RESEARCH ARTICLE OPEN ACCESS

Optimization of Processes Through Machine Learning and Data Acquisition Based on IOT

[1]Vennila P, [2]Maniraj V
[1]Research Scholar, [2]Associate Professor
Department of Computer Science, A.V.V.M Sri Pushpam College, Poondi, Thanjavur.
Email: pj.vennila@gmail.com

Abstract:
As industrial data has expanded, monitoring systems have become crucial for management decision-making. Contemporary technologies, like sensors derived from the IoT, might aid in monitoring the production process. This paper suggests a system for ongoing monitoring that makes use of IoT sensors, big data processing techniques, and a hybrid predictive model. A sensor for the Internet of Things was created to gauge temperature, humidity, accelerometer data, and gyroscope information. The sensor data generated by IoT in factories is real-time, massive in volume, and unstructured. Apache Kafka is used for alerts, Apache Storm for real-time analysis, and MongoDB for storing factory sensor data on the large-scale data analysis platform. Secondly, the hybrid prediction model combined outlier identification based on DBSCAN with categorization using Random Forest to eliminate outlier sensor data and identify manufacturing defects. The vehicle was tested on a Korean car production line. The findings demonstrated that an efficient monitoring of production can be achieved through the use of IoT-based sensors in conjunction with the proposed big data processing framework. When it comes to fault prediction, the hybrid model that makes use of sensor data provides greater accuracy than alternative models. The proposed system is anticipated to improve managerial decision-making and reduce losses linked to manufacturing errors.

Keywords: **Monitoring framework; IoT-based sensor; large-scale data processing; fault identification; DBSCAN; Random Forest**

1. INTRODUCTION:

Manufacturing plays a crucial role in economic development and is regarded as essential for economic growth in the era of globalization [1]. It positively affects the growth of both developed and developing countries. To enhance the economic competitiveness of individual manufacturers and foster sustainability throughout the entire industrial sector, the manufacturing sector employs emerging technologies [2]. The integration of information and communication technology (ICT) into manufacturing enables a transition from traditional to advanced manufacturing processes [3]. In the context of utilizing ICT, monitoring systems are essential for overseeing and handling manufacturing processes. The latest developments in information technology allow for the integration of various monitoring applications into one comprehensive system that covers the whole supply chain. Generally, it is essential to use a monitoring system for forecasting diseases, increasing production, lowering costs, and creating an early warning mechanism. Modern technologies, such as sensors based on the Internet of Things (IoT), can be used to improve and integrate monitoring systems. Studies in the manufacturing industry have shown that IoT-based sensors for monitoring offer significant benefits. These advantages encompass improvements to working conditions, prevention of design mistakes [4-6], fault diagnosis, quality forecasting, and aiding managers in making better decisions. It is anticipated that the expanding accessibility of IoT sensing devices will result in a data explosion from the manufacturing sector (including process logs, events, images, and sensor data). Such data is called “big data”. The examination of big data has led to significant progress in the manufacturing sector [7], such as decreased energy consumption, improvements in production scheduling and logistics planning, mitigation of social risks, and encouragement of better decision-making. Previous studies have shown significant benefits from different big data technologies for the quick processing and storage of large amounts of data, including Apache Kafka, Apache Storm [8], and NoSQL MongoDB.

Prior studies showed significant advantages from the integration of big data technologies, such as decreased processing times for home automation systems, effective and efficient solutions for handling IoT-generated data in smart cities, and real-time management of large amounts of smart environmental data [9-11]. Data processing systems have integrated the aforementioned big data technologies, resulting in considerable advantages from the effective processing of vast amounts of streaming spatiotemporal data and large quantities of manufacturing sensor data. In order to process, store, and present substantial volumes of streaming sensor data from manufacturing in real time, it is crucial to integrate Apache Kafka, Apache Storm, and MongoDB into big data processing systems used in the manufacturing sector.

