Predictive Edge Computing for Time Series of Industrial IoT and Large Scale Critical Infrastructure based on Open-source Software Analytic of Big Data

Emmanuel Oyekanlu
Dept. of Electrical & Computer Engineering
Drexel University
Philadelphia, Pennsylvania, USA
Email: eao48@drexel.edu

***Abstract*—The Industrial Internet of Things (IIoT) is quite different from the general IoT in terms of latency, bandwidth, cost, security and connectivity. Most existing IoT platforms are designed for general IoT needs, and thus cannot handle the specificities of IIoT. With the anticipated big data generation in IIoT, an open source platform capable of minimizing the amount of data being sent from the edge and at the same time, that can effectively monitor and communicate the condition of the large-scale engineering system by doing efficient real-time edge analytics is sorely needed. In this work, an industrial machine condition-monitoring open-source software database, equipped with a dictionary and small enough to fit into the memory of edge data-analytic devices is created. The database-dictionary system will prevent excessive industrial and smart grid machine data from being sent to the cloud since only fault report and requisite recommendations, sourced from the edge dictionary and database will be sent. An open source software (Python SQLite) situated on Linux operating system is used to create the edge database and the dictionary so that inter-platform portability will be achieved and most IIoT machines will be able to use the platform. Statistical analysis at the network edge using well known industrial methods such as kurtosis and skewness reveal significant differences between generated machine signal and reference signal. This database-dictionary approach is a new paradigm since it is different from legacy methods in which databases are situated only in the cloud with huge memory and servers. The open source deployment will also help to satisfy the criteria of Industrial IoT Consortium and the Open Fog Architecture.**

***Index Terms*— Databases, prediction, statistics, Industrial Internet of Things, Time series analysis**

I. INTRODUCTION

In recent years, much attempt is ongoing at creating Internet-of-Things (IoT) platforms capable of handling the real-time demands of IoT big data generation. Many existing platforms are vertical in nature since they are targeted to cater to the needs of specific IoT systems [1]. Thus, the quest for an all-encompassing IoT platform capable of satisfying the needs of all IoT systems continues. In the case of IIoT, efforts are also ongoing to create IIoT platforms that will satisfy the constraints of IIoT huge data generation and real-time decision making. The IIoT is quite different from the general IoT in terms of communication bandwidth needed to handle big data transmission in real-time, with reduced cost, improved latency and robust connectivity, such that real-time decisions that will result to efficiency, safety and stability of large scale IIoT and other engineering systems will be achieved [4]. For an effective IIoT platform, the need to handle big data timely and efficiently must be satisfied since IIoT systems always include critical national infrastructures such as the smart grid [5], [6], and other systems on which human safety and economic stability rely. Consider the case of the Boeing 787 shown in Fig. 1 and the industrial machine gear systems shown in Fig. 2.

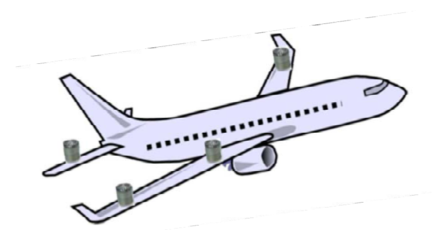


Fig. 1. Aircraft engines such as in Boeing 787 generates about 1 Terabyte of sensor data every 24 hours

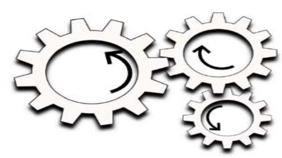


Fig. 2 Industrial machines and their parts such as gear system generates huge amount of vibration data every second.

Huge data sets of the order of terabytes are generated by sensors, arrayed on the aircraft fuselage, and used in studying the effect of airframe noise [2] on flight stability of the Boeing 787 and other similar aircrafts. Also, in industrial settings, huge

The author would like to thank Schweitzer Engineering Laboratory (SEL), Pullman, Washington, USA, for the donation of equipment that aided this research.

978-1-5386-2715-0/17/$31.00 ©2017 IEEE

data sets are generated every second by machine parts such as the gear systems and these data sets must be efficiently handled and analyzed in real-time so that crucial decisions involving safety of machine and human lives such as in the case of the NASA Max Launch Abort System (MLAS) shown in Fig. 3 are taken. The MLAS is used in ensuring timely ejection of space mission personnel from space-bound stricken spacecrafts.

