Research Article

A Process Of Developing An Internet Of Things Based Model For Manufacture Monitoring In Automobile Industry

Gokul . M[1], Balamurali . M[2]

[1]Assistant Professor, Department of Computer Science and Engineering, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai- 600062.
[2]Assistant Professor, Department of Mechanical Engineering, Thiagarajar college of Engineering, Madurai – 625015.
gokulknocks@gmail.com

Abstract: In this modern era, monitoring various aspects like temperature, pressure, equipment status, working environment, inventory, and productivity is complex. The deviation in the values due to inaccurate measurement will have a significant effect on productivity. To overcome this problem,the Internet of Things-based sensor system plays an essential role in effectively monitoring the manufacturing systems. In this research, a real-time monitoring system using Internet of things based sensors for big-data processing and prediction has been proposed for applications in theautomobile industries. Initially, the IoT-based sensor collects important parametric data. The obtained big-data is processed using Apache Kafka, which is a type of message quence. For further real-time processing, and Apache storm engine is proposed. MongoDB is used to store theinformation from the sensor data collected during the manufacturing process. DBSCAN [Density-Based Spatial Clustering of Applications] and Random forest-based noise-based outlier detection are used to remove the outlier information from the sensor data. It points out the fault detection during the manufacturing process. The model's outcome proves that the Internet of Things-based sensors for processing big data seems to be a successful process during product manufacturing. From the outcome, it was found that the proposed model is having 100% more accuracy than the other existing model. The study's outcome proves that it may help the automobile management regarding unexpected loss due to fault in the initial stage of manufacturing, and it also improves the decision-making for the manufacturing process.

Keywords: fault detection, big data processing, IoT, monitoring system, automobile industry

1. Introduction

Due to the competition and technological advancement in recent times, manufacturing should cope with the speed while being intelligent and responsive by means of quality improvement and operation. For smart manufacturing, just-in-time manufacturing has been widely used for increasing product productivity. The manufacturing industry faces many challenges imposed by development policies, other opponent manufacturers, green manufacturing, etc. They are also facing constraints in material wastage, so re-manufacturing is the only option that may help them reuse the materials, resource reduction, low production costs, energy consumption, and alleviate environmental pollution. Re-manufacturing is considered an effective way towards cleaner production and sustainable production. It also impacts enhancing energy efficiency and tries to reduce environmental constraints. Re-manufacturing begins with already parts, which is done through the steps as follows:

1. Partial or complete disassembly of products
2. Cleaning the used products
3. Inspecting and storage of parts
4. Rechecking and replacing of certain spare parts
5. Reassembly of products

In the final step, a brand new product will be produced. Remanufacturing also belongs to smart manufacturing, and here the products may be produced with better quality and performance. The re-manufacturing industry is expected to attain €70bn and €100bn by the end of 2030 with the employment of almost 430,000 to 600,000 employers. Outlier detection can be applied to identify and remove outliers, thus improving classification models' performance [2]. DBSCAN [Density-Based Spatial Clustering of Applications with Noise] is considered the best technique for outlier detection [3]. This process has been implemented in many fields and is regarded as the most successful one for detecting outliers [4]. Along with DBSCAN, Random Forest [RF] should join hand-in-hand to detect outliers and detect abnormalities in the manufacturing process accurately. In this research, the Internet of Things-based sensors is used to collect the humidity, temperature, accelerometer, and gyroscope data from the initial assembling process and the big data processing used for handling the platform and storing large generated data structure. The hybrid prediction model DBSCAN and RF is used for outlier detection and classification; this model helps in enhancing fault detection and provides information during the manufacturing process itself.

2. Literature Survey

2.1 Manufacturing planning and scheduling

Many research has been conducting regarding manufacturing planning and scheduling. [5] proposed a situation-based optimization approach for real-time analysis of a shop with batch-wise processing machines. The proposed model is used for identifying the uncertainty with processing and job size. [6] proposed a framework for real-time manufacturing considering the cost, risk of stakeholders, energy consumption, etc. here four different kinds of scheduling model such as the central, consumer-centered, operator-centered, and system-centered model was analyzed. When compared to the other scheduling model, the system-centered model attained high utility and efficiency, demand for manufacturing, service capability, etc. [7] proposed a planning system with the integration of process planning, design, and shop floor scheduling, etc. the proposed model was applied on multiple agent system. [8] proposed an international model for manufacturing and scheduling for a multi-centric system. The proposed model was based on the availability, quantity, and collaborative cost of the manufacturing process.

[9,10] The quantity of IoT-based sensors and other related parts is expanding essentially. The selection of IoT in assembling empowers the change from customary to present-day digitalized fabricating. As the number of gadgets gathering sensor information in assembling builds, the potential for new sorts of utilizations then dealt with the contribution of a lot of sensor information, such as enormous information innovation. It fostered a theoretical structure by coordinating massive information innovation in IoT, which is relied upon to help essential dynamics. By using huge information preparation, the gigantic measure of information gathered by numerous heterogeneous sources (sensor gadgets) can be taken care of and introduced in a proficient way. In this way, they can help supervisors with better dynamics.

2.2 Application of IoT in manufacturing and scheduling

The assembling process is characterized as 'the way toward allotting fabricating assets over the long haul to the arrangement of assembling measures in the process plan'. In examination with the booking interaction in customary assembling framework that is surveyed by some standard single presentation measures or bio measures, like expense, energy, and profitability energy,[11] the JIT fabricating is stressed on JIT creation and conveyance, which implies both earliness and lateness are debilitate on the grounds that the earliness will expand the stock expense and prompts excessive waste in the creation cycle while the lateness will make late conveyance that disappoints clients or in some cases even conceivably abuses contracts.

