mathematics

MDPI

Article

Smart Machinery Monitoring System with Reduced Information Transmission and Fault Prediction Methods Using Industrial Internet of Things

Ming-Fong Tsai [1,*], Yen-Ching Chu [2], Min-Hao Li [1] and Lien-Wu Chen [2]

[1] Department of Electronic Engineering, National United University, Miaoli 360, Taiwan; u0522115@gm.nuu.edu.tw
[2] Department Information Engineering and Computer Science, Feng Chia University, Taichung 407, Taiwan; yenching203@mail.fcu.edu.tw (Y.-C.C.); lwuchen@fcu.edu.tw (L.-W.C.)
* Correspondence: mingfongtsai@gmail.com or mftsai@nuu.edu.tw

Abstract: A monitoring system for smart machinery has been considered to be one of the most important goals in recent enterprises. This monitoring system will encounter huge difficulties, such as more data uploaded by smart machines, and the available internet bandwidth will influence the transmission speed of data and the reliability of the equipment monitoring platform. This paper proposes reducing the periodical information that has been uploaded to the monitoring platform by setting an upload event through the traits of production data from machines. The proposed methods reduce bandwidth and power consumption. The monitoring information is reconstructed by the proposed methods, so history data will not reduce storage in the cloud server database. In order to reduce the halt time caused by machine error, the proposed system uses machine-learning technology to model the operating status of machinery for fault prediction. In the experimental results, the smart machinery monitoring system using the Industrial Internet of Things reduces the volume of information uploaded by 54.57% and obtains a 98% prediction accuracy.

Keywords: smart machinery; reducing information; fault prediction

check for updates

Citation: Tsai, M.; Chu, Y.; Li, M.; Chen, L. Smart Machinery Monitoring System with Reduced Information Transmission and Fault Prediction Methods Using Industrial Internet of Things. *Mathematics* **2021**, *9*, 3. https://dx.doi.org/10.3390/math9010003

Received: 27 October 2020
Accepted: 17 December 2020
Published: 22 December 2020

1. Introduction

The development and application of smart machines are important in enhancing the efficiency of production and management. Hence, the applications of Internet of Things to smart machines are generally defined as the Industrial Internet of Things [1]. Machinery industries around the world have utilized the technology of smart machines and the automation of assembly lines, and the basic step of developing this technology is to send the production data of these machines to a monitoring platform in real-time, such as inspection of ball-bearing malfunction [2], industrial data-driven monitoring [3], ball-bearing vibration data [4], remote wind turbine condition monitoring [5], vibration monitoring for smart maintenance [6] and analysis of vibration time histories [7]. As all of the data, including error signals, voltage, current and the operation status of machines in the plant, are sent to the monitoring platform via the Internet, the manager can make decisions using information such as infographics and smart analysis from the monitoring platform to reduce the possibility of machine failure, such as health monitoring on marine vessels [8], creating an ecosystem for fault detection in predictive maintenance [9], peak-load forecasting for small industries [10], identification of bearing faults from vibration signals [11] and fuzzy modelling applications [12]. In view of the fact that the machine equipment controller requires thousands of production parameters to be processed during the manufacturing process, the production parameters are periodically returned to the cloud server monitoring platform through the Internet. This occurs in addition to the waste of available bandwidth on the Internet and makes it difficult to achieve real-time monitoring due to the long manufacturing time of machine production components or electronic parts, which will be subject to deformation and wear from the influence of rotation or movement. When the production parameter data and abnormal condition sensing device data can be successfully stored after reaching the cloud server monitoring platform, the system can perform predictive maintenance data analysis for each production component to facilitate the early replacement and repair of production components or electronic parts; this makes it possible to avoid unnecessary mechanical equipment downtime or production component wear and other related property loss, such as machine maintenance management and repair predictions [13], real-world dataset from the automobile parts [14], machinery fault detection and diagnosis [15], a cloud platform for equipment component recognition [16], identification of multiple faults in rotor-bearing systems [17] and classification of machinery vibration signals [18]. The significant operational parameters for sensing and identification in a real industrial system are very important and affect prediction accuracy [19].

In order to effectively reduce the upload of production parameter data to the cloud server real-time monitoring platform, this paper proposes setting the return event trigger according to the changing characteristics of the production parameter data of the mechanical equipment and to customize the tolerance value based on the change of the parameter values of each piece of mechanical equipment. A lower frequency of transmission to the cloud server real-time monitoring platform will greatly reduce the return of production parameter data and abnormal condition sensing device data, thereby reducing the use of available internet bandwidth and the cloud server database burden. In addition, in order to reduce the related property losses caused by the downtime of mechanical equipment, this paper proposes analyzing and modelling the historical abnormal information data of the mechanical equipment to provide replacement or repair decisions for production components or electronic parts. The machine-learning algorithms use computational methods to obtain advance information, but the design and planning classification features remain very important issues. Hence, this paper proposes using the abnormal condition sensing component data for equipment failure, analyzing and modelling to avoid mechanical equipment downtime and making production component or electronic part replacement and maintenance reminders early. The classification features include the individual values for the first, second and third seconds: the average value of the first and second seconds; the average value of the second and third seconds; the average value of all three seconds. This paper is the first to use machine-learning algorithms for a smart machinery monitoring system for bottle blowing machines. In Section 2, we review some related works. In Section 3, we introduce the proposed system architecture and methods. In Section 4, we describe the experimental environment and the results. Finally, the conclusions of this study are shown in Section 5.

2. Related Work

This paper is the first to use machine-learning algorithms for a smart machinery monitoring system for bottle blowing machines. Hence, this section discusses the background of Industry 4.0 and the abnormal signals of machines.

2.1. Background

Three revolutions have occurred in industrial history. The first industrial revolution was the use of steam power to run machines. The second industrial revolution was the operation of machines with electricity. In the third industrial revolution, machines were controlled by digital electronic devices. The fourth industrial revolution is based on smart factories and smart manufacturing. A machine uses the Internet to closely connect factories and customers for intelligent production. Industry 4.0 is not only a new industrial technology but also the integration and digitization of machine and enterprise management, combining industrial technology, the Internet of Things and smart sensor components. It involves letting each machine communicate by itself, changing different materials according to different orders and allowing engineers to use computers for remote use. In order to realize the automatic adjustment of production steps, it is necessary to predict the abnormal state of a machine, and a cloud server should be built to store, analyze and monitor, for example, the key aspects of and solutions provided by machine learning [20].

