regression followed closely with a global accuracy of 95.02%. Furthermore, the performance of the random forest algorithm improves significantly when the dataset is divided into the individual subprocesses. It achieves perfect accuracy not only in the cooler and valve subprocesses but also in the pump subprocess, with 100% accuracy. Additionally, the accumulator subprocess accuracy reaches 97.43%, showcasing an overall improvement across all subprocesses when compared to the global results. This indicates that random forest handles the specific characteristics of each subprocess more effectively, leading to better performance when each component is treated independently.

Similarly, both K nearest neighbors (KNN) and neural networks show significant improvements when applied to individual subprocesses. Although their global accuracies are relatively lower (77.19% for KNN and 74.77% for neural networks), both algorithms perform remarkably well in the cooler and pump subprocesses, with KNN achieving 99.55% and 96.07% and neural networks reaching 99.85% and 86.40%, respectively. These improvements highlight the ability of both methods to perform better on specific tasks within the dataset, compensating for their weaker performance in the overall model.

5.4. Multi-Stage Dataset

As shown in Table 9, random forest achieved the highest global accuracy of 97.57%, with notable improvements when applied to individual stages, reaching 98.62% in the first stage and 97.66% in the second stage. logistic regression also performed well, with a global accuracy of 95.79%, improving to 97.34% in the first stage and 96.05% in the second stage. Both models show a clear advantage when the dataset is divided, achieving better accuracy in each stage compared to the global result.

K nearest neighbors (KNN) and neural networks, while having slightly lower global accuracies (94.79% and 92.83%, respectively) also demonstrated significant improvements. KNN reached 96.65% accuracy in the first stage and 94.89% in the second stage, showing better performance when focusing on specific stages. Neural networks, despite their lower global performance, excelled in the second stage, achieving the highest accuracy among all models at 98.63%, though their first stage performance remained at 95.65%, still an improvement over the global score.

Finally, Gaussian naive Bayes, although the weakest model with a global accuracy of 82.54%, showed slight improvements when divided into stages, with accuracies of 83.31% and 84.76% in the first and second stages, respectively. Despite these gains, its performance lags behind the other models.

Naive Bayes underperformed due to its strong independence assumption, which is unsuitable for the complex feature interactions present in the datasets. This limitation makes it less capable of capturing the dependencies and correlations between features that are critical for accurate predictions in these datasets.

In contrast, random forest consistently outperformed other models due to its ability to handle high-dimensional data and its robustness to overfitting in smaller subsets. Its

ensemble-based approach allows it to leverage multiple decision trees, capturing feature interactions and providing a more flexible and accurate model even in complex scenarios.

The purpose of the edge computing models developed in this study is to enable real-time process monitoring, anomaly detection, and decision-making within industrial IoT-enabled environments. These models are specifically designed to operate at the data source, such as machines or production lines, reducing latency compared to traditional cloud computing architectures. In practical terms, edge computing models can be integrated into existing industrial systems such as supervisory control and data acquisition (SCADA) or manufacturing execution systems (MES) to enhance localized data processing and responsiveness. For example, in a vehicle painting process, real-time monitoring of temperature and humidity levels through edge models allows for immediate corrective actions, ensuring consistent quality and minimizing rework.

Furthermore, edge models can complement or replace centralized cloud solutions where low-latency responses are critical. These models are particularly applicable to environments where production lines generate large amounts of sensor data, as processing the information locally avoids network delays, ensuring faster feedback. By providing localized control and decision-making capabilities, edge computing empowers manufacturers to achieve greater operational efficiency and reliability, even in highly automated, IoT-connected production settings.

To better illustrate the deployment of edge computing models and their comparison with traditional cloud computing systems, we included a graphical representation of their integration into a typical industrial IT infrastructure. The right side of Figure 5 shows the flow of data from IoT-enabled machines and sensors to edge devices, where real-time processing occurs. This local processing significantly reduces latency and supports immediate decision-making.

In contrast, traditional cloud computing architecture, shown in the left side of Figure 5, involves transmitting data to centralized servers, which can introduce latency due to network transmission times.

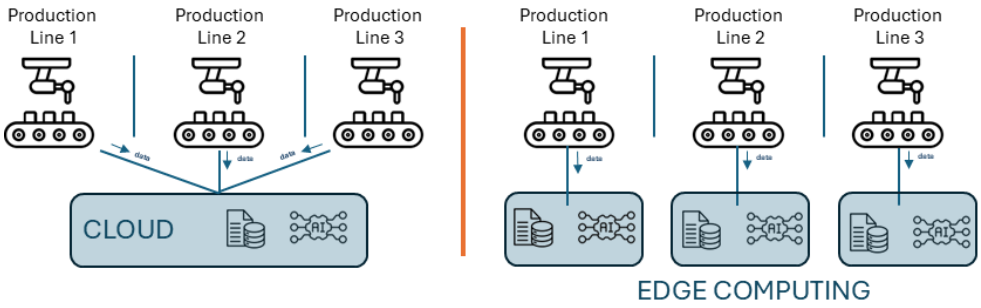

Figure 5. Cloud Computing vs. Edge Computing architecture.