[12] proposed a framework using the internet of things based supply chain. The proposed model seems to be an advantage for interoperability, strong visibility, and practicality. [13] proposed a framework using RFID [Radio-Frequency Identification] for real-time monitoring of production where the RFID system is integrated with the IT [Information Technology] system and manufacturing process. [14] proposed an RFID-based SS [Support System] for monitoring the status of the manufacturing process in industries. [15] proposed a hybrid model with IoT and RFID for controlling and monitoring production in a manufacturing industry.

2.3 Big Data Processing using IoT in the Automobile industry

Industrialization 4.0 is the combination of vertical and horizontal integration. Here the Vertical integration is the transformation of the existing system to a new one. Horizontal integration is the link between communication and information for production and service [15]. Industrialization 4.0 is possible only with the help of the past 3 revolutions. 3rd revolution is the key where IT and technology have been widely used. The present revolution is due to the Internet of Things (IoT), cyber-physical systems (CPS), Internet of Energy (IoE), Internet of People (IoP).

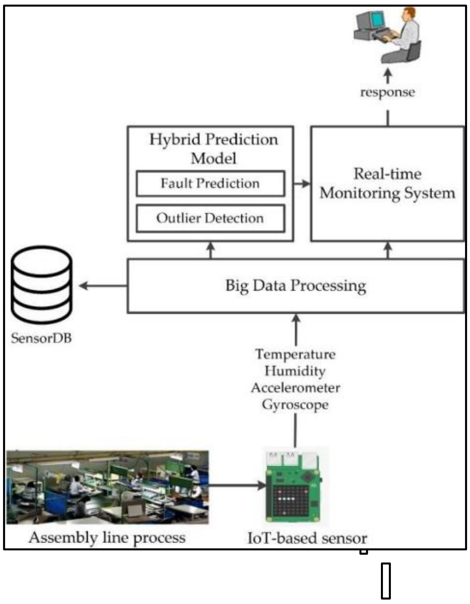
2.4 Sheet Metal Forming [SMF] in Automobile Industry

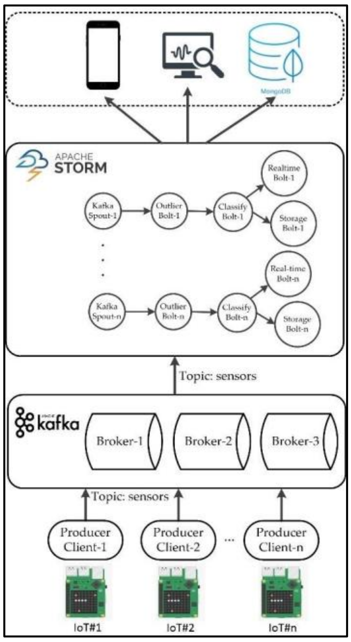
SMF is the process of converting a metal sheet to the desired shape [16]. During SMF, a metal sheet commonly known as blank will be placed inbetween the die and a holder where pressure is needed to hold the blank. After punching, the desired shape from the blank can be attained. Few drawbacks of SMF are variation in temperature, elastic deformation of dies, and varying tribology conditions [17,18]. Some of the products obtained after the SMF process will lack quality and face surface defects, necking, fractures, surface defects, wrinkles, and geometric specifications due to excessive springback. All these defects increase the cost, lead to wastage of resources, production, etc. [19] proposed an SMF framework to examine the tribology and material impact for predicting the quality through the simulation process. The simulation model can predict and validate the accuracy—the mathematical model guides for controlling the algorithm for a real-time manufacturing system.

3. Methodology

3.1. System Design

A real-time monitoring system has been proposed to assist managers in monitoring the assembling process in the automobile manufacturing industry. It also helps in early fault detection. The proposed real-time monitoring system combines IoT sensors, big-data processors, hybrid DBSCAN and RF, for outlier detection and classification. Figure 1.a shows the IoT sensor, which is attached to the assembling workstation. The Internet of Things sensors consists of humidity, temperature, gyroscope, and accelerometer sensor. The gathered information from the IoT sensor will be sent to the cloud server for big data processing. Here the system tries to process a large amount of data quickly before storing them in the MongoDB database. After this process, the proposed hybrid model is used for detecting and classifying the outlier detection. The obtained data from this process will be sent to the manager as real-time data through a web-based monitoring system along with fault prediction outcomes.





(**a**) (**b**)

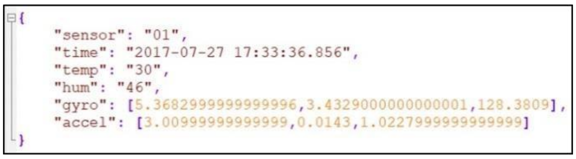
Figure 1. a) systematic set-up for real-time monitoring in assembling process, b) system design for big data processing

The big-data processing is done through Apache Storm, Kafka, and MongoDB. Figure 1 b) shows the real-time monitoring via big data processing system. The IoT sensor sends the data to the python programmed Kafka server. From here, the data will be published and then will be sent for processing by parallel storm. In the parallel storm, the outlier detection and classification take place. After implementing the test, the data will be stored in a MongoDB for presenting it as a web-based monitoring system for the managers.

The Internet of Things sensor data will be in the following characteristic features as follows:

1. Unstructured format
2. Large amount
3. Continuous generation

Figure 2 illustrates the data generated by the Internet of things sensor. The data is generated in JSON format. The obtained data will be sent to the parallel storm system for the proposed hybrid model. The data and the predicted outcome are stored in NoSQL MongoDB in order to improve the performance [16]. It was found that the proposed model is suitable for large sensor data; it is capable of handling fast read and write the version according to it [17]. In this research, an embedding scheme-based sensor data repository is used. Figure 2 b) illustrates the ID for the IoT device, processing time, recorded time, sensor data, and prediction outcome. The sensor data such as humidity, temperature, accelerometer data, and gyroscope are also shown in the subdocument.