2.2. Error Signals of Machines

Currently, monitoring systems for smart machinery involve connecting computers to controllers using Ethernet and reading real-time data through the parameter name or the memory location in the controller process. Hence, a machine's status and abnormal signals can be uploaded to the database via Ethernet. In addition to recording real-time data from the machine, a smartphone application will be provided to allow employees to perform repairs and record maintenance. Maintenance records owned by the engineer will also be recorded. Hence, it uses the abnormal state reported by the employees to analyze the abnormal state of the machine, and it records the abnormal signal of the machine and the engineer's record, such as state recognition and failure prediction for the axial piston pump [21].

3. Materials and Methods

The method proposed in this paper includes reduced data transmission, high accuracy machine status analysis and high accuracy machine maintenance prediction to monitor machines in order to reduce halt time.

3.1. Reduced Data Transmission

The Smart Machinery Monitoring System includes two methods to reduce the volume of data being uploaded to the monitoring platform. The first method is comparing the latest data with the previous data. If the difference exceeds the tolerance value, the latest data will be uploaded, and vice versa, as shown in Figure 1. The second method can decide whether data should be uploaded to reduce the amount of uploaded data. As shown in Figure 2, the values of C, D, E and G in the latest data are related to the previous data. The C value has a linearly decreasing correlation, the G value has a linearly increasing correlation and the D and E values are the same as the previous D and E values. The changes in these values are the same as the rules of record, so these values will not be uploaded. The remaining values, which are not related to the previous data, will be uploaded packed in JavaScript Object Notation (JSON) format. If the cloud server database does not receive the value of a parameter, it will automatically add the value according to the rules. When the cloud server receives the upload request, the system will retrieve the two most recent pieces of data from the database to calculate the slope (rules of record). The system uses the linear interpolation method to fill in missing values and adds them to the cloud server database.

3.2. High Accuracy Machine Status Analysis

As shown in Figure 3, the vibration values of the machine will be filtered by a high-pass filter to grab the high-frequency acceleration values. Due to the fact that the machine analyzed in this paper has a cycle time of three seconds, the samples were calculated from the average value of the first and second seconds, the average value of the second and third seconds and the average of all three seconds after the acceleration values of a cycle time were collected. As shown in Figure 4 on the X axis, label a is the value of the first second, label b is the value of the second second, label c is the value of the third second, label d is the average value of the first and second seconds, label e is the average value of the second and third seconds and label f is the average value of all three seconds. The Y axis represents the acceleration value. There are six classified traits, including standby, idle, operation I, operation II, operation III and operation IV, as Figure 4 shows. After collecting the six traits and types of labels, the proposed system uses machine learning to classify the operating status. The system calculates the distance vector between all of the samples and the characteristics of each different classifier. It arranges these in order, calculates individual probabilities, builds a vector space, grabs the nearest sample point to a vote, makes a dividing line according to the sample distance or a branch according to the conditions and finally trains six models of six classifiers.

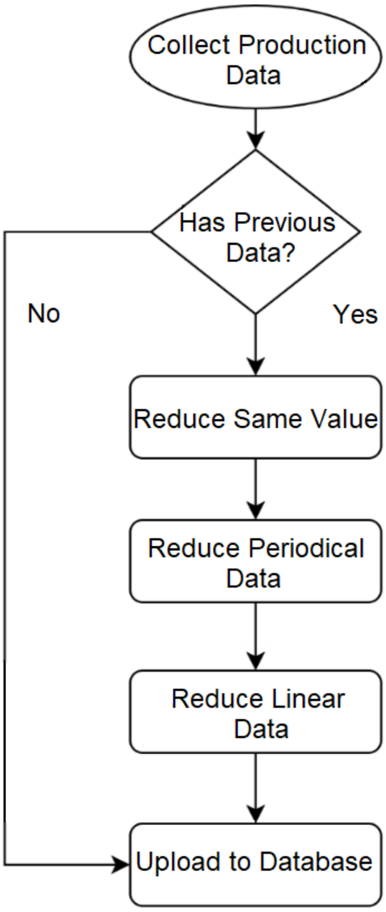
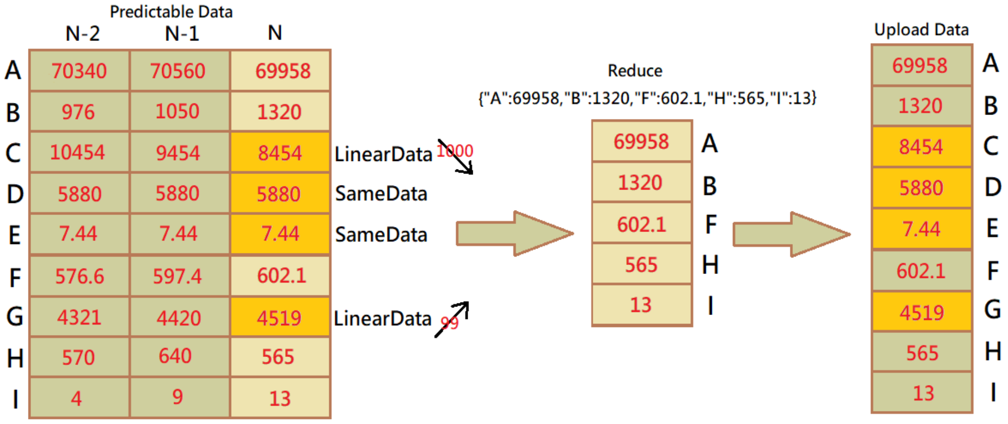


Figure 1. Reducing data transmission flowchart.