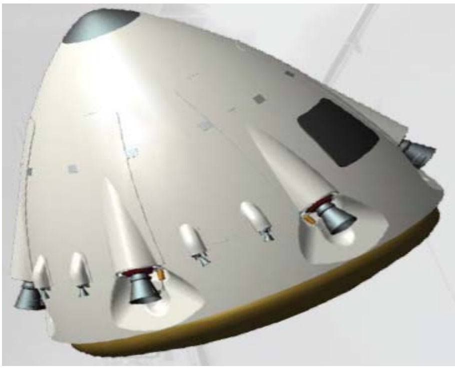


Fig. 3. Accurate big data analytics of large scale engineering projects like the NASA MLAS is crucial to human safety in space mission projects [3]

From the foregoing, it could be inferred that, for IIoT, safety of costly equipment, large scale critical infrastructures and personnel lives depends heavily on efficient data analytics. To satisfy the need for efficient real-time data analytics for IIoT and IoT in general, much of data analytics crucial to timely decision making will be done at the edge of the network, closer to where the actual system data is being generated [4], [11].

Thus, in this work, an open source database, small enough to fit into the memory of edge analytic device is designed using a light-weight database software that is not constrained by the need for a server, client, password schemes and all other needs synonymous with traditional database systems. The aim of having such database system close to the edge is not really for storage of data, but to effectively support real-time computing for IIoT systems such that very minimal reports about system conditions will be sent to the cloud always. Thus, with the presence of such lightweight, low-memory footprint database system at the edge, the reliability of the entire IIoT system will be amplified. Other advantages include overcoming network latency problem and bandwidth limitation etc. Other benefits of having an edge database to aid edge computing in IoT systems is listed in [7], however, the authors provide no direct method of implementation. In [8], authors proposed a light-weight, distributed, predictive platform needed to support efficient aggregation at the edge on IoT systems, but this system does not provide much clarity on how the proposed system may support reduced data traffic to the cloud. In [9], authors proposed a database system that will ensure data replication enabling delivery of dynamic content but the database is not small enough to be warehoused in the edge device. In [10], authors proposed a mobile edge computing solution to the workload placement problem, but the database system in this solution paradigm remains in the cloud. The present work is thus a pioneering attempt at placing a very lightweight database system in the edge of an IIoT system. The designed database is light enough to fit into the memory of generic edge devices. The computation work required to interface with the database and thus utilize its benefits in condition monitoring for IIoT machines is not greater than the computation work required for generic day-to-day work of normal edge devices. The contribution of this work is as stated:

- Design of a lightweight database for edge devices. The database is small enough to fit into generic edge devices, and its function is to aid real-time edge computing and to reduce the amount of data being sent to the middle fog layer and onward into the cloud.
- The database is designed with open source software (SQLite) and tested on a widely deployed open source operating system (Ubuntu 14.04). The SQLite is the most widely deployed mobile node, SQL database engine in the whole world [12], [13]. Due to its ubiquitous deployment, all IIoT and IoT platforms will be able to use the edge database for condition monitoring of IIoT machines. This database thus satisfies the semantic demands of standard bodies such as the Industrial Internet Consortium and Open Fog Architecture [14], [15].
- The combination of IIoT machine condition monitoring and the designed database dictionary uses well known semantics for industrial machine vibration interpretation and rotating machinery analysis such as those in use in the Type 2515 Vibration Analyzer and the IBM Type 7616 Application Software [16]. Thus, the combination of IIoT edge condition monitoring analytics and the designed dictionary-database will be able to satisfy heavy industrial and most IIoT machine needs.

The entire contribution of this work is shown in Fig. 4. The created edge dictionary and lightweight SQLite database will work together in the IIoT edge device to provide needed analytics for condition monitoring of industrial machines at the IIoT edge. Time-series waveforms of industrial machines when the machine is healthy will be captured and stored in the edge analytic device. The time-series of the machine when it is running will also be continually collected and windowed (using edge digital signal processor (DSP), e.g. Texas Instruments' TMS320C2000 C28x), such that the length of the time series will be the same. Both time series will be compared at specific time intervals by doing suitable statistical analysis including correlation and time series magnitude analysis such as crest factor, variance, mean, kurtosis and skewness at the IIoT edge. In case, when the length of the reference time series and the collected machine time series are different, then cross-correlation of both time-series will be a suitable analytical tool. When the vibration magnitude is beyond a certain crest factor, an alarm signal, and a recommendation from the database dictionary corresponding to the reference machine waveform will be sent to the cloud.

The remaining parts of paper is organized as follows. Section 2 discusses condition monitoring for industrial machines and large scale critical infrastructures such as the smart grid. Section 3 is a detailed discussion of common rotating machine problems and presentation of time series