{
"sensor": "01",
"time": "2017-07-27 17:33:36.856",
"temp": "30",
"hum": "46",
"gyro": [5.3682999999999996,3.4329000000000001,128.3809],
"accel": [3.00999999999999,0.0143,1.0227999999999999]
}

{
"_id":"5a845f42822934453781a87e",
"sensorNo":"01",
"regTime":"2017-07-27 17:33:36.856",
"processedAt":"2017-07-27 17:33:37.035",
"sensorsData":{
"temp":"30",
"hum":"46",
"gyro":{
"x":"5.3682999999999996",
"y":"3.4329000000000001",
"z":"128.3809"
},
"accel":{
"x":"3.009999999999999",
"y":"0.0143",
"z":"1.0227999999999999"
}
},
"predictionresult":"normal"
}

(**a**) (**b**)

Figure 2. a) illustrates the data generated by the Internet of things sensor, b) the ID for the IoT device stored in NoSQL MongoDB format.

3.2. System Implementation

This research monitoring system is used for monitoring the assembling process of the automobile industry in India, illustrated in figure 3. The proposed Internet of things sensor consists of a single mainboard Raspberry Pi [18] and an add-on sensor board Sense hat [19]. The raspberry pi is made of 85.60 mm × 53.98 mm × 17 mm in dimension with 45 g in weightage. It also has LAN, USB, audio, HDMI, video ports for input and output operations. The GPIO **[**general-purpose input-output] connectors are used for additional add-ons board connected with the mainboard [20]. Table 1 presents the details of the Raspberry Pi specification. The add-on sensor board Sense-hat is the one that measures the proposed parameters. Table 2 shows the detailed specification of the add-on sensor board Sense-HAT. The add-on board is connected with the mainboard through GPIO 40 pins. Figure 3 illustrates the implementation of a real-case version of Internet of Things sensors. [21] a python-based program, the API [application programming interface] is used for gathering the sensor data from the Internet of things sensors. The IoT- sensor collects the data of the proposed parameter and then transmits it to the cloud server via WSN [Wireless Network]. The IoT sensor can also sense the environmental condition and send the data to the cloud system every five seconds. The sensor data is processed using big data processing system, and further, the real-time model is analyzed. Later the sensor data is stored in MongoDB and represented on a cloud-based monitoring system.

Table 1. Raspberry Pi-3 Model specification details

tions	RAM	GPU	CPU	Storage	Wireless	Power Consumption	Ethernet	GPIO	Siz
tion	1GB	400MHz Videocore IV	Quad Cortex A53@1.2GHz	Micro-SD	Wireless LAN 802.11n/Bluetooth	5V	10/100Mbps	40 pins	85.6 x 56.5m

Table 2. Sense-HAT detailed specification

Terms	Input	Display	Magnetometer	Te Temp.	Barometric Pressure	Accelerometer
Data	S small 5 5 N joystick button	8 x 8 LED Display matrix	Magnetic Sensor (accurate to + 4/8/12/16 gauss)	T Temp sensor () (accurate to $+2^{0C}$)	P Pressure Sensor (accurate to + 2/4/8/16 G-	Accelerometer sensor (accurate to +245/500/2000 degrees per second)

					forces)	

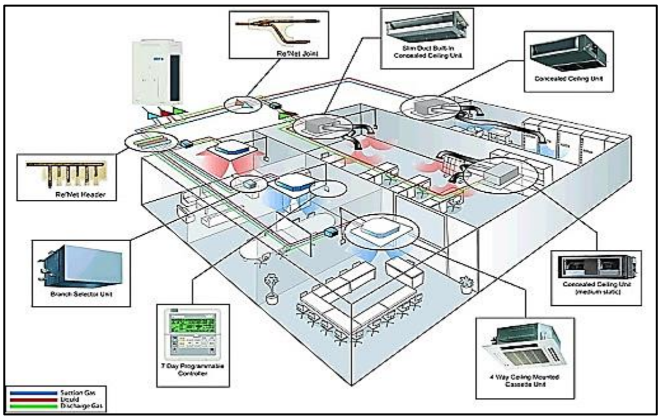


Figure 3. Implementation of a real-case version of Internet of Things sensors assembling process

3.3 Hybrid Prediction Model for Fault Detection

The hybrid model is used for predicting the outlier detection and abnormalities of the real-time system. Figure 4 illustrates the most frequently occurring normal and abnormal error during the default manufacturing process. The hybrid prediction is based on proposed DBSCAN for outlier detection, and RF-based classification is used for predicting the normal and abnormalities that occur in the outlier process. The proposed hybrid prediction model is compared with other classification processes for performance evaluation.

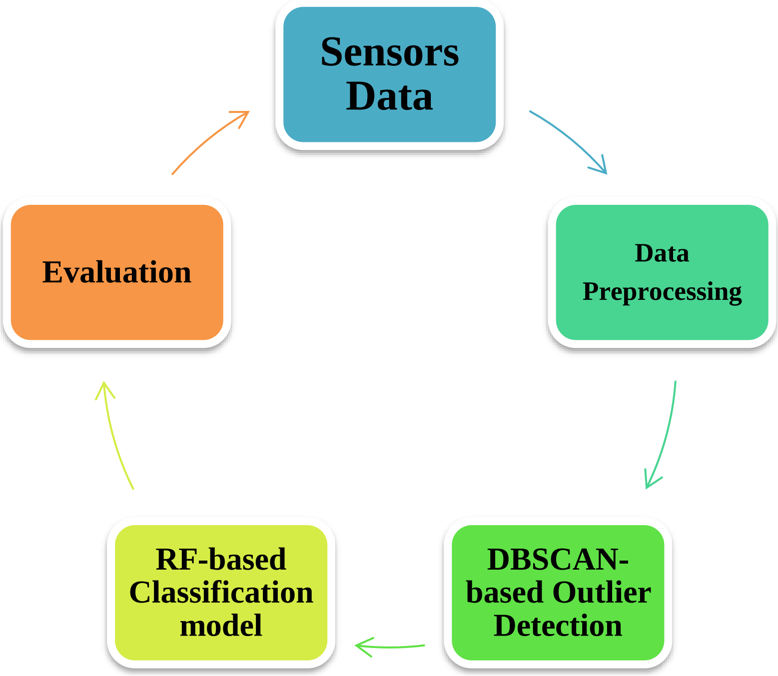


Figure 4. Frequently occurring normal and abnormal errors during the default manufacturing process.