	N-2	N-1	N
A	70340	70560	69958
B	976	1050	1320
C	10454	9454	8454
D	5880	5880	5880
E	7.44	7.44	7.44
F	576.6	597.4	602.1
G	4321	4420	4519
H	570	640	565
I	4	9	13

Figure 2. Reducing data transmission methods.

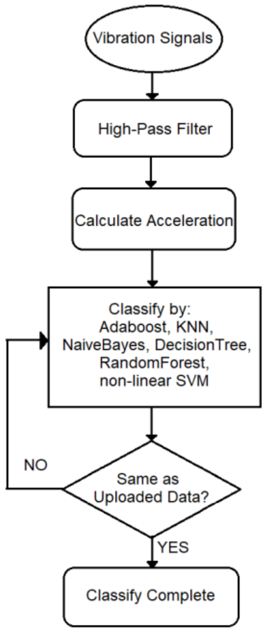


Figure 3. Machine status classification flowchart.

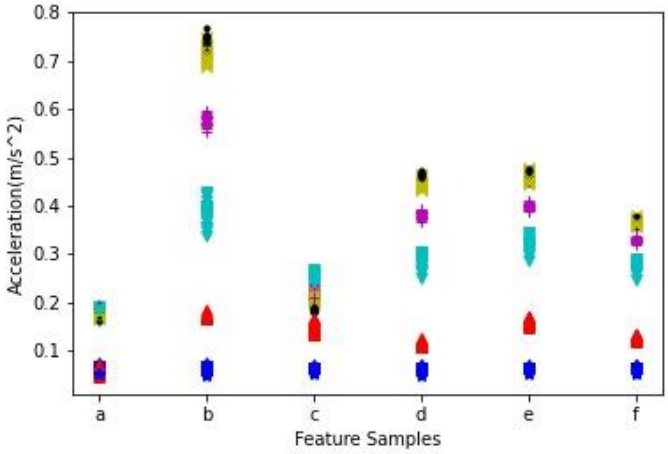


Figure 4. Feature samples.

3.3. *High Accuracy Machine Maintenance Prediction*

There are several sensors added to the key components of a machine in order to monitor its status. The values of the current, voltage, torque, temperature and time of switch of the machine are monitored, preventing these values from going over the limits of the machine and causing the machine to halt. As Figure 5 shows, the method proposed in this paper uses linear regression to predict the time to inspect or perform maintenance on the machine. Some important values that are suitable to be collected correspond to critical components. For instance, rotation speed or current is suitable for monitoring fans, current or torque is suitable for monitoring a servo, the relationship between current and temperature or time is suitable for monitoring a heater and the pressure or time of grab and release is suitable for monitoring a gripper. Figure 6 shows a linear regression model created using grab time (Y axis) and release (X axis). The method compares the latest collected times of grab and release with the predicted values. The system will notify the manager to check or perform maintenance on the machine when the difference between the collected values and the predicted values is too large.

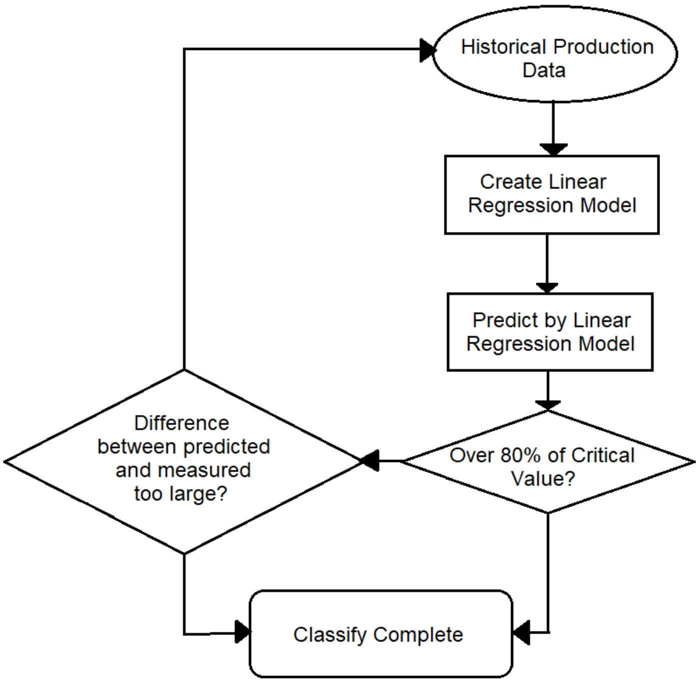

Historical Production Data
Create Linear Regression Model
Predict by Linear Regression Model
Over 80% of Critical Value?
Difference between predicted and measured too large?
Classify Complete

Figure 5. Machine status analysis flow chart.

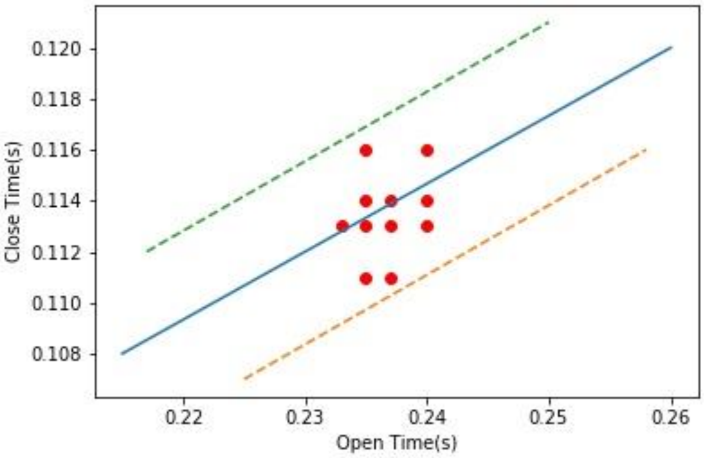


Figure 6. Create model using linear regression.