The proposed framework leverages edge nodes to process data closer to their source, significantly reducing the dependence on centralized cloud infrastructure. By localizing data handling, the framework achieves lower latency, reduces transmission overhead, and ensures greater data privacy and security.

6. Conclusions

Upon completing experiments across four distinct industrial datasets related to production lines and analyzing the accuracy results, it is evident that the models generated using datasets from specific parts of the process achieve better results than large models that encompass multiple subprocesses. This pattern has been consistently demonstrated across all four industrial datasets—Bosch, vehicle painting, ZEMA, and multi-stage continuous

flow manufacturing—indicating that this approach works particularly well with datasets of this type. This paper's experiments demonstrated that models created for individual production lines achieved higher accuracy compared to those using data from all lines combined. For example, models for specific subprocesses such as enamel and primer coat reached accuracies of 78.10% and 67.01%, respectively, while the combined model of both subprocesses showed significantly lower performance, barely reaching 30%. Dividing the data into subprocesses related to distinct components also led to notable improvements in accuracy across all subprocesses, with random forest achieving perfect accuracy in several cases. This consistent pattern across all datasets highlights the advantages of focusing on individual subprocesses or stages to improve model performance. These findings reinforce the conclusion that having more data does not necessarily result in better model performance. This phenomenon occurs because data from one independent process can introduce 'noise' when predicting another process, as observed in the Bosch production lines and in the enameling and primer coat subprocesses from the vehicle painting dataset. The same effect was observed in the ZEMA and multi-stage datasets, where dividing data into individual subprocesses or stages consistently improved accuracy. This research further validates the primary objective of the study: to demonstrate the effectiveness of edge computing models in representing subprocesses within industrial settings compared to traditional cloud computing paradigms. With the rise of digitization and the Internet of Things (IoT), industries have increasingly sensored all processes, leading to the creation of large artificial intelligence models. However, this research demonstrates that for industrial processes, smaller, specialized models are more effective. By subdividing large datasets into smaller, more manageable sets and developing specialized edge models that operate near the source of data collection, latency is reduced, and both the speed of data processing and anomaly detection are enhanced. This approach provides a practical solution to the challenges posed by the immense volume of data in digitized industrial environments and offers a promising path towards more efficient and responsive industrial systems.

The framework was validated using real-world industrial datasets, such as those from vehicle painting processes and multi-stage manufacturing lines. By demonstrating its efficacy in real-time data processing and anomaly detection within these practical scenarios, the study establishes a direct connection between theoretical models and industrial applications.

The modular approach of developing specialized edge computing models for subprocesses, rather than comprehensive models for entire processes, provides a scalable and adaptable framework for real-time industrial systems. This modular design sets a benchmark for future research, particularly for applications requiring low-latency processing in Industry 4.0 environments.

Author Contributions: Conceptualization, N.G.L. and B.S.U.; methodology, N.G.L. and J.G.B.; software, N.G.L. and I.P.L.; validation, N.G.L. and P.G.B.; formal analysis, N.G.L.; investigation, N.G.L.; resources, N.G.L.; data curation, N.G.L.; writing—original draft preparation, N.G.L.; writing—review and editing, B.S.U., J.G.B. and I.P.L.; project administration, N.G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: In this research, experiments were conducted using four different datasets. The first experimentation was carried out with a dataset obtained from a production plant with which I was collaborating during my doctoral thesis. Upon conducting the experiment and observing the highly promising results, I decided to validate both the technique and the methodology with public datasets. Publishing an article with confidential data that I cannot share would not be particularly meaningful. Therefore, the data from the first research, referring to the vehicle