For comparative analysis, the dataset is obtained from a lab with an IoT sensor installed to it. The obtained data from the lab has 340 instances. Those instances are divided either as normal or abnormal events. The features of the comparative analysis dataset are as follows:

1. Humidity
2. Temperature
3. Value of the accelerometer for (x,y,z-axis)
4. Value of gyroscope (x-axis,y-axis, and z-axis)

Almost 100 datasets are labeled as 'YES,' and 240 datasets are labeled as "NO." here, the YES is used to represent the abnormal event, and the NO is used for representing the abnormal events class means which does not take place during the manufacturing process. The predicted outcome is presented in the IoT sensor after the classification of the monitoring system. Table 3 illustrates the dataset distribution based on the mean and STD deviation. [22] IG [Information Gain] can be used for analyzing the significance of the features. [23] utilized Weka version 3.6.15 software for analyzing the s significance of the features via IG. Table 4 illustrates the data attributes and Information gain score. From the outcome, the temperature is one of the significant factors responsible for affecting the abnormal event during the manufacturing process.

Table 3. Distribution of dataset.

Feature	Description	Normal class		Abnormal class	
		Mean	STD	Mean	STD
Temp	Temperature	23.6754336	1.3456548	25.988475	2.87773920
Hum	Humidity	20.6745489	1.6856344	20.987657	12.8973738
ax	x value of accelerometer	1.76398493	1.8975664	-1.876568	5.89937399
ay	y value of accelerometer	2.1897647	1.2384783	4.9877637	9.09837884
az	z value of accelerometer	16.87904	1.8709833	49.876366	155.898738
gx	x value of accelerometer	-0.134567297	0.9087677	-0.0128767	0.12378644
gy	y value of accelerometer	-0.17865343	0.0983456	-0.2345667	0.18975349
gz	z value of accelerometer	0.98546526	0.0345673	0.7653637	0.08736637

Table 4. Features of the IG score

Feature	**Temp**	**ay**	**gy**	**hum**	**ax**	**gz**	**az**	**gx**
IG score	2.0987	1.87	1.845	0.9878	0.6743	0.9864	0.8683	0.7654

The Density-Based Spatial Clustering of Applications with Noise was used for outlier detection. Outlier is known as the point placed outside the primary region. The critical point parameters of DBSCAN are the minimum points (*MinPts*) and the Epsilon (*eps*). Epsilon (*eps*) is a distance of the neighborhood, and minimum points (*MinPts)* define the lowest number of neighborhood pints within the eps radius. Both the minimum points (*MinPts*) and the Epsilon (*eps*) should not be too big. If it is bigger than the expected range, it will be classified as normal data. The experiment found that the minimum points (MinPts) are known by the value five, and the Epsilon (*eps*) is known by the value 7. Figure 5 illustrates the outcome of Density-Based Spatial Clustering of Applications with Noise implementation in 2-D graph. In the DBSCAN, the dataset has been grouped into 3 clusters. The cluster is represented as 1,2 and 3. But the outlier did not belong to this cluster group and was defined as the zeroth cluster. Table 5 illustrates the outcome of the DBSCAN outlier detection. The outlier data will be removed from the dataset and will be sent for further analysis.

. Finally, the outlier data were removed from the dataset, and the remaining data were used for further analysis.

Table 5. The outcome of DBSCAN process

# Instance (Original)	347
Minpts	6
Eps	8
# Outlier Data	5

# Normal Data	340

RF is the most commonly known classification method, especially for real-world related classification problems. [24] RF is the combination of decision trees for stable prediction and accuracy. The trees of RF are constructed independently for selecting the subset randomly with bootstrap sampling and features of the dataset. For each decision, the RF will generate a prediction output, and the final vote is based on the last prediction output. From the outcome of RF, it is seen that it perfectly detected the crash and stopped the system from further movements like forecasting and maneuvering. In the proposed hybrid study, the DBSCAN is used for detecting the outlier data and remove the data from the dataset, and the RF is used for dataset training. The outcome from the hybrid model is compared to know the testing accuracy of the model.

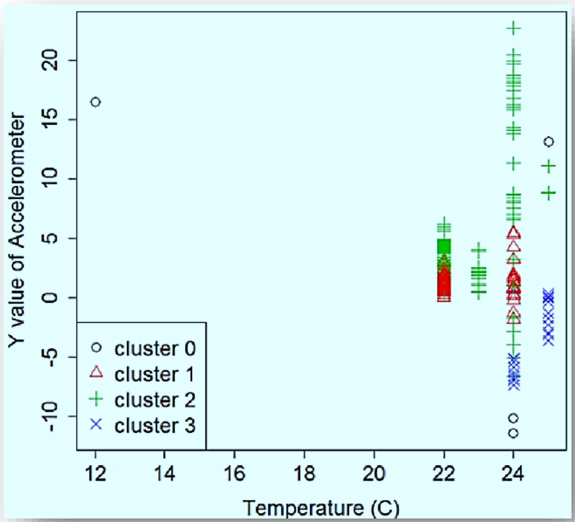


Figure 5. The outcome from DBSCAN

According to the confusion matrix [CM], the predicted output can only have 4 possible outputs, as described in table 6. The TP and TN, commonly known as the true positive and true negative, are used to number the correctly classified points. The FP [False positive] and the FN [False Negative] are used for numbering the incorrect points. Where YES is used for positive, and NO is used for negative. But in the case of the proposed model, YES is used for representing the abnormal events, and NO is for normal events. 10-fold validation is used for testing and training the classification model. The outcome is gained by validating the average from all the fold. [27] utilized a Weka software version 3.6.15 for running the classification model. Table 7 illustrates the performance evaluation for the classification model.