4. Results and Discussion

The system architecture of the experimental environment is shown in Figure 7. The machine is a blow molding machine. The production data were recorded by the controller installed in the machine, which grabs the data of all of the sensors in this machine. There was an industrial personal computer connected to the controller via Ethernet, which grabbed production data periodically or monitored some parameters from the controller. The data were uploaded to the monitoring platform if the values of these parameters changed. In addition, there was also an accelerometer installed above the molding module of the machine which was set horizontally relative to the direction of the sliding rail. The grabber loaded the signal from the accelerometer through the wire. The industrial personal computer grabbed the data from the grabber via Ethernet, filtered the data by high-pass filter and calculated the instant acceleration value. The values collected every 3 s need to be predicted and classified by the machine-learning classifier. The prediction result from the classifier was uploaded to the cloud platform and compared with the machine status data stored in the cloud platform. The prediction result was sent to the sample data for the classifier to train again if it was different from the machine status data. If there are too many differences between the prediction results and the machine status data, the system will notify the manager that the production status is different from the parameter settings of the controller.

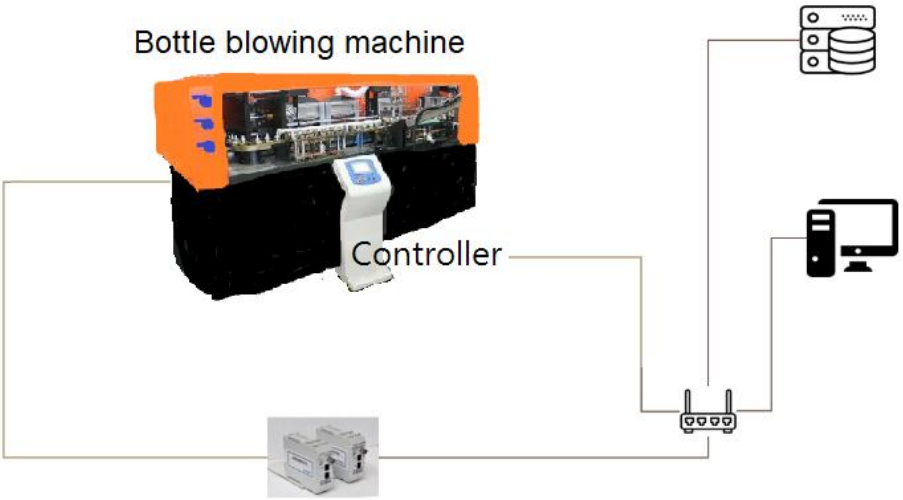


Figure 7. System architecture.

4.1. *Efficiency of Reducing Data Transmission*

The experiment in this paper used an industrial personal computer to grab production data, such as the current production capacity, the volume of produced bottles, the volume of broken bottles, the total volume of produced bottles, the temperature of preforms and the pressure from the controller, and uploaded these data to a server. The results are shown in Figure 8, where the X axis is the time of data collection and the Y axis shows the size of the uploaded data. The total data collection time was eight hours; it is the same as the average production time of the factory. The dotted blue line is the original data length, and the spotted green line is the uploaded data length, which uses the first method proposed in this paper. For the first set of uploaded data in the figure, all of the values were uploaded, because there were no previous data to use to decide which values could be reduced. After the first set of data, previous data were available for comparison. Thus, it was possible to reduce some values by the algorithm. As Table 1 shows, using only the first method makes it possible to reduce the uploaded data length by 36.47%. As Figure 9 and Table 2 show, if both method I and method II are used at the same time, the data length can be reduced by about 54.57% for each set of data.

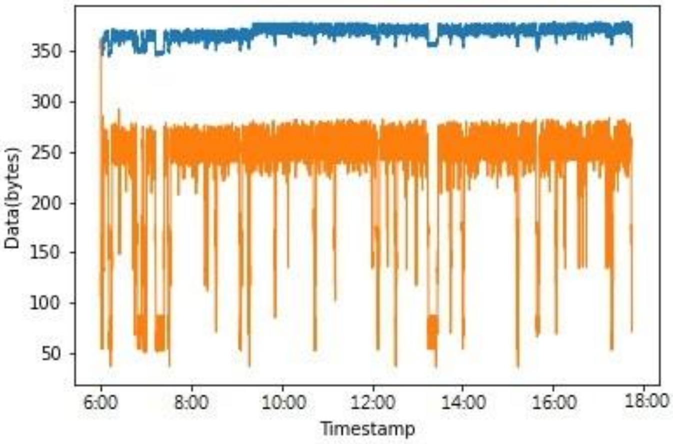


Figure 8. Reduce the length of uploaded data (using only Method I).

Table 1. Average length of data (using only Method I).

Average upload length of each group of data (no algorithm)	368 bytes
Average upload length of each group of data (with algorithm)	233.8 bytes
Percentage reduction	36.47%

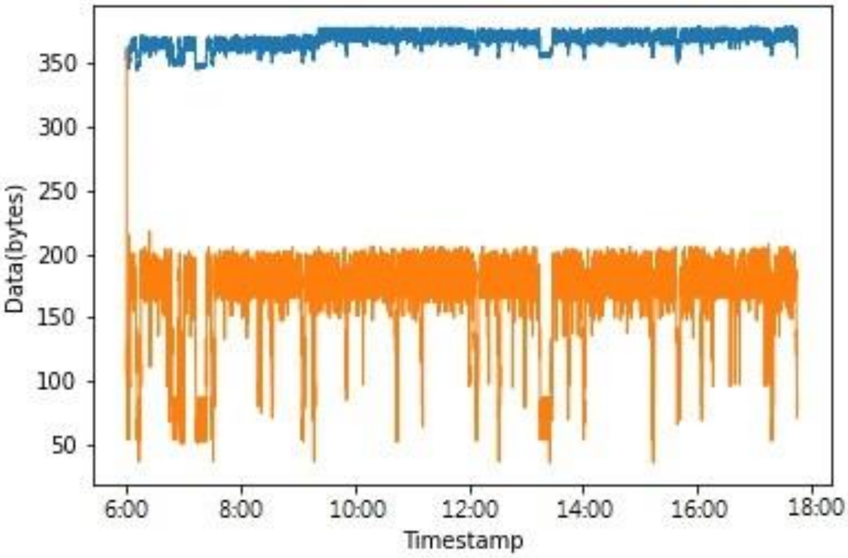


Figure 9. Reduce the length of uploaded data (using both methods).