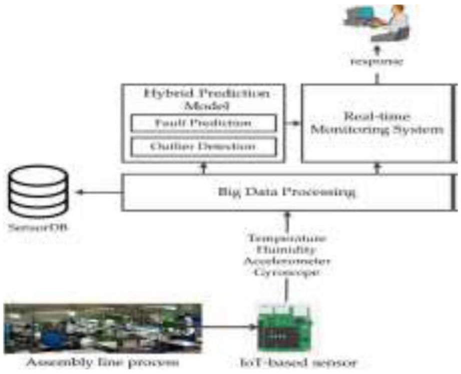
It is necessary to analyze data produced by the manufacturing sector in order to assist managers with their decision-making [12]. Methods of machine learning can be regarded as advanced technology with significant promise for data analysis, having been effectively utilized in a range of fields including fault detection, quality prediction, defect classification, and visual inspection [13]. Machine learning algorithms like Random Forest are very efficient at identifying irregular occurrences in a process when it comes to fault prediction, making them helpful in preventing productivity decline. Nonetheless, machine learning algorithms face challenges when dealing with outlier data [14], which can lead to a decrease in the classification model's accuracy. Outlier detection can be utilized to identify and eliminate outliers, thereby enhancing the performance of classification models. A method employed for outlier detection is Density-Based Spatial Clustering of Applications with Noise (DBSCAN). DBSCAN has been applied across various domains and has proven effective in identifying true outliers [15]. To achieve a more precise identification of irregular occurrences in the course of manufacturing, it is essential to combine Random Forest with outlier detection based on DBSCAN. The outcomes of the studies mentioned above have demonstrated considerable benefits of IoT-based sensors, big data technology, and machine learning models in enhancing management decision-making. Nonetheless, no research has been conducted on combining IoT-based sensors, big data technology, and machine learning models into a comprehensive monitoring system tailored for automotive manufacturing. Therefore, we put forth a real-time monitoring system for the automotive sector that incorporates IoT-based sensors, big data processing, and a hybrid prediction model. The suggested IoT-based sensor gathers temperature, humidity, accelerometer, and gyroscope data from the assembly line process, while a big data processing platform manages and stores the substantial volumes of sensor data it produces. Ultimately, the suggested hybrid prediction model—comprising DBSCAN-based outlier detection and Random Forest classification—is employed to eliminate outlier data and facilitate fault detection in the manufacturing process, respectively.

This paper is structured in the following way. Section 2 provides an explanation of the methodology, and Section 3 presents the results along with discussions. In Section 4, concluding remarks are presented and several limitations and remaining challenges are discussed.

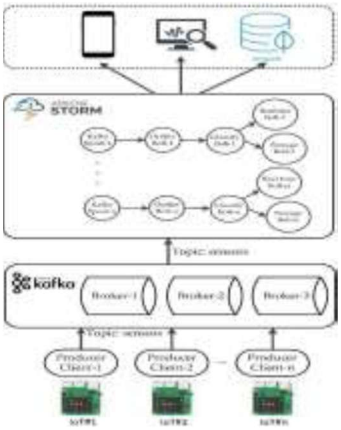
2. METHODOLOGY

2.1. System Design

The proposed real-time monitoring system was designed to assist managers in more effectively overseeing the assembly line process in automotive manufacturing and to offer early warnings upon fault detection. The suggested system employs IoT-based sensors, big data analytics, and a hybrid forecasting model. The hybrid prediction model comprises a clustering-based outlier detection method and a machine learning-based classification model. Figure 1a shows that sensors based on IoT technology are affixed to the workstation desk along the assembly line. The sensors based on IoT technology include temperature, humidity, accelerometer, and gyroscope sensors. Sensor data generated by the IoT is sent wirelessly to a cloud server that hosts the big data processing system. The system enables the rapid processing of large volumes of sensor data prior to their storage in the MongoDB database. A clustering-based approach is employed to eliminate outliers from the sensor data. Furthermore, a classification model based on data analytics and machine learning is utilized to forecast faults based on the current sensor data in the assembly line process. Finally, the manager receives a real-time presentation of the complete history of sensor data, including temperature, humidity, accelerometer, and gyroscope readings, along with the fault prediction results, through a web-based monitoring system.



(a)



(b)

Figure 1. Architecture of the real-time monitoring system in an assembly line process (a) and system design for big data processing (b).

The suggested system for processing big data makes use of Apache Kafka, Apache Storm, and MongoDB. Apache Kafka is a message queue system that features low latency, high throughput, and fault tolerance, and can publish data streams. Apache Storm is a real-time, parallel data processing system that offers horizontal scalability, fault tolerance, and guaranteed data processing. It can handle large volumes of high-velocity data streams. The proposed system design for the big data processing system intended for real-time monitoring is illustrated in Figure 1b. Sensor data from the IoT-based sensor device is transmitted wirelessly using a Python-based program designed to function as the “producer” for the Kafka server. The client referred to as the “producer” publishes data streams to Kafka “topics” that are distributed across one or more cluster nodes/servers known as “brokers”. Storm processes the published data streams from Kafka in real-time and in parallel. Storm contains implementations for outlier detection and classification. Real-time presentation of sensor data and classification results occurs in a web-based monitoring system, with storage in MongoDB.