Table 6. The predicted output of Confusion Matrix

	Actual “Yes”	Actual “No”
Classified as “Yes”	TP	FP
Classified as “No”	FN	TN

Table 7. Classification Model’s Performance matrics.

Performance Metric	**Precision**	**Recall/Sensitivity**	**Accuracy**
Formula	TP/(TP+FP)	TP/(TP+FN)	(TP+TN)/(TP+TN+FP+FN)

4. Results and Discussions

4.1. Real-Time Monitoring System

For the proposed model, the data visualization process is developed via JavaScript for presenting real-time data. The IoT sensor sends the data to the python programmed Kafka server. From here, the data will be published and then will be sent for processing by parallel storm. In the parallel storm, the outlier detection and classification take place. After implementing the test, the data will be stored in a MongoDB for presenting it as a web-based monitoring system for the managers. Figure 6 shows the web-based monitoring system. The proposed model was applied for one of the automobile industries in India from August 1st, 2017, till March 31st of 2018. 4 Internet of things sensors were installed to the assembling process, and it transmits the data to the cloud server every five seconds. Around 19 million records (with an approximate size is 3 gigabytes) have been collected during this testing period.

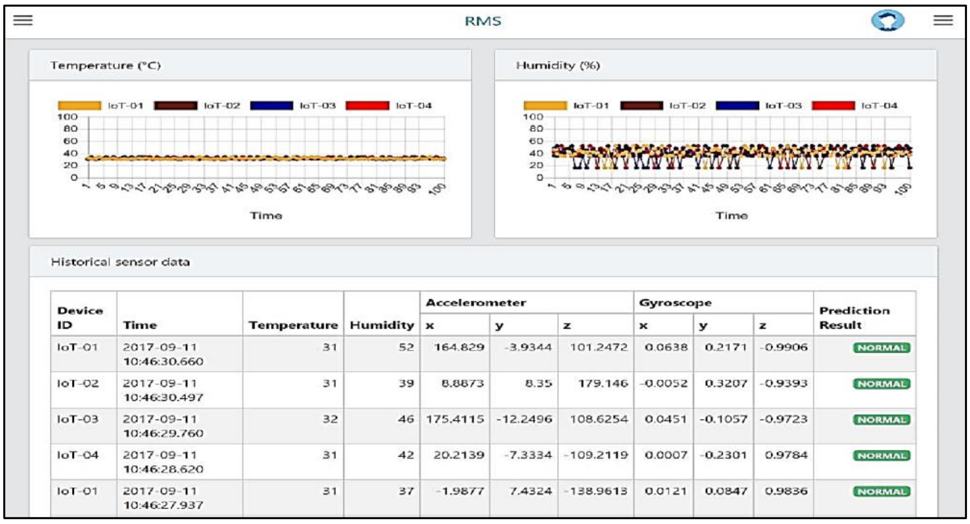
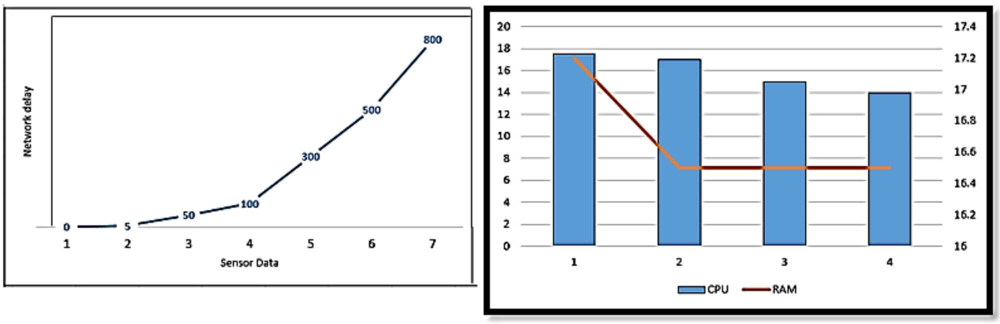


Figure 6. Cloud-based real-time monitoring system

4.2. Performance of the IoT-Based Sensor

The Internet of things sensor is made up of sensors to store and retrieve the sensor data and send them to the cloud system. The performance of the IoT sensor must be analyzed under various conditions. Here the performance metrics are the CPU system, network delay, memory usage, etc. need to be explored. [28] considered network delay as a performance metric while [29] considered CPU storage as the primary metric to evaluate the performance of IoT sensors in various conditions. In the proposed study, network delay is denoted as the average time taken for sending the sensor from the source to the receiver in the destination (MongoDB). At the same time, the 2nd important performance metric is the CPU and memory usage of the client program under various circumstances. Here the client program is the proposed parameter needed to be identified. For experimentation, the Internet of things sensor with Linux Raspbian OS Jessie; 2 GB RAM was utilized. Communication between the sensor and the cloud system is done with the usage of a Wi-Fi connection. Figure 7 a) demonstrates the network delay of various sensor data. From the figure, it is understood that delay in the network increases with the increase in the transfer of sensor data to the other devices. Approximately it takes about fifty seconds to send 1000 data points at-a-time. Figure b) demonstrates the CPU and memory storage from the client-side. In this process, 4 different reading was noted for every 5,10,15,30 seconds. From the outcome, it can be found that the CPU and memory storage has less impact.



a) b)