Table 2. Average length of data (using both methods).

Average upload length of each group of data (no algorithm)	368 bytes
Average upload length of each group of data (with algorithm)	167.2 bytes
Percentage reduction	54.57%

4.2. Accuracy of Machinery Status Analytics

This experiment takes 441 s of the acceleration value, where each second of the acceleration value is filtered by a high-pass filter, and grabs the frequencies higher than 2 kHz. It takes the acceleration values from every 3 s interval as trait samples. The samples used in the paper were the acceleration values of the first second, the second second, the third second, the average of the first and second seconds, the average of the second and third seconds and the average of all three seconds. We acquired a total of 147 groups of trait samples. The Adaboost training model performed several rounds of classification training, first giving each datum the same weight, and trained with several weak classifiers. It reduced the weights of correctly classified data and increased the weights of incorrectly classified data. Next, it re-adjusted the weight distribution and performed the next round of training classification. After several rounds of classification, the weight with the smallest error probability was obtained. Finally, the weak classifiers were combined according to the weight to form a strong classifier to complete the classification model. As shown in the top part of Table 3, all data began with the same weight of 1/6. After being classified by the weak classifier, it was assumed that Blowing III and Blowing IV were incorrectly predicted. The weights were redeployed to increase the predicted error from 1/6 to 1/4, and the correct prediction was reduced from 1/6 to 1/8. As shown in the middle part of Table 3, after the weights were deployed, the classification was performed again. This time the assumption is wrong with regard to Blowing III and Blowing II. The classifier redeployed the weights again, and the final result, Blowing III, accounts for 1/3 of the weight. As shown in the bottom part of Table 3, the combination of strong classifiers can accurately predict that the weak classifier of Blowing III will account for a larger proportion of the entire classifier.

Table 3. Adaboost weight change.

	The n Second	The n + 1 Second	The n + 2 Second	Avg of n, n + 1	Avg of n + 1, n + 2	Avg of all	Weight
StandBy	0.051349	0.054914	0.053374	0.053132	0.054144	0.053212	1/6
Idling	0.065304	0.174465	0.161900	0.119885	0.168183	0.133890	1/6
Blowing I	0.163742	0.338481	0.232462	0.251112	0.285472	0.244895	1/6
Blowing II	0.182730	0.567566	0.234054	0.375148	0.400810	0.328117	1/6
Blowing III	0.165524	0.696522	0.215582	0.431023	0.456052	0.359209	1/6
Blowing IV	0.175751	0.720361	0.203247	0.448056	0.461804	0.366453	1/6
StandBy	0.051349	0.054914	0.053374	0.053132	0.054144	0.053212	1/8
Idling	0.065304	0.174465	0.161900	0.119885	0.168183	0.133890	1/8
Blowing I	0.163742	0.338481	0.232462	0.251112	0.285472	0.244895	1/8
Blowing II	0.182730	0.567566	0.234054	0.375148	0.400810	0.328117	1/8
Blowing III	0.165524	0.696522	0.215582	0.431023	0.456052	0.359209	1/4
Blowing IV	0.175751	0.720361	0.203247	0.448056	0.461804	0.366453	1/4
StandBy	0.051349	0.054914	0.053374	0.053132	0.054144	0.053212	1/10
Idling	0.065304	0.174465	0.161900	0.119885	0.168183	0.133890	1/10
Blowing I	0.163742	0.338481	0.232462	0.251112	0.285472	0.244895	1/10
Blowing II	0.182730	0.567566	0.234054	0.375148	0.400810	0.328117	1/6
Blowing III	0.165524	0.696522	0.215582	0.431023	0.456052	0.359209	1/3
Blowing IV	0.175751	0.720361	0.203247	0.448056	0.461804	0.366453	1/5

To train the K-Nearest Neighbors (KNN) model, first, the initial K value must be given. For each piece of data, the closest K pieces of data were captured, and these pieces of data were classified by voting on these data. The distance between this piece of data and the closest K pieces of data can be used to summarize how the data should be classified. The KNN model for data classification can be regarded as a relatively good predictive algorithm for a bottle blowing machine with a fixed pattern and fixed operating changes. When the latest value was obtained, the K data closest to this value were captured. As shown in Table 4, among these data, Blowing I accounts for one piece and Bowling IV accounts for five. After voting, these data were defined as Bowling IV. As shown in Table 5, the next data also captured the closest K data for voting. In the end, Blowing I received one vote, Blowing III received four votes and Blowing IV one vote. The data were defined as Blowing III. The Naive Bayes classifier first assumed that every second of the data was an independent item. Due to the fact that the conditions were independent, each label could be represented by the multiplication of probability of each condition. There are six kinds of labels in total. Each label has a resulting conditional probability. The classifier compares the probability of each label and finally acquires the label with the highest probability as a result of this classification. Since the bottle blowing machine has fixed and correct information, there is a very high probability of having a label in the calculation, but it is also difficult to catch the deviating data. As shown in Table 6, the classifier received a piece of a new sample for classification. The probability will be calculated according to the conditions of each label. If none of the data of StandBy matches, the probability is 0. Idling has one statistic close, with a probability of 1/6. Blowing I has two data close, and its probability is 2/6. The probability of Blowing II is 4/6. The probability of Blowing III is

6/6. The probability of Blowing IV is 5/6. Finally, the classifier compared the probabilities of each of the labels and defined these data as Blowing III.

Table 4. Blowing IV is the result.

	Feature Sample					
	The n Second	**The n + 1 Second**	**The n + 2 Second**	**Avg of n, n + 1**	**Avg of n + 1, n + 2**	**Avg of all**
Blowing I	0.163742	0.338481	0.232462	0.251112	0.285472	0.244895
Blowing IV	0.175751	0.720361	0.203247	0.448056	0.461804	0.366453
Blowing IV	0.171148	0.731996	0.184157	0.451572	0.458077	0.362434
Blowing IV	0.176430	0.736780	0.187903	0.456605	0.462342	0.367038
Blowing IV	0.180478	0.736250	0.185272	0.458364	0.460761	0.367333
Blowing IV	0.179346	0.725493	0.193486	0.452420	0.459490	0.366108

Table 5. Blowing III is the result.