2.2. System Implementation

The IoT-based sensor developed comprises a Raspberry Pi as the sole mainboard and a Sense-HAT as an additional sensor board. The Raspberry Pi is a compact single-board computer measuring 85.60 mm × 53.98 mm × 17 mm and weighing just 45 g. It is budget-friendly, with a cost of around $25–35 USD. It features USB, LAN, HDMI, audio, and video ports for different input and output tasks. Moreover, general-purpose input-output (GPIO) connectors allow for the connection of extra devices or add-on boards like sensors to the main board. We created a Python-based client program in this study that utilizes the provided official application programming interface (API) to collect sensor data from IoT-based sensors. The IoT-based sensors continuously collect temperature, humidity, gyroscope, and Accelerometer data, which are transmitted to a cloud server wirelessly.

2.3. Hybrid Prediction Model for Fault Detection

The hybrid prediction model is employed in this study to determine if the process is functioning normally or abnormally. The procedure for identifying normal and abnormal occurrences in the course of manufacturing is illustrated in Figure 2. The hybrid prediction model employs an outlier detection method based on DBSCAN to identify and eliminate outliers from the sensor data, along with a Random Forest-based classification model to predict normal and abnormal events. Lastly, the evaluation of the performance involves a comparison between the hybrid prediction model and other classification models.

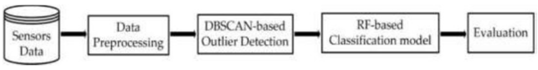


Figure 2. Hybrid Prediction Model using Density-Based Spatial Clustering of Applications with Noise (DBSCAN)-based outlier detection and Random Forest (RF)- based classifier.

3. RESULTS AND DISCUSSIONS

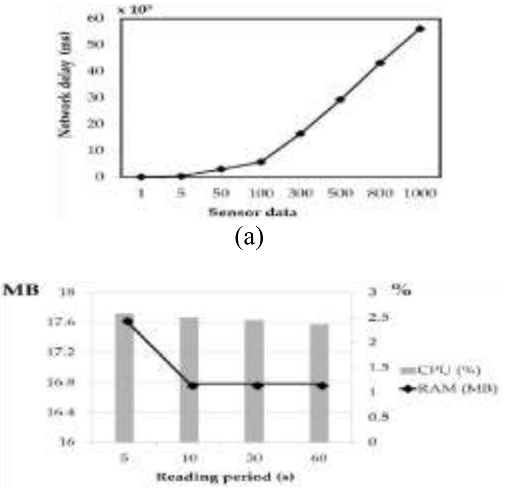
3.1. Real-Time Monitoring System

Data visualization was created using a JavaScript framework to serve as a monitoring system that displays sensor data in real-time. The proposed system would enable the manager to oversee the status of the assembly line process and receive early warnings in real time when an abnormal event (fault) is detected. The sensor data collected by IoT-based sensor devices is transmitted to Apache Kafka. Subsequently, Apache Storm processes this data and sends it along with its fault prediction results to the monitoring system in real-time. Ultimately, both the sensor data and the prediction results are stored in MongoDB.

3.2. Performance of the IoT-Based Sensor

A sensor based on IoT technology comprises a sensing device and a client application that collects sensor data and transmits it to a cloud server. Analyzing the performance of IoT-based sensors under different conditions is crucial. This study employed performance metrics like network delay, as well as CPU and memory usage. The sensor device performance was evaluated by Alazzawi and Elkateeb using network delay as a metric, whereas Morón et al. used CPU usage to assess IoT device capabilities in various scenarios. In our investigation, we characterized network delay as the mean duration between transmitting sensor data from the source (sensor device) and achieving a successful receipt of that data at the destination (MongoDB). The second performance metric involved assessing the average CPU and memory utilization of the client program across different scenarios.