painting dataset is not readily available because the data are confidential to the vehicle manufacturing company. I have permission to share the results, but not the data. However, in order to share this knowledge with the scientific community, I have conducted the experiment with three other public datasets from the same sector. These public datasets are available in www.kaggle.com and in www.zenodo.org. These data were derived from the following resources available in the public domain: https://www.kaggle.com/c/bosch-production-line-performance/data, https://zenodo.org/records/1323611#.XfzEAEFCeUm, https://www.kaggle.com/datasets/supergus/multistage-continuousflow-manufacturing-process/data (access on 26 December 2024).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Premsankar, G.; Di Francesco, M.; Taleb, T. Edge computing for the Internet of Things: A case study. *IEEE Internet Things J.* **2018**, *5*, 1275–1284. [CrossRef]
2. Björkdahl, J. Strategies for digitalization in manufacturing firms. *Calif. Manag. Rev.* **2020**, *62*, 17–36. [CrossRef]
3. Stojmenovic, I. Fog computing: A cloud to the ground support for smart things and machine-to-machine networks. In Proceedings of the 2014 Australasian Telecommunication Networks and Applications Conference (ATNAC), Melbourne, Australia, 26–28 November 2014; pp. 117–122.
4. Oh, Y.; Park, H.; Yoo, A.; Kim, N.; Kim, Y.; Kim, D.; Choi, J.; Yoon, S.; Yang, H. A Product Quality Prediction Model Using Real-Time Process Monitoring in Manufacturing Supply Chain. *J. Korean Inst. Ind. Eng.* **2013**, *39*, 271–277. [CrossRef]
5. Zhu, Z. Research on the Application of Real-Time Monitoring System for Manufacturing Quality of Industrial Production Based on Industrial 4.0. In *Lecture Notes in Electrical Engineering*; Springer: Singapore, 2017. [CrossRef]
6. Jiao, L.; Friedman, R.; Fu, X.; Secci, S.; Smoreda, Z.; Tschofenig, H. Cloud-based computation offloading for mobile devices: State of the art, challenges and opportunities. In Proceedings of the 2013 Future Network & Mobile Summit, Lisboa, Portugal, 3–5 July 2013; pp. 1–11.
7. Barboni, L. Machine Learning and Edge Computing for Industry 4.0 Applications: Concepts and Extensive Review. In *Innovation and Competitiveness in Industry 4.0 Based on Intelligent Systems*; Springer: Cham, Switzerland, 2023; pp. 3–19.
8. Willner, A.; Gowtham, V. Toward a reference architecture model for industrial edge computing. *IEEE Commun. Stand. Mag.* **2020**, *4*, 42–48. [CrossRef]
9. Veith, A.; Assunção, M.; Lefèvre, L. *Latency-Aware Placement of Data Stream Analytics on Edge Computing*; Springer: Cham, Switzerland, 2018; pp. 215–229. [CrossRef]
10. Aral, A.; Brandić, I.; Uriarte, R.B.; Nicola, R.D.; Scoca, V. Addressing Application Latency Requirements through Edge Scheduling. *J. Grid Comput.* **2019**, *17*, 677–698. [CrossRef]
11. Zhang, Y.; Pang, C.; Yang, G. A Real-time Computation Task Reconfiguration Mechanism for Industrial Edge Computing. In Proceedings of the IECON 2020 the 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 18–21 October 2020; pp. 3799–3804. [CrossRef]
12. Bacchiani, L.; Palma, G.; Sciullo, L.; Bravetti, M.; Felice, M.D.; Gabbrielli, M.; Zavattaro, G.; Penna, R.D. Low-Latency Anomaly Detection on the Edge-Cloud Continuum for Industry 4.0 Applications: The SEAWALL Case Study. *IEEE Internet Things Mag.* **2022**, *5*, 32–37. [CrossRef]
13. Abouaomar, A.; Cherkaoui, S.; Mlika, Z.; Kobbane, A. Resource Provisioning in Edge Computing for Latency-Sensitive Applications. *IEEE Internet Things J.* **2021**, *8*, 11088–11099. [CrossRef]
14. Jo, J.; Jeong, S.; Kang, P. Benchmarking gpu-accelerated edge devices. In Proceedings of the 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), Busan, Republic of Korea, 19–22 February 2020; pp. 117–120.
15. Hussain, R.F.; Salehi, M.A. Resource allocation of industry 4.0 micro-service applications across serverless fog federation. *Future Gener. Comput. Syst.* **2024**, *154*, 479–490. [CrossRef]
16. Anagnostopoulos, C.; Mylonas, G.; Fournaris, A.P.; Koulamas, C. A Design Approach and Prototype Implementation for Factory Monitoring Based on Virtual and Augmented Reality at the Edge of Industry 4.0. In Proceedings of the 2023 IEEE 21st International Conference on Industrial Informatics (INDIN), Lemgo, Germany, 17–20 July 2023; pp. 1–8.
17. Törngren, M.; Thompson, H.; Herzog, E.; Inam, R.; Gross, J.; Dán, G. Industrial edge-based cyber-physical systems-application needs and concerns for realization. In Proceedings of the 2021 IEEE/ACM Symposium on Edge Computing (SEC), San Jose, CA, USA, 14–17 December 2021; pp. 409–415.
18. Risdal, M.; Prasanth, R.s.S.W.; Cukierski, W. *Bosch Production Line Performance*; Kaggle: San Francisco, CA, USA, 2016.
19. Schneider, T.; Klein, S.; Bastuck, M. *Condition Monitoring of Hydraulic Systems Data Set at ZeMA*; Zenodo: Genève, Switzerland, 2018.

20. Kaggle. *Multi-Stage Continuous-Flow Manufacturing Process*; Kaggle: San Francisco, CA, USA, 2018.
21. Vani, S.; Rao, T.M. An experimental approach towards the performance assessment of various optimizers on convolutional neural network. In Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 23–25 April 2019; pp. 331–336.