	Feature Sample					
	The n Second	**The n + 1 Second**	**The n + 2 Second**	**Avg of n, n + 1**	**Avg of n + 1, n + 2**	**Avg of all**
Blowing I	0.163742	0.338481	0.232462	0.251112	0.285472	0.244895
Blowing III	0.182643	0.721272	0.201569	0.451958	0.461421	0.368495
Blowing III	0.167709	0.716777	0.201444	0.442243	0.459111	0.361977
Blowing III	0.165524	0.696522	0.215582	0.431023	0.456052	0.359209
Blowing III	0.168686	0.724333	0.207251	0.446510	0.465792	0.366757
Blowing IV	0.179346	0.725493	0.193486	0.452420	0.459490	0.366108

Table 6. Conditional probability of blowing III.

	The n Second	**The n + 1 Second**	**The n + 2 Second**	**Avg of n, n + 1**	**Avg of n + 1, n + 2**	**Avg of all**
StandBy	0.051349	0.054914	0.053374	0.053132	0.054144	0.053212
Idling	0.065304	0.174465	0.161900	0.119885	0.168183	0.133890
Blowing I	0.163742	0.338481	0.232462	0.251112	0.285472	0.244895
Blowing II	0.182730	0.567566	0.234054	0.375148	0.400810	0.328117
Blowing III	0.165524	0.696522	0.215582	0.431023	0.456052	0.359209
Blowing IV	0.175751	0.720361	0.203247	0.448056	0.461804	0.366453
			Feature Sample			
Blowing III	0.167709	0.716777	0.201444	0.442243	0.459111	0.361977

The decision tree classifier performed multiple conditional judgments on the data and separated the data. It started with a conditional assignment from the beginning to separate the data belonging to the same label. Among the remaining data, it used another condition to separate the data belonging to another label. Then, it continued to allocate and divide the remaining data until all the data were in the same label group. Finally, these conditions were unified to get a decision tree model. As shown in Figure 10, first, it took the first second as the data. It separated StandBy/Idling and Bowling I/II/III/IV with 0.1 m/s as the condition. It took the second second as the data and classified it as StandBy when it was less than 0.05 m/s, Idling when it was greater than 0.05 m/s, Bowling I when it was greater than 0.05 m/s and less than 0.5 m/s, Bowling II when it was greater than 0.5 m/s and less than 0.7 m/s, Bowling III when it was greater than 0.7 m/s and less than 0.72 m/s and Bowling IV when it was greater than 0.72 m/s. Random Forest is a more advanced decision tree classifier that is composed of multiple decision trees. The Random Forest classifier uses multiple decision trees to vote to find the best result. Synthesizing a strong classifier through multiple weak classifiers is similar to Adaboost. However, Random Forest does not allocate weights, which means that every decision tree classifier has the same weight. A support vector machine classifier finds the dividing line with the largest interval. As shown in Figure 11, it found the curve separating different labels in the eigenspace formed by the data eigen. After finding the curve, it compared all the labels on both sides of the dividing line to find the dividing line with the largest distance from all the labels. This dividing line is the model of the Support Vector Machine (SVM). As shown in Table 7, decision tree is the least accurate classifier, which has a 90% chance of correct prediction. The Random Forest is 93% accurate, as are KNN and Naive Bayes. The Adaboost has a 96% chance of correct prediction. The best classifier is nonlinear SVM, which has a 98% chance of correctly predicting the operating status.

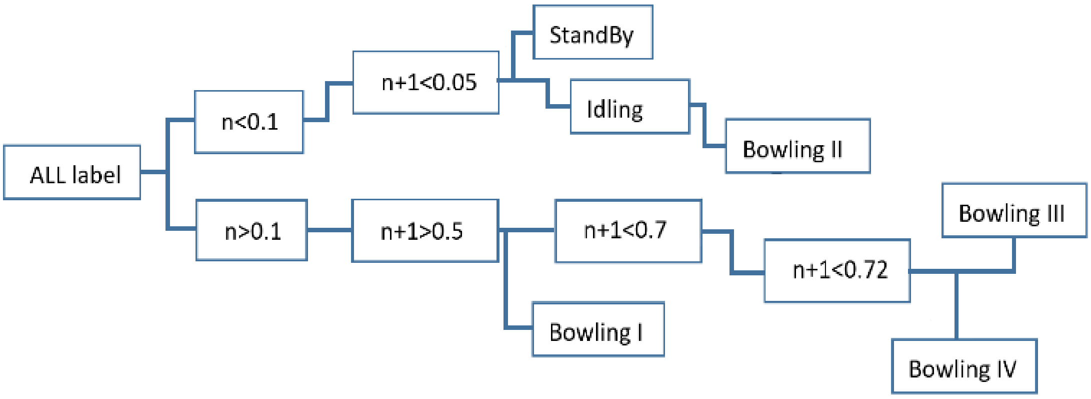


Figure 10. Decision tree branch.

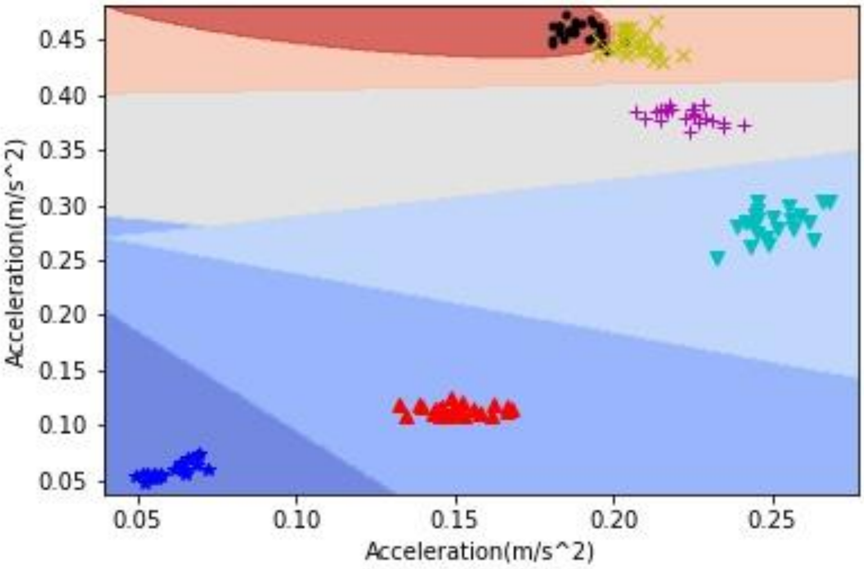


Figure 11. Support Vector Machine (SVM) Eigenspace.