The client program utilized in this study was a Python-based application operating on an IoT sensor device, which gathered sensor data including temperature, humidity, gyroscope, and accelerometer readings. For the experiment, an IoT-based sensor running on Linux Raspbian OS Jessie with 1 GB RAM was used. Wi-Fi was utilized to implement communication between the cloud server and the IoT-based sensor. As depicted in Figure 3a, the network delay varies with different quantities of sensor data. The findings indicate that the network delay grows with the increase in sensor data transmitted by the sensor device. The IoT-based sensor requires about 50 seconds to simultaneously transmit 1000 sensor data points. In actual implementations, sending the sensor data takes less than 0.02 seconds, as we configure it to send only one sensor data point (temperature, humidity, gyroscope, or accelerometer) every 5 seconds. Figure 3b also illustrates the client program's CPU and memory utilization. Four distinct scenarios for the reading period were assessed, where the client program read and transmitted sensor data to the cloud server at intervals of 5, 10, 30, and 60 seconds. The findings indicated that the reading period has a negligible impact on CPU or memory usage. Regarding the computational cost of the client program, it should be noted that the program used less than 3% CPU and 18 MB for all reading periods.



(a)

(b)

Figure 3. The IoT-based sensor system’s (a) network delay, and (b) CPU and memory usage.

3.3. The Performance of Big Data Processing

Assessing how well big data processing performs in different circumstances is crucial. This study employed performance metrics including system latency, throughput, and concurrency. Pereira et al. assessed the performance of big data technology during various operations by measuring system latency and throughput, whereas Van der Veen et al. evaluated it under multiple clients using concurrency. In our research, system latency refers to the duration required by the proposed system to manage, process, and save the sensor data in the database. Throughput refers to the total amount of sensor data processed every second. The final metric is concurrency, which refers to the number of clients that accessed the system at the same time. Various server quantities were used to carry out the experiments, and the response time was gathered for analysis. The Java program was created to serve as a simulator that produces sensor data and transmits it to the big data processing servers. Apache Kafka, Apache Storm, and MongoDB were installed on the server. The threads were employed by a Java program to emulate multiple clients. Moreover, each simulated data item is approximately 211 bytes in size and includes the device ID, the date and time of data generation, and the sensor data value.

3.4. Hybrid Prediction Model for Fault Detection

The IoT-based sensor device transmits sensor data to the big data processing system, which saves it in NoSQL MongoDB during dataset generation. The sensor, which is based on IoT technology, gathers data from various kinds of operations, encompassing both normal and abnormal occurrences. Expert users label the dataset according to the process status (normal or abnormal) during the time the sensor data was collected. Subsequently, the dataset is examined with the hybrid prediction model to forecast the fault status. Table 1 shows the results of comparing the performance of various classification models. A comparison was made between the hybrid prediction model and several conventional classification models, including Naïve Bayes (NB), Logistic Regression (LR), Multilayer Perceptron (MLP), and Random Forest (RF), for the purpose of identifying and predicting abnormal events. Compared to other classification models, the proposed model achieved the highest accuracy (100%). After implementing outlier detection based on DBSCAN, there was a minor enhancement in model accuracy. The accuracy of the conventional Random Forest was improved by up to 1.462% through the combination of Random Forest and outlier detection based on DBSCAN. Moreover, after using DBSCAN for outlier detection, accuracy enhancements of up to 3.173%, 0.567%, and 2.026% have been observed in other conventional classification models such as Naïve Bayes, Logistic Regression, and Multilayer Perceptron, respectively. The suggested model was realized in Apache Storm, allowing for parallel and real-time processing and prediction of data streams from Kafka.

Table 1. Performance comparison of several classification models for fault prediction.

Model	Precision (%)	Recall (%)	Accuracy (%)
Naïve Bayes (NB)	94.1	93.6	93.567
Logistics Regression (LR)	98	98	97.953
Multilayer Perceptron (MLP)	96.8	96.8	96.784
Random Forest (RF)	98.5|	98.5	98.538
DBSCAN + NB	96.8	96.7	96.74
DBSCAN + LR	98.6	98.5	98.52
DBSCAN + MLP	98.8	98.8	98.81
Hybrid Prediction Model (DBSCAN + RF)	100	100	100