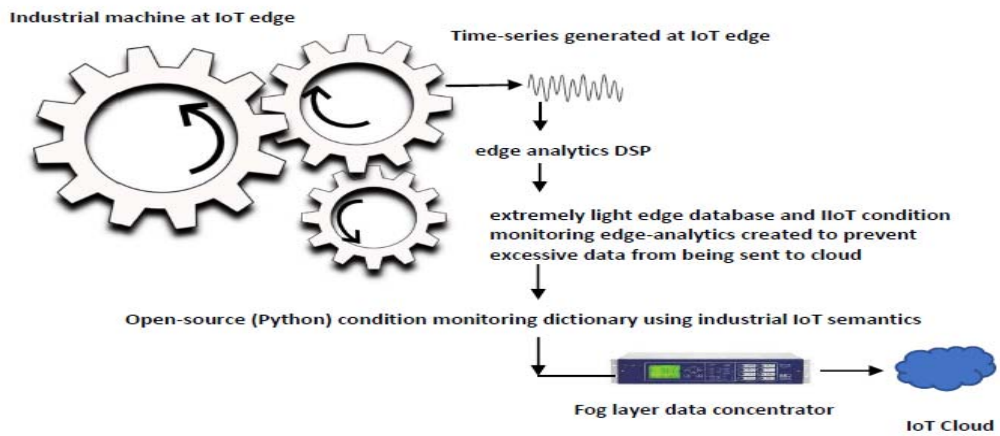


Fig. 4. Framework for the design of IIoT edge condition monitoring database and dictionary

waveforms of some of the problems. Section 4 is detailed discussion concerning the design of open-source edge analytics and condition monitoring database. Section 5 is a discussion of statistical measures suitable for condition monitoring at the Internet for industrial machines. Section 6 is conclusion of the paper.

II. CONDITION MONITORING FOR CRITICAL INFRASTRUCTURES AND IIOT MACHINES

Rotating IIoT machines are always part of critical infrastructures such as the electric power system. Failure of such machines are always catastrophic since repair could be expensive and time-consuming. Thus, condition monitoring of IIoT machines is always crucial since it prevents excessive downtime, reduces maintenance cost, prevent total loss of expensive machines, and it also could lead to improved safety for operating personnel. Condition monitoring involves obtaining signals from the rotating machine while the machine is in service and it allows for detection of faults, possible faults and the prediction of machine failures. Most industrial machines time series are either stationary random, stationary deterministic or transient signal types [18]. The time-series of IIoT machines or their frequency spectra are always obtained when the machine is in good condition and such signals are used as reference signals for comparison with subsequently obtained machine signals to identify the source of machine faults. In addition, such comparison could give an earlier indication of developing fault and could also serve as a prediction of the expected failure time of the machine. Popular rotating machines condition monitoring methods include time domain analysis, frequency domain, magnitude and wavelet analysis. Table I show details of some machine problems that may be detected by some of these condition monitoring methods. In addition to the condition monitoring methods shown in Table I, the use of a method known as order tracking is common in condition monitoring for industrial machines. Many vibrations and machine faults are directly related to the speed or rotation per minute (RPM) of the machine.

Table I: Common IIoT machine problems that can be detected by condition monitoring at the IIoT edge [18]

Machine/Machine Parts	Faults
Rotors & Shafts	Bent shaft, loose components, unbalance, misalignment, eccentric journals, Rubs etc.
Electric Machines	Broken rotor bars, variations in air gap geometry,
Gears	Misalignment, broken/worn teeth, eccentric gears
Journal Bearings	Rub, Oil whirl
Flexible couplings	Unbalance, misalignment etc.

Order tracking is a method used to normalize the time-series and the frequency domain fast Fourier transform (FFT) of the time series to the rotating speeds so that the signals will be easily related to the running speeds [17], [19]. The rotating speed of the machine is called order 1 (1X) while a fault that is known to occur at two times the rotating speed of the machine will be said to occur at order 2 (2X). For example, for a time-series that is analyzed using order tracking, bent shaft problem can be identified and separated from unbalance machine problem if a large component is occurring at 2 x rpm. Unbalance machine problem always occur at 1 x rpm [20]. This is illustrated in Fig. 5(a) and bent shaft problem which normally occur at order 2 is shown in Fig. 5(b). Using the combination of time domain magnitude analysis, frequency domain magnitude and position, and order tracking, several machine problems can be identified, and relevant solutions to the problem is provided. Some machine problems can only be identified when the machine is made to start and run up to full speed or coast-down from full speed during which the time series is obtained and analyzed. A common method to observe machine condition in this way is to plot the waterfall diagram of the machine. An example of this is shown in Fig. 5(c).

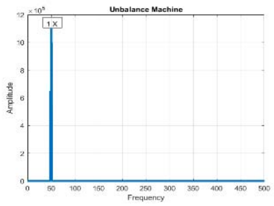


Fig. 5(a)

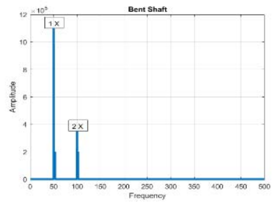


Fig. 5(b)

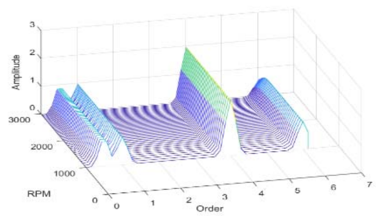


Fig. 5(c)

Fig. 5(a). Unbalance machine problem occurs at order 1; Fig. 5(b). Bent shaft problem occur at order 2; Fig. 5(c). Waterfall plot of a machine with 4 orders

In Fig. 5(c), the machine has four orders signifying four vibrating components and it is being tested by coasting up from 15 RPM to 50 RPM.