Figure 7. a) Network delay, b) Memory and CPU storage of IoT system

4.3. The Performance of Big Data Processing

In this study, throughput, concurrency, and system latency are vital metrics for big data processing. In the proposed study system, latency is regarded as the first important performance metric used to process, handle, and store the data to the DB [DataBase] system. Throughput is the overall sensor for data processing for every second. Lastly, concurrency is the access made by the clients to the system. The experimentation was conducted for various servers, and the RT [Response Time] was obtained from the analysis. The data visualization process is developed via JavaScript for presenting real-time data. The IoT sensor sends the data to the python programmed Kafka server. From here, the data will be published and then will be sent for processing by parallel storm. With the help of the java program, any client can access the system. Table 8 gives the detailed specification of the client. The data size was found to be 211 bytes which include the date, time, device ID. Figure 8a describes that the data in the cloud system increases with the increase in RT. The RT is also affected by clients. But the RT can be reduced by additional servers, as described in fig 8 b. Figure 8 (c and d) shows the throughput value with different clients. It was found that better performance is attained by increasing the servers. Figure 8 (e and f) compares the database size and latency of CouchDB and MongoDB. In this process, with the usage of 1 client, different sensor data is sent to the cloud system at-a-time. On the client-side to send sensor data java program is used for sending it to the cloud system. From the experimentation, it is clear that MongoDB overpowers CouchDB during increased sensor data time. Also, MongoDB requires less database size than CouchDB.

Table 8. Detailed description of client and server system

		Server	**Client**
Hardware	Processor	Core i8-5478	Core i7-4876
	CPU RAM	3.78 GHz x 8 cores 17 GB	3.78 GHz x 8 cores 17 GB
	HDD	SSD 127 GB	SSD 129 GB
Software	OS	Ubuntu Server	Windows 10 Pro
	Node.js	15.98	64 bit
	Express	8.5.8	
	Socket.IO	4.9.8	
	Apache	2.8.9	
	Kafka	1.7.4	
	Apache	0.9.4	
	Storm		
	MongoDB		
	JDK	-	1.9.8_123
	Eclipse	-	5.8.6
	HttpClient	-	5.4.3

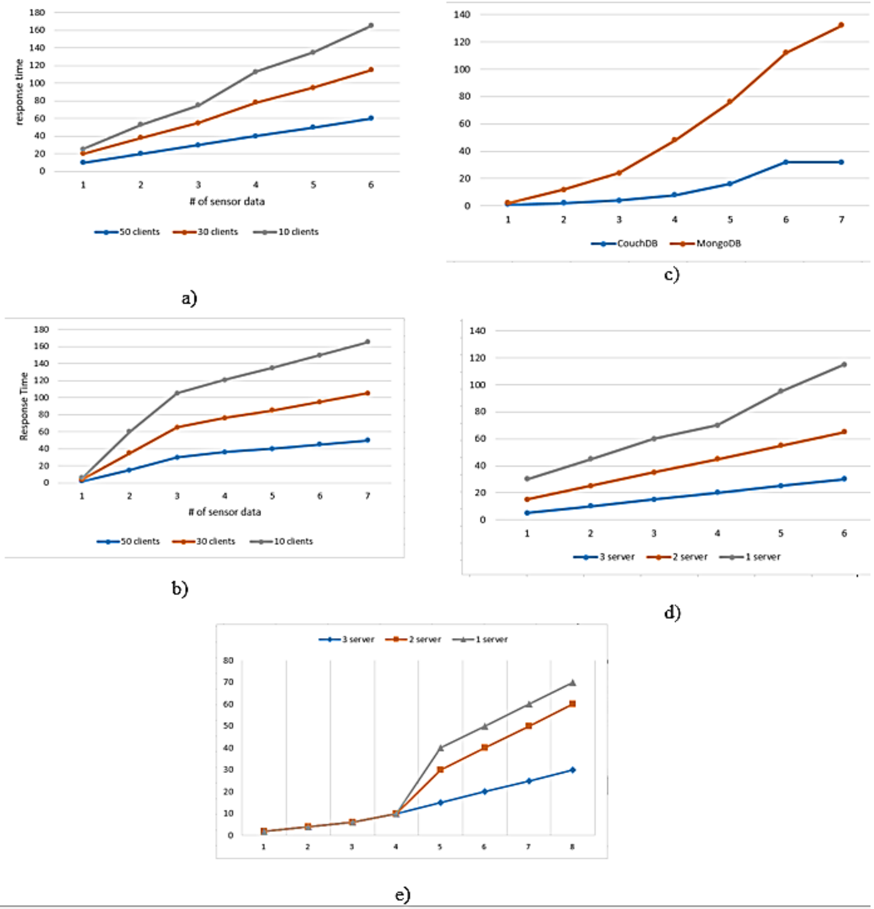


Figure 8. a) evaluating the performance using different clients, b) throughput value using different clients, c) server system value, d) comparision between the CouchDB & MongoDB, e) value of data size.

Hybrid Prediction Model for Fault Detection

In this process, the system receives data from the IoT system during data generation and then stores the data in the MongoDB dataset. The IoT data is collected from various processes, such as during the operation of normal and abnormal events. The collected dataset is labeled. The labeled dataset will be analyzed using the proposed hybrid model for outlier detection. Table 9 shows the comparison outcome of the classification model. The comparison is made by comparison with the well-known classification models, which include the Logistic Regression (LR), Naïve Bayes (NB), and Multilayer Perceptron (MLP). The proposed random forest model attained 100% accuracy when compared to the other existing model. The accuracy of the model increased after the implementation of Density-Based Spatial Clustering of Applications for outlier detection. The hybrid model altogether increased the accuracy by about 1.46% compared to the non-hybrid RF. The accuracy of outlier detection increased by about 3.17% than the other existing model. Figure 6 describes the outcome of the implemented real-time prediction in Apache Storm. The study results prove that it may help the automobile management regarding unexpected loss due to fault in the initial stage of manufacturing, and it also improves the decision-making for the manufacturing process.