3.5. Managerial Implications

The proposed system in this study is composed of three components: the IoT-based sensor, big data processing, and a machine learning model. To begin with, the IoT-based sensor device created in this research utilizes a Raspberry Pi, which is a compact, affordable, and high-performance single-board computer. Prior research has demonstrated considerable benefits of using Raspberry Pi for various applications, including controlling and overseeing IoT systems, real-time estimation of a vehicle's roll angle through embedded neural networks, hosting and providing the user interface for eHealth care systems, and monitoring lava lake temperatures with near-infrared thermal cameras. Hence, the IoT-based sensor device proposed and developed in this research could serve to monitor the manufacturing process as it happens. Second, due to the rise in the number of IoT devices, there is a need to create new big data processing methods to manage, process, and store the data effectively without incurring noticeable performance degradation. Prior research indicated that organizations can realize economic benefits related to software development productivity, product quality, and reduced costs (such as licensing fees) and external support availability by employing open source software (OSS). In our research, we developed a big data processing platform that utilizes OSS, which is a cost-effective option for implementation and integration. Third, various processes in manufacturing and predictive maintenance across different industries have utilized machine learning for monitoring systems. Machine learning offers potent tools for ongoing enhancement of quality in a process as intricate and extensive as semiconductor production. In our research, the machine learning model serves to identify faults (anomalous events) occurring in real-time during the assembly line process.

Consequently, it is anticipated to assist management in enhancing decision-making and averting unforeseen losses due to faults that occur early in the manufacturing process. Ultimately, the study's overall findings can serve as a reference for industrial practitioners looking to incorporate IoT, big data, and machine learning into their manufacturing processes. Several aspects of big data have been explored by earlier academics and practitioners. Big data is often described in terms of 4 V's, they are volume, variety, velocity and veracity. Nonetheless, certain academics concentrate on one or several facets of the big data concept. Davenport et al. concentrated more on the variety aspect of data sources, whereas other authors highlighted the storage (volume) and analysis components in relation to big data. Our study has developed a processing system for big data that can efficiently manage the rapid influx (velocity) and vast quantities (volume) of sensor data. Ultimately, the combination of an IoT-based sensor, big data analytics, and a machine learning model can be employed to efficiently oversee the manufacturing process and provide early warning alerts upon real-time detection of anomalies.

4. Conclusions

This research involved the creation of a real-time monitoring system that utilizes IoT-based sensors, big data processing with tools such as Apache Kafka, Apache Storm, and MongoDB, and a hybrid predictive model to detect manufacturing faults. The system's effective real-time processing of large-scale sensor data ensures scalability and low computing costs. Utilizing Random Forest for fault identification and DBSCAN for outlier detection, the accuracy of fault detection is exceptional when compared to other models. This method results in enhanced monitoring of the assembly line, a reduced likelihood of unexpected losses, and improved assistance for management in making decisions. Future research directions include enhancing IoT security and expanding the dataset to incorporate more complex failure scenarios.

Reference:

1) Syafrudin, Muhammad, et al. "Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing." Sensors 18.9 (2018): 2946.
2) Abbas, Farwa, et al. "Use of Big Data in IoT-Enabled Robotics Manufacturing for Process Optimization." Journal of Computing & Biomedical Informatics 7.01 (2024): 239-248.
3) Irshad, Omer, et al. "Performance optimization of IoT based biological systems using deep learning." Computer Communications 155 (2020): 24-31.
4) Bhaskaran, Priyanka E., et al. "IoT Based monitoring and control of fluid transportation using machine learning." Computers & Electrical Engineering 89 (2021): 106899.
5) Tran, Minh-Quang, et al. "Machine learning and IoT-based approach for tool condition monitoring: A review and future prospects." Measurement 207 (2023):112351.
6) Tang, Lui Sieng, et al. "A Low-Cost IoT-Based System for Manufacturing Process Data Acquisition." 2020 13th International UNIMAS Engineering Conference (EnCon). IEEE, 2020.
7) Shurrab, Mohammed, et al. "Iot sensor selection for target localization: A reinforcement learning based approach." Ad Hoc Networks 134 (2022): 102927.
8) Algabroun, Hatem, and Lars Håkansson. "Parametric Machine Learning-Based Adaptive Sampling Algorithm for Efficient IoT Data Collection in Environmental Monitoring." Journal of Network and Systems Management 33.1 (2025): 5.
9) Elkateb, Sherien, et al. "Machine learning and IoT–Based predictive maintenance approach for industrial applications." Alexandria Engineering Journal 88 (2024): 298- 309.
10) Amodu, Oluwatosin Ahmed, et al. "Deep Reinforcement Learning for AoI minimization in UAV-aided data collection for WSN and IoT: A survey." IEEE Access (2024).