III. COMMON VIBRATION PROBLEMS FOR IIOT AND ANALYSIS OF THEIR WAVEFORMS.

General behaviors of IIoT machines, their vibration patterns, their stability modes and time-series that they generate when they are operating acceptably are well known and reported in literature. Also, their time series at the onset of fault and when a fault condition reaches an un-acceptable level that could lead to total loss of the machine are well known and reported in literature as well. Hence, these vibration patterns could be used at the IIoT edge to aid edge analytics and condition monitoring of industrial machines [17]-[24]. Also, since the time series of machines when they are operating acceptably are well known, these time-series could be obtained and used as reference signals for edge analytics. For a machine at the edge of the IIoT system, its time series could be obtained and stored in memory at the edge as reference time series. Condition monitoring for such machine will then include capturing the vibration signal of the rotating machine in time, and comparing such signal with the reference signal. If a certain predetermined threshold is breached when the reference signal and obtained signal are compared, then a concise alert signal using acceptable industrial semantics will be sent to the cloud so that appropriate action could be taken. In this work, the concise alert signal that will be sent to the cloud for appropriate action, is stored in a lightweight database inside the edge analytic device, situated close to where the machine is, at the edge of the IIoT. Some well-known IIoT machine time series, frequency domain and wavelet transforms that could be used as reference signals are shown in Fig. 6 to Fig. 8.

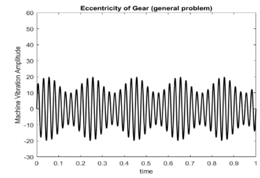


Fig. 6(a)

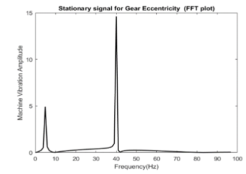


Fig. 6(b)

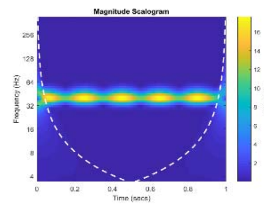


Fig. 6(c)

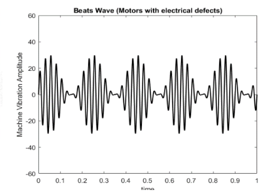


Fig. 6(d)

Fig. 6(a). Time series for gear eccentricity problem; Fig. 6 (b). Frequency domain (FFT) transform for gear eccentricity time series. Fig. 6 (c). Wavelet transform for gear eccentricity time series, Fig. 6 (d). Time series of motor with electrical defects

Fig. 7(a)

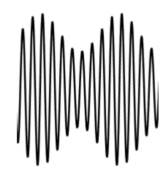


Fig. 7(b)

Fig.7. Difference between gear eccentricity (Fig. 7 (a)) and beats (Fig. 7(b)) time series

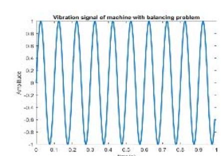


Fig. 8(a)

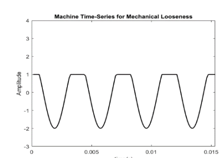


Fig. 8(b)

Fig. 8(a). Time series of IIoT machine with balancing problem (b). Time series of IIoT machine experiencing mechanical looseness