Table 9. Performance of various classification model for detecting the fault

Model	NB	LR	MLP	RF	DBSCAN+NB	DBSCAN+LR	DBSCAN+MLP	DBSCAN+I
Precision)	95.2	97	97.9	97.9	97.9	97.6	97.8	100
Recall)	94.6	97	97.9	97.6	97.3	97.8	99.7	100
Accuracy)	94.786	98.966	95.789	98.567	97.890	98.09	97.67	100

4.4. Managerial Implications

The proposed model of the study can be grouped into 3 parts as

1. IoT sensor system
2. Big data processing
3. Machine Learning Model

In the initial stage, the Internet of Things sensor was developed based on Raspberry Pi. Raspberry Pi is used due to its low cost, small size, 1-board computing. By analyzing the previous study, it is proved that the proposed Internet of Things sensor can 100% be used for monitoring the real-time manufacturing in the automobile industry. Also, it was found that the integration of IoT, big data processing, and ML seems to be a promising one to effectively monitor the manufacturing process. Early fault detection is also possible to know.

5. Conclusions

In this research, a real-time monitoring system using Internet of things based sensors for big-data processing and prediction has been proposed for applications in theautomobile industries. Initially, the IoT-based sensor collects important parametric data. The obtained big-data is processed using Apache Kafka, which is a type of message queue. For further real-time processing, and Apache storm engine is proposed. MongoDB is used to store theinformation from the sensor data collected during the manufacturing process. DBSCAN [Density-Based Spatial Clustering of Applications] and Random forest-based noise-based outlier detection are used to remove the outlier information from the sensor data. From the experimentation, the proposed system is having better scalability and can also process large quantities of data more effectively and efficiently than the conventional model. The proposed Internet of Things sensors performance metric, including the CPU storage, network delay, and memory usage, was analyzed. The IoT system seems to be a good one for data collection and data transmission with less computational time and cost. Fault detection is the one that helps in identifying the abnormalities. The proposed DBSCAN separates the outliers from normal data; the RF classifies the error in the separated outlier data used as input. The study's outcome proves that it may help the automobile management regarding unexpected loss due to fault in the initial stage of manufacturing, and it also improves the decision-making for the manufacturing process. It should also be taken into concern for the security issue when more of IoT device is used. Future studies will be based on security detection for the proposed model. Also, the variety of abnormalities conditions for the manufacturing process should be focused.

References

1. Cantore, Nicola, Michele Clara, Alejandro Lavopa, and Camelia Soare. "Manufacturing as an engine of growth: Which is the best fuel?." Structural Change and Economic Dynamics 42 (2017): 56-66.
2. Tallón-Ballesteros, Antonio J., and José C. Riquelme. "Deleting or keeping outliers for classifier training?." In 2014 Sixth World Congress on Nature and Biologically Inspired Computing (NaBIC 2014), pp. 281-286. IEEE, 2014.
3. Ijaz, Muhammad Fazal, Ganjar Alfian, Muhammad Syafrudin, and Jongtae Rhee. "Hybrid prediction model for type 2 diabetes and hypertension using DBSCAN-based outlier detection, synthetic minority over sampling technique (SMOTE), and random forest." Applied Sciences 8, no. 8 (2018): 1325.
4. Ester, M., H. Kriegel, J. Sander, and X. Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Association for the Advancement of Artificial Intelligence." (2018).
5. Zhu, Xianglin, Khalil Ur Rehman, Bo Wang, and Muhammad Shahzad. "Modern soft-sensing modeling methods for fermentation processes." Sensors 20, no. 6 (2020): 1771.
6. Abid, Aymen, Abdennaceur Kachouri, and Adel Mahfoudhi. "Outlier detection for wireless sensor networks using density-based clustering approach." IET Wireless Sensor Systems 7, no. 4 (2017): 83-90.
7. Shahmoradi-Moghaddam, Hani, Kaveh Akbari, Seyed Jafar Sadjadi, and Mahdi Heydari. "A scenario-based robust optimization approach for batch processing scheduling." Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 230, no. 12 (2016): 2286-2295.
8. Zheng, Hao, Yixiong Feng, and Jianrong Tan. "A hybrid energy-aware resource allocation approach in cloud manufacturing environment." IEEE Access 5 (2017): 12648-12656.
9. Bouzary, Hamed, and F. Frank Chen. "Service optimal selection and composition in cloud manufacturing: a comprehensive survey." The International Journal of Advanced Manufacturing Technology 97, no. 1 (2018): 795-808.
10. Li, Xinyu, and Liang Gao. "An Agent-Based Approach for IPPS." In Effective Methods for Integrated Process Planning and Scheduling, pp. 191-208. Springer, Berlin, Heidelberg, 2020.
11. Lin, Ting Yu, Chen Yang, Changhui Zhuang, Yingying Xiao, Fei Tao, Guoqiang Shi, and Chao Geng. "Multi-centric management and optimized allocation of manufacturing resource and capability in cloud

manufacturing system." Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 231, no. 12 (2017): 2159-2172.