Table 7. All classifier accuracies.

Methods	KNN	Adaboost	Naïve Bayes	Decision Tree	Random Forest	Non-Linear SVM
Accuracies	93%	96%	93%	90%	93%	98%

4.3. Maintenance Prediction Analysis

The grabber was driven by the gas pressure controlled by the solenoid valve in the blow molding machine. The condition of the O rings and pistons in the solenoid valve affected the motion of the grabber due to the lack of air pressure. If the sum of the opening or closing time was over 0.5 s, the machine was unable to initiate production. In this situation, the controller will halt the machine and generate an error signal. The data used in this experiment were the opening and closing times of the grabber. In total, 1000 opening and closing times from the historical production data saved in the server were used in this experiment. Figure 12 shows the results where X and Y represent the trait samples used to filter the same traits of samples and train the linear regression model. The formula Y = X × 0.43396226 + 0.04321132 is the result of the training. As Figure 13 shows, when the grabber opening time was 0.153 s, the measured closing time was 0.102 s and the predicted closing time was 0.11 s. This experiment proves that the model is able to predict the closing time of the grabber accurately. The proposed system will notify the manager to give the machine an inspection or maintenance under the conditions where the difference between the predicted and measured times is too large or both of these times are over 80% of the critical value.

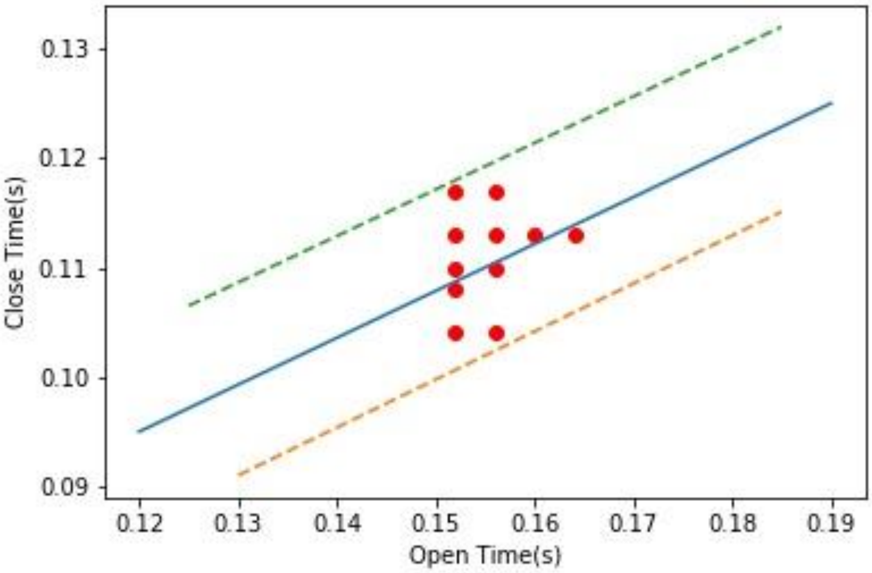


Figure 12. Linear regression model.

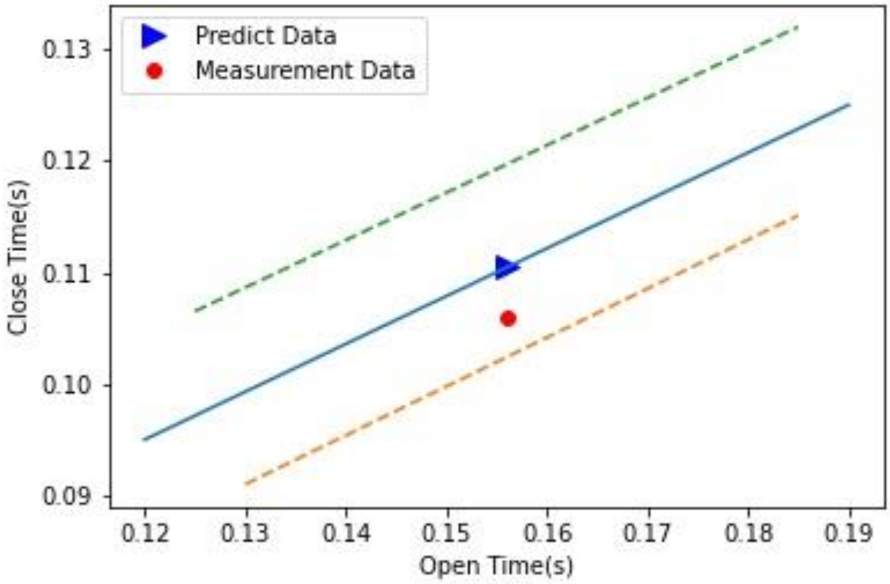


Figure 13. Prediction result of linear regression.

5. Conclusions

A smart machinery monitoring system with reduced information transmission and fault prediction methods using Industrial Internet of Things is proposed in this paper. The proposed system can effectively reduce 54.57% of the volume of each piece of data to be uploaded. The proposed system does not merely reduce the usage of internet bandwidth but also lets the monitoring platform operate in real-time. Moreover, the proposed system uses machine learning to classify the current machine's operation status with self-learning and self-correcting functions. The proposed system can effectively obtain a 98% prediction accuracy. Furthermore, the proposed system uses production data to predict when it is time to perform an inspection or maintenance on a machine, so it is able to decrease production and maintenance costs.