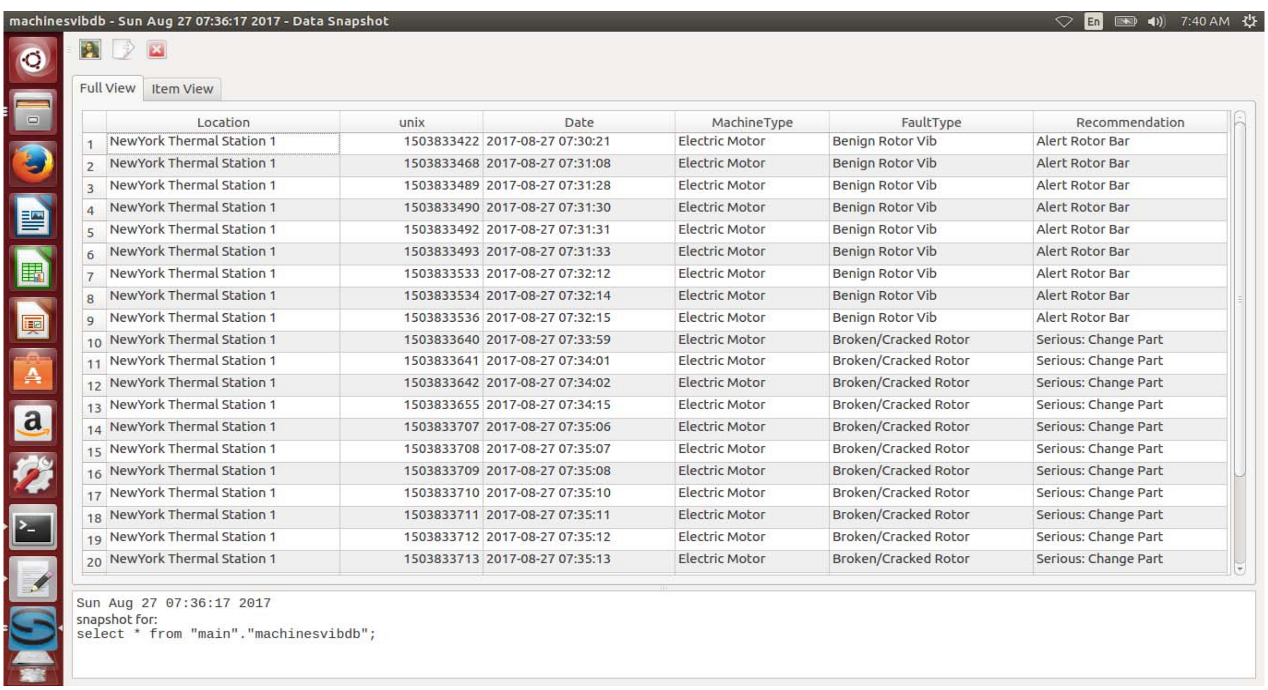
The time series shown in Fig. 6 to Fig.8 are selected to represent a broad range of machine vibration signals from different IIoT industries, but with emphasis on power generating (smart grid) industry. Popular examples of such machines include electric motor (A.C), turbines, generators, DC motors [25] etc. In Fig. 7, the difference between the time series of Fig. 6 (a) which represent waveforms of machine that is experiencing eccentricity of gear problem and the waveform of Fig. 6(d), which represent time series of motors with electrical defects [20] is explained. Close inspection of the both signals shown in Fig. 7 reveals that there is complete phase reversal at the area of minimum amplitude in Fig. 6(a), whereas this does not happen in Fig. 6(d) [20], [22], [23].

VI. DESIGN OF OPEN SOURCE LIGHTWEIGHT EDGE ANALYTIC DATABASE FOR IIOT.

The database software used in this work is the SQLite. It is a light version of SQL and it is the most widely deployed mobile node, SQL database engine in the whole world [12], [13]. Unlike SQL, the SQLite database can be created without creating a server, client, permissions, users, passwords and other database scheme and structures synonymous with the conventional SQL database system. In fact, the entire, database created with SQLite can be in a single file, much like just a plain text document. SQLite have the benefit of being a database but with lightness of a flat file. The entire database in SQLite is a single file, with tables containing data. It is also quite efficient to use because, if there is need to update the SQLite database, the user only need to update the needed part of the database. The main difference between an SQLite database and a flat file document is that with a flat file, the user has to pull the entire flat file from memory and iterate thru the entire document in order to locate a data point. SQLite is also open source, and it could be programmed with Python software. In this work, SQLite version 3 is used to show the idea of a light database system that is small enough to fit into the memory of any edge analytic device. The database is constructed using Python 2.7 on Ubuntu 14.04 operating, also selected since it is open source and widely deployed. The open source nature of both SQLite and the operating system satisfies the objectives of both Industrial Internet Consortium and the Open Fog Reference Architecture for Fog Computing standards for edge analytic devices which include the provision that any edge analytic solution for deployment for IIoT must:

- be compatible with Open Platform Communication Unified Architecture
- support Data Distributed Service (DDS), in which both a database and a data-bus implement the data centric abstraction.
- support Distributed Data Interoperability and management.

The open source nature of the developed database and on open source and widely available operating system makes it compatible with these standards. The developed database that simulates the database that may be available on an edge analytic device is shown in Fig. 9. It simulates the database in an edge analytic device being used to monitor an electric motor in an electric power thermal station in New York. As could be observed in Fig. 9, the recommendation column of the database is a python dictionary tuple which details the state of the electric motor that is being monitored and a recommendation on what should be done to prevent the collapse of the machine. For example, row 12, shows the location of the thermal station, the date and time of the event being monitored, the fault type and the final recommendation. The recommendation is based on acceptable industrial standards for monitoring rotating machines. The industrial standard is shown in Fig. 10 and could be obtained from [24]. It could be observed that by following acceptable industrial semantics, the edge analytic device will on only send an alert signal when the electric machine vibration is benign. This is observable in row 9 of the database. The signal and recommendation from the database dictionary will only be sent after the new machine time series is obtained and compared with a reference time series as