12. Liu, Mi, Shuping Yi, and Peihan Wen. "Quantum-inspired hybrid algorithm for integrated process planning and scheduling." Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 232, no. 6 (2018): 1105-1122.
13. Lüder, Arndt, Ambra Calá, Jacek Zawisza, and Ronald Rosendahl. "Design pattern for agent based production system control—A survey." In 2017 13th IEEE Conference on Automation Science and Engineering (CASE), pp. 717-722. IEEE, 2017.
14. Darshanapriya, GB Prabash. "Multi-agent based dynamic scheduling system for manufacturing." PhD diss., 2019.
15. Saeidlou, Salman, Mozafar Saadat, Ebrahim Amini Sharifi, and Guiovanni D. Jules. "Agent-based distributed manufacturing scheduling: an ontological approach." Cogent Engineering 6, no. 1 (2019): 1565630.
16. Batygin, Yuriy, Marina Barbashova, and Oleh Sabokar. Electromagnetic metal forming for advanced processing technologies. Springer International Publishing, 2018.
17. Tatipala, Sravan, Johan Pilthammar, Mats Sigvant, Johan Wall, and Christian M. Johansson. "Introductory study of sheet metal forming simulations to evaluate process robustness." In IOP Conference Series: Materials Science and Engineering, vol. 418, no. 1, p. 012111. IOP Publishing, 2018.
18. Cillaurren, Joseba, Lander Galdos, Mario Sanchez, Alaitz Zabala, Eneko Saenz de Argandoña, and Joseba Mendiguren. "Contact pressure and sliding velocity ranges in sheet metal forming simulations." (2021).
19. Stefan, D. "Model based control of product properties: Challenges and Opportunities." In proceedings of Forming Technology Forum, pp. 2-3. 2017.
20. Xiao, Wenxing, Gang Du, Yingying Zhang, and Xiaojie Liu. "Coordinated optimization of low-carbon product family and its manufacturing process design by a bilevel game-theoretic model." Journal of Cleaner Production 184 (2018): 754-773.
21. Dai, M., D. B. Tang, Y. C. Xu, and W. D. Li. "Energy-aware Integrated Process Planning and Scheduling for Job Shops." In Sustainable Manufacturing and Remanufacturing Management, pp. 13-36. Springer, Cham, 2019.
22. Xu, Yanni, Sébastien Thomassey, and Xianyi Zeng. "AI for apparel manufacturing in Big data era: a focus on cutting and sewing." In Artificial Intelligence for Fashion Industry in the Big Data Era, pp. 125-151. Springer, Singapore, 2018.
23. Cruz Introini, Sabrina, Andrés Boza, and María Del Mar Alemany Díaz. "Traceability in the Food Supply Chain: Review of the literature from a technological perspective." Dirección y Organización 64 (2018): 50-55.
24. Jachimczyk, Bartosz, Rafal Tkaczyk, Tomasz Piotrowski, Sven Johansson, and Wlodek Kulesza. "IoT-based Dairy Supply Chain-An Ontological Approach." Elektronika ir Elektrotechnika 27, no. 1 (2021): 71-83.
25. Qian, Xiaoming, Jiachen Tu, and Peihuang Lou. "A general architecture of a 3D visualization system for shop floor management." Journal of Intelligent Manufacturing 30, no. 4 (2019): 1531-1545.
26. Chen, Wei. "Intelligent manufacturing production line data monitoring system for industrial internet of things." Computer Communications 151 (2020): 31-41.
27. Dalenogare, Lucas Santos, Guilherme Brittes Benitez, Néstor Fabián Ayala, and Alejandro Germán Frank. "The expected contribution of Industry 4.0 technologies for industrial performance." International Journal of Production Economics 204 (2018): 383-394.
28. Syafrudin, Muhammad, Ganjar Alfian, Norma Latif Fitriyani, and Jongtae Rhee. "Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing." Sensors 18, no. 9 (2018): 2946.
29. Hajjaji, Yosra, Wadii Boulila, Imed Riadh Farah, Imed Romdhani, and Amir Hussain. "Big data and IoT-based applications in smart environments: A systematic review." Computer Science Review 39 (2021): 100318.
30. McDonald, Daniel, and S. Jeff Cold. "Using raspberry pi computers to teach lamp and wordpress." Journal of Computing Sciences in Colleges 33, no. 2 (2017): 148-154.
31. Balogh, Zoltán, Martin Magdin, and György Molnár. "Motion Detection and Face Recognition using Raspberry Pi, as a Part of, the Internet of Things." Acta Polytechnica Hungarica 16, no. 3 (2019): 167-185.
32. Dow, Colin. Internet of things programming projects: build modern IoT solutions with the Raspberry Pi 3 and Python. Packt Publishing Ltd, 2018.
33. Prasdika, Prasdika, and Bambang Sugiantoro. "A Review Paper on Big Data and Data Mining Concepts and Techniques." IJID (International Journal on Informatics for Development) 7, no. 1 (2018): 36-38.

34. Han, Jiawei, Micheline Kamber, and Jian Pei. "Data mining concepts and techniques third edition." The Morgan Kaufmann Series in Data Management Systems 5, no. 4 (2011): 83-124.
35. Alazzawi, L., and A. Elkateeb. "Performance evaluation of the WSN routing protocols scalability." Journal of Computer Systems, Networks, and Communications 2008 (2008).
36. Singh, Harmanpreet, and Damanpreet Singh. "Hierarchical clustering and routing protocol to ensure scalability and reliability in large-scale wireless sensor networks." The Journal of Supercomputing (2021): 1-19.
37. Mouiz, Anas, Abdelmajid Badri, Abdennaceur Baghdad, and Aïcha Sahel. "Performance Evaluation of OLSR and AODV Routing Protocols with different Parameters in Mobile Ad-hoc Networks using NS2 Simulator." In Proceedings of the 2019 5th International Conference on Computer and Technology Applications, pp. 134-139. 2019.
38. Sajwan, Mohit, Ajay K. Sharma, and Karan Verma. "Analysis of scalability for hierarchical routing protocols in wireless sensor networks." In Proceedings of ICETIT 2019, pp. 107-116. Springer, Cham, 2020.
39. Sangat, Prajwol, Maria Indrawan-Santiago, and David Taniar. "Sensor data management in the cloud: Data storage, data ingestion, and data retrieval." Concurrency and Computation: Practice and Experience 30, no. 1 (2018): e4354.
40. Witthayawiroj, Niti, and Pongpon Nilaphruek. "The development of smart home system for controlling and monitoring energy consumption using WebSocket protocol." In IOP conference series: materials science and engineering, vol. 185, no. 1, p. 012019. IOP Publishing, 2017.
41. Qi, Xinbo, Guofeng Chen, Yong Li, Xuan Cheng, and Changpeng Li. "Applying neural-network-based machine learning to additive manufacturing: current applications, challenges, and future perspectives." Engineering 5, no. 4 (2019): 721-729.