Author Contributions: Supervision, M.-F.T.; Writing—original draft, M.-F.T., Y.-C.C., M.-H.L. and L.-W.C.; Writing—review & editing, M.-F.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Chumpower Machinery Corporation, Taiwan and the APC was funded by National United University, Taiwan.

Data Availability Statement: The data presented in this study are available in Section 4.

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

References

1. Narayanan, A.; Sena, A.; Gutierrez-Rojas, D.; Melgarejo, D.; Hussain, H.; Ullah, M.; Bayhan, S.; Nardelli, P. Key Advances in Pervasive Edge Computing for Industrial Internet of Things in 5G and Beyond. *IEEE Access J.* **2020**, *8*, 206734–206754. [CrossRef]
2. Lin, C.; Su, X.; Yu, K.; Jian, B.; Yau, H. Inspection on Ball Bearing Malfunction by Chen-Lee Chaos System. *IEEE Access J.* **2020**, *8*, 28267–28275. [CrossRef]
3. Saucedo-Dorantes, J.; Delgado-Prieto, M.; Osornio-Rios, R.; Romero-Troncoso, R. Industrial Data-driven Monitoring Based on Incremental Learning Applied to the Detection of Novel Faults. *IEEE Trans. Ind. Inform. J.* **2020**. [CrossRef]
4. Chu, Y.; Pham, T.; Hsu, F.; Tuw, M.; Tan, C.; Chay, M.; Lim, S.; Tsai, M. An effective method for monitoring the vibration data of bearings to diagnose and minimize defects. In Proceedings of the IASED International Joint Conference on Information and Communication Engineering, Shanghai, China, 27–29 April 2018; pp. 1–6.
5. Peng, Y.; Qiao, W.; Qu, L.; Wang, J. Sensor Fault Detection and Isolation for a Wireless Sensor Network-Based Remote Wind Turbine Condition Monitoring System. *IEEE Trans. Ind. Appl. J.* **2017**, *54*, 1072–1079. [CrossRef]
6. Khademi, A.; Raji, F.; Sadeghi, M. IoT enabled vibration monitoring toward smart maintenance. In Proceedings of the IEEE International Conference on Internet of Things and Applications, Isfahan, Iran, 17–18 April 2019; pp. 1–6.

7. Muslewski1, L.; Pajak, M.; Grzadziela, A.; Musial, J. Analysis of Vibration Time Histories in the Time Domain for Propulsion Systems of Minesweepers. *J. Vibroengineering* **2015**, *17*, 1309–1316.
8. Asalapuram, V.; Khan, I.; Rao, K. A novel architecture for condition based machinery health monitoring on marine vessels using deep learning and edge computing. In Proceedings of the IEEE International Symposium on Measurement and Control in Robotics, Houston, TX, USA, 19–21 September 2019; pp. 1–6.
9. Yu, W.; Dillon, T.; Mostafa, F.; Rahayu, W.; Liu, Y. A global manufacturing big data ecosystem for fault detection in predictive maintenance. *IEEE Trans. Ind. Inform. J.* **2020**, *16*, 183–192. [CrossRef]
10. Kim, D.; Lee, E.; Qureshi, N. Peak-Load Forecasting for Small Industries: A Machine Learning Approach. *Sustainability* **2020**, *12*, 6539. [CrossRef]
11. Nayana, B.; Geethanjali, P. Analysis of Statistical Time-Domain Features Effectiveness in Identification of Bearing Faults from Vibration Signal. *IEEE Sens. J.* **2017**, *17*, 5618–5625. [CrossRef]
12. Michal, P. Fuzzy Identification of a Threat of the Inability State Occurrence. *J. Intell. Fuzzy Syst.* **2018**, *35*, 3593–3604.
13. Lin, Z.; Li, Y.; Liu, C.; Liao, I.; Hsieh, S.; Tsai, M. Machine maintenance management and repair prediction system. In Proceedings of the Mobile Computing Workshop, Tainan, Taiwan, China, 26–27 August 2018; pp. 1–3.
14. Borith, T.; Bakhit, S.; Nasridinov, A.; Yoo, K. Prediction of Machine Inactivation Status Using Statistical Feature Extraction and Machine Learning. *Appl. Sci.* **2020**, *10*, 7413. [CrossRef]
15. Saufi, S.; Ahmad, Z.; Leong, M.; Lim, M. Challenges and Opportunities of Deep Learning Models for Machinery Fault Detection and Diagnosis: A Review. *IEEE Access J.* **2019**, *7*, 122644–122662. [CrossRef]
16. Huang, S.; Huang, J.; Huang, C.; Chen, J.; Liao, I.; Hsieh, S.; Tsai, M. Equipment component recognition cloud platform with information security. In Proceedings of the Workshop on Consumer Electronics, Yunlin, Taiwan, China, 29 November 2019; pp. 1–3.
17. Gong, X.; Du, W.; Georgiadis, A.; Zhao, B. Identification of Multi-fault in Rotor-bearing System using Spectral Kurtosis and EEMD. *J. Vibroengineering* **2017**, *19*, 5036–5046. [CrossRef]
18. Yu, F.; Fan, F.; Wang, S.; Zhou, F. Transform-domain Sparse Representation Based Classification for Machinery Vibration Signals. *J. Vibroengineering* **2018**, *20*, 979–987.
19. Pajak, M. Identification of the Operating Parameters of a Complex Technical System Important from the Operational Potential Point of View. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2018**, *232*, 62–78. [CrossRef]
20. Angelopoulos, A.; Michailidis, E.; Nomikos, N.; Trakadas, P.; Hatziefremidis, A.; Voliotis, S.; Zahariadis, T. Tackling Faults in the Industry 4.0 Era—A Survey of Machine-Learning Solutions and Key Aspects. *Sensors* **2020**, *20*, 109. [CrossRef] [PubMed]
21. Guo, R.; Zhao, Z.; Huo, S.; Jin, Z.; Zhao, J.; Gao, D. Research on State Recognition and Failure Prediction of Axial Piston Pump based on Performance Degradation Data. *Process. J.* **2020**, *8*, 609. [CrossRef]