	Location	unix	Date	MachineType	FaultType	Recommendation
1	NewYork Thermal Station 1	1503833422	2017-08-27 07:30:21	Electric Motor	Benign Rotor Vib	Alert Rotor Bar
2	NewYork Thermal Station 1	1503833468	2017-08-27 07:31:08	Electric Motor	Benign Rotor Vib	Alert Rotor Bar
3	NewYork Thermal Station 1	1503833489	2017-08-27 07:31:28	Electric Motor	Benign Rotor Vib	Alert Rotor Bar
4	NewYork Thermal Station 1	1503833490	2017-08-27 07:31:30	Electric Motor	Benign Rotor Vib	Alert Rotor Bar
5	NewYork Thermal Station 1	1503833492	2017-08-27 07:31:31	Electric Motor	Benign Rotor Vib	Alert Rotor Bar
6	NewYork Thermal Station 1	1503833493	2017-08-27 07:31:33	Electric Motor	Benign Rotor Vib	Alert Rotor Bar
7	NewYork Thermal Station 1	1503833533	2017-08-27 07:32:12	Electric Motor	Benign Rotor Vib	Alert Rotor Bar
8	NewYork Thermal Station 1	1503833534	2017-08-27 07:32:14	Electric Motor	Benign Rotor Vib	Alert Rotor Bar
9	NewYork Thermal Station 1	1503833536	2017-08-27 07:32:15	Electric Motor	Benign Rotor Vib	Alert Rotor Bar
10	NewYork Thermal Station 1	1503833640	2017-08-27 07:33:59	Electric Motor	Broken/Cracked Rotor	Serious: Change Part
11	NewYork Thermal Station 1	1503833641	2017-08-27 07:34:01	Electric Motor	Broken/Cracked Rotor	Serious: Change Part
12	NewYork Thermal Station 1	1503833642	2017-08-27 07:34:02	Electric Motor	Broken/Cracked Rotor	Serious: Change Part
13	NewYork Thermal Station 1	1503833655	2017-08-27 07:34:15	Electric Motor	Broken/Cracked Rotor	Serious: Change Part
14	NewYork Thermal Station 1	1503833707	2017-08-27 07:35:06	Electric Motor	Broken/Cracked Rotor	Serious: Change Part
15	NewYork Thermal Station 1	1503833708	2017-08-27 07:35:07	Electric Motor	Broken/Cracked Rotor	Serious: Change Part
16	NewYork Thermal Station 1	1503833709	2017-08-27 07:35:08	Electric Motor	Broken/Cracked Rotor	Serious: Change Part
17	NewYork Thermal Station 1	1503833710	2017-08-27 07:35:10	Electric Motor	Broken/Cracked Rotor	Serious: Change Part
18	NewYork Thermal Station 1	1503833711	2017-08-27 07:35:11	Electric Motor	Broken/Cracked Rotor	Serious: Change Part
19	NewYork Thermal Station 1	1503833712	2017-08-27 07:35:12	Electric Motor	Broken/Cracked Rotor	Serious: Change Part
20	NewYork Thermal Station 1	1503833713	2017-08-27 07:35:13	Electric Motor	Broken/Cracked Rotor	Serious: Change Part

Fig. 9. Edge analytic database for monitoring, prediction and recommendation for IIoT machines

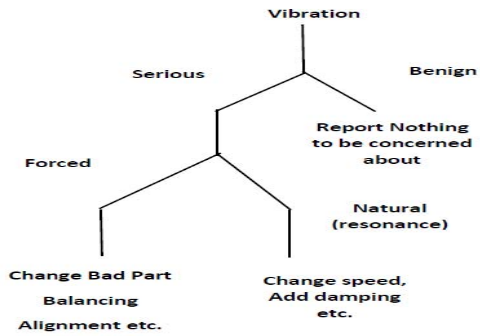


Fig. 10. Industrial machine vibration flowchart[24]

explained earlier. The use of the database will completely prevent sending the obtained time series signal, or any other redundant data. Only the alert signal and recommendation will be sent to the fog layer and onward to the cloud for appropriate action that will prevent total loss of the machine.

V. STATISTICAL ANALYSIS FOR CONDITION MONITORING FOR IIOT EDGE ANALYTICS

For IIoT edge condition monitoring based on the edge database to work as intended, the edge analytic device must be able to correctly compare the obtained time series with a reference time series in memory. Appropriate statistical analysis tool must be available at the edge through which the edge analytic device will do the needed computation. Appropriate analytical tools that are not compute-intensive and that can be used at the edge are provided in [18] and they are magnitude analysis, frequency domain and wavelet analytics. An electric motor reference time series signals (Fig. 11(a)) generated using open-source Python 2.7, stored in the memory of the edge-analytic device is used to show the example of edge analytics needed for condition monitoring and prediction for time series of the industrial machine at the edge of the IIoT. Fig. 11(b) is used as the newly captured time series by the edge analytic device. It is observable that the magnitude of the newly captured time series in Fig. 11 (b) is lower than the magnitude of the reference time series (Fig. 11 (a)) stored in the memorry of the edge analytic device.

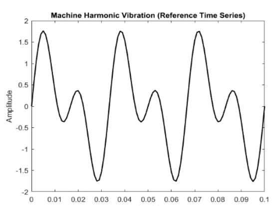


Fig. 11 (a)

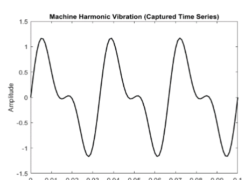


Fig. 11 (b)

Fig. 11 (a). Reference time series of electric motor vibration ; Fig. 11(b).Newly ingested time series of electric motor vibration

Appropriate statistical analysis in the stance wiill include only magnitde analysis which in this case is kurtosis, skewness, root mean square and crest factors. The equations used in comparing the reference with the newly obtained time series is given in equation (1) - (4). In (1) to (4), μ is the mean of the time series, σ is the standard deviation, and E(t) is the expected value [37], [38], [39]. The result of the edge analytics using equations (1) – (4) is given in Table II. It is observable in Table II that any of the magnitude analysis tool chosen for comparing the reference time series to a newly acquired time series at the

$$\text{Kurtosis} = E\left[\left(\frac{x-\mu}{\sigma}\right)^4\right] = \frac{\mu^4}{\sigma^4} = \frac{E[(x-\mu)^4]}{(E[(x-\mu^2)])^2} \quad (1)$$

$$\text{Skewness} = E\left[\left(\frac{x-\mu}{\sigma}\right)^3\right] = \frac{\mu^3}{\sigma^3} = \frac{E[(x-\mu)^3]}{(E[(x-\mu^2)])^{\frac{3}{2}}} \quad (2)$$

$$X_{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}|X_n|^2} \quad (3)$$

$$\text{Crest-factor} = \frac{||X||_\infty}{\sqrt{\frac{1}{N}\sum_{n=1}^{N}|X_n|^2}} \quad (4)$$

IIoT edge will work since any of the statistical equation used show clear difference between the reference time series and the new machine time series. After this statistical computation, the edge analytical device will go into the database dictionary to select the appropriate recommendation based on the threshold of the difference between the reference and the acquired signal. The selected recommendation will be sent to the cloud. As stated earlier, if only recommendation are sent to the cloud, it will relieve the network from congestion in bandwidth-constrained networks such as IIoT networks and similar networks described in [28] - [31]. An approximate data efficiency of this system as related to data storage in the cloud system can be established by comparing the ratio of the total possible data that can be sent to the cloud in a particular time frame to the recommendation data sent to the cloud using [35]

$$compression\ ratio = \frac{size\ of\ uncompressed\ data}{size\ of\ compressed\ data} \quad (5)$$

In the edge database shown in Fig. 9, using (5), an approximate data compression savings of about 45% will be achieved if only rows 10 to 20, with ‘serious:change part’ recommendations are sent to the cloud instead of sending all data rows from 1-20.

VI. CONCLUSION

In this work, a small and lightweight open-source database capable of fitting into the memory of IIoT edge analytic devices is designed. The database could be used to send prediction alerts about the condition of the machine being monitored to the IIoT cloud. The machine time series when the machine is healthy is obtained and stored as the reference signal. Newly acquired machine times series is compared with

Table II: Statistical analysis to aid machine condition monitoring and predition at IIoT edge

	Crest Factor	R.M.S	Extreme Value (Max)	Extreme Value (Min)	Mean	Variance	Skewness	Kurtosis
Reference Time Series	1.7689	0.9950	1.7601	-1.7601	- 7.898 E-17	1	3.5927E-16	2.2725
New Machine Time Series	1.6864	0.6939	1.1701	-1.1701	- 5.7644E-18	0.4868	3.702E-17	2.09 37

the reference using statistical equations to compute the difference between the magnitude of the machine time series and the reference time series. A predictive recommendation is made based on the result of the computation and only the recommendation obtained from the edge database is sent to the cloud. It is believed that when this new paradigm is fully deployed, it will can achieve acceptable condition monitoring for IIoT machine right at the edge of the network, and, also, capable of minimizing the amount of data sent to the IIoT cloud.

REFERENCES

[1] M. Floeck, A. Papageorgiou, A. Schuelke and J. Song, "Horizontal M2M Platforms Boost Vertical Industry: Effectiveness Study for Building Energy Management Systems", 2014 IEEE World Forum on IoT, Seoul, 2014, pp. 15-20

[2] S. Tamas, "Sound and Vibration Technologies and Applications", White Paper, National Instruments, 2016

[3] NASA, "The NASA Engineering and Safety Center Technical Update", NASA/TM-2014-218505, 2014.

[4] K. Karachalios, "A Vision for the Next Wave of Connectedness", IEEE GSMA PSMC, Barcelona, February 2017

[5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, *Edge Computing: Vision and Challenges*, IEEE Journal of Internet of Things, Vol.3, No. 5, Oct. 2016

[6] S. E. Bush, "Smart Grid Communication-Enabled Intelligence for the Electric Power Grid", IEEE Press, John Wiley, 2014.

[7] Cisco, "The Cisco Edge Analytics Fabric System: A New Approach for Enabling Hyper-Distributed Implementation", White Paper, Cisco, 2016.

[8] M. Satyanarayana, "The Emergence of Edge Computing", IEEE Computer Journal, IEEE, 2017.

[9] Y. Lin, B. Kemme, M. Patino-Martinez, R. Jimenez-Peris, "Enhancing Edge Computing with Database Replication", IEEE, 2007.

[10] S. Wang, M. Zafer, and K.K. Leung, "Online Placement of Multi Component Applications in Edge Computing Environments", in IEEE Access, Vol. 5, pp. 2514-2533, 2017.

[11] M. Satyanarayanan et'al, "Edge Analytics in the Internet of Things", IEEE, 2015.

[12] S. Mutti, E. Bacis, S. Paraboschi, "SeSQLite: Security Enhanced SQLite Mandatory Access Control for Android Databases", ACSAC 15, Los Angeles, CA, USA, December 2015.

[13] V. Matos, "Persistent SQL Databases", Cleveland State University, 2016.

[14] Industrial Internet Consortium, "The Industrial Internet of Things Vol G5: Connectivity Framework", Vol. 1.0, 2017

[15] Open Fog, "OpenFog Reference Architecture for Fog Computing", Online, available, www. OpenFogConsortium.org, February, 2017.

[16] G. Bates, "Application Notes: Vibration Diagnostics for Industrial Electric Motor Drives", Bruel and Kjaer, BO 0269-12.

[17] A. Brandt, "Noise and Vibration Analysis Signal Analysis and Experimental Procedures", John Wiley, 2010

[18] M. P. Norton, D.G. Karczub, "Fundamental of Noise and Vibration Analysis for Engineers", Cambridge Univ. Press, 2003.

[19] National Instruments, "Advanced Measurement Systems and Signal Processing, Case Studies: Wind Power", White Paper, National Instruments, online, available, www.ni.com; August, 2017

[20] J. Basrah, "Vibration Analysis and Diagnostic Guide", ResearchGate, December, 2016.

[21] IITG, "Vibration Based Condition Monitoring in Rotating Machines", online, available, *www.iitg.ernet.in/scifac/qip/public_html/cd_cell/chapters/.../rt_chapter 17.pdf*; August, 2017

[22] AZIMA, "" Examples of Some Waveforms and Their Spectra", online available, www.azimadli.com/vibman; 2017

[23] DLI, "Vibration Reference Guide Basic Vibration Theory", online, available, www.DLIengineering.com, August, 2017.

[24] V. Wowk, "" A Practical Approach to Solving Machine Vibration Problems", P/PM Technology, 1995.

[25] N. Bianchi, L. Alberti, M. Barcaro, "Design and Tests of a Four-Layer Fractional-Slot Interior Permanent-Magnet Motor", IEEE Trans. on Industrial Applications, Vol. 52, No. 3, June 2016.

[26] G. Choi, T. M. Jahns, "PM Synchronous Machine Drive Response to Asymmetrical Short-Circuit Faults", IEEE Trans. on Industrial Applications, Vol. 52, No. 3, June 2016.

[27] P. H. Westfall, "Kurtosis as Peakedness", Journal of the American Statistician, Vol. 68, Iss. 3, 2014

[28] J. Uddin, E. Oyekanlu, C. H. Kim, and J. M. Kim, "High Performance Computing for Large Graphs of Internet Applications using GPU", 2014.

[29] D. Shah. "Message-passing in Stochastic Processing Networks" Surveys in Operations Research and Management Science, 16(2):83–104, 2011.

[30] D. Shah, "Network Scheduling and Message Passing", in, Performance Modeling and Engineering, Editors Z. Liu and C. Xia, a collection of tutorials from ACM Sigmetrics/ Performance, 2008.

[31] D. Shah, J. N. Tsitsiklis, and Y. Young, "Qualitative Properties of α-Fair Policies in Bandwidth-Sharing Networks", The Annals of Applied Probability, 2014, vol. 24, No. 1, 76-113.

[32] S. Mittal, J.S. Vetter, "A Survey of Architectural Approaches for Data Compression in Cache and Main Memory Systems", IEEE Transaction on Parallel and Distributed Systems, Vol. 27, No. 5, May 2016.