Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 151 (2019) 607–613

Procedia Computer Science

www.elsevier.com/locate/procedia

The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40)
April 29 - May 2, 2019, Leuven, Belgium

IoT-based predictive maintenance for fleet management

Patrick Killeen[a,*], Bo Ding[b], Iluju Kiringa[a], Tet Yeap[a]

[a]University of Ottawa, 75 Laurier Ave E,Ottawa K1N 6N5, Canada
[b]Harbin University of Science and Technology, Harbin, China

Abstract

In recent years, the Internet of Things (IoT) and big data have been hot topics. With all this data being produced, new applications such as predictive maintenance are possible. Consensus self-organized models approach (COSMO) is an example of a predictive maintenance system for a fleet of public transport buses, which attempts to diagnose faulty buses that deviate from the rest of the bus fleet. The present work proposes a novel IoT architecture for predictive maintenance and proposes a semi-supervised machine learning algorithm that attempts to improve the sensor selection performed in COSMO. With the help of the Société de Transport de l'Outaouais, a minimally viable prototype of the architecture has been deployed and J1939 sensor data have been acquired.

Keywords: J1939;internet of things;predictive maintenance;sensor selection;fleet management;predictive analytics

1. Introduction

Internet of Things (IoT) is a new paradigm that is growing quickly. By 2020 many believe billions of devices will be connected to the internet [2][15]. Some of the applications of IoT are smart farming, smart transport, smart health, smart cities, smart homes, and smart grids [2]. With all these devices connected to the internet, big data becomes more prevalent. Modern data analytic algorithms struggle to process the massive amounts of data produced by these IoT devices. Predictive maintenance, an example of smart transportation, attempts to predict the health of equipment using machine learning. Predicting vehicle faults is not a trivial task. Labeled datasets of vehicle faults are rare compared to historical data of normal vehicle behavior because it is not financially viable to break equipment to gather fault data. Common prediction measures are equipment remaining useful life and the health status of the equipment.

The state of the art of predictive maintenance and related IoT architectures follow: [17] deployed a fleet-wide predictive maintenance system for a fleet of 19 buses and were able to detect buses that deviated from rest of fleet and to diagnose the deviations using a history of fault data. According to [10], MineFleet is a commercial predictive

* Patrick Killeen
E-mail address: pkill013@uottawa.ca

10.1016/j.procs.2019.04.184

maintenance system that gathers data from a vehicle using a gateway and sends aggregations of the signals to a backend server for analysis. [1] gathered OBD-II vehicle data over Bluetooth using an Android smart phone, which acted as the gateway for predictive analytics and data visualization. [6] proposed an IoT architecture using SenML. They used gateways to connect legacy devices to the internet. [20] used expert systems on the cloud to analyze vehicle data to help mechanics make decisions. [8] analyzed CAN-bus data of military vehicles to perform vehicle diagnostics. [4] proposed an IoT middleware for managing IoT devices, using gateways and software defined networks.

The present work involves an IoT architecture for a predictive maintenance fleet management system and fleet-wide data analytics, currently under development, that runs on a proposed architecture. The predictive maintenance architecture of this work is focused on public transport buses. This paper is structured as follows: section 2 provides a literature review on IoT and predictive maintenance; section 3 proposes an IoT architecture; section 4 proposes a semi-supervised sensor selection model for predictive maintenance, which is currently under development; section 5 mentions this work's minimally viable prototype and results; and section 6 concludes this work and mentions future work.

2. Background

IoT connects multiple devices [2], and the devices can sense and interact with the environment around them. A literature review of [11] revealed that IoT can be split into five layers: sensing, network, storage, learning, and application. The sensing layer gathers data from the environment and interacts with it using sensors and actuators. The network layer connects lower level nodes to the cloud/fog [2]. A company named Libelium is working on providing global wireless sensors network coverage [13]. The storage layer stores sensor data, aggregations, and other types of data. The learning layer performs data analytics on stored sensor data for knowledge discovery; for example, anomaly detection, or deviation detection, which attempts to detect when an instance deviates from its norm, can be performed in the learning layer. The application layer provides the interface to the IoT system by providing lower layer information access and control.

According to [21], there are four basic methods for detecting outliers: counting objects in k-neighborhoods, *k*th nearest neighbor distances, outlier scores, and comparing object neighbor to its neighbor. Outlier scores are useful, since in most deviation detection approaches binary labels ('*anomaly*' or '*normal*') are used, which does not enable comparing the level of deviation between two '*anomaly*' objects. Outlier detection is sensitive to dataset size, which can be tackled by dimension reduction techniques. At times it may be necessary to rank how much a point is deviating from the norm. Evaluating outlier detection algorithms may be necessary at times, which can be accomplished using a receiver operating characteristic (ROC) curve's area. An application of anomaly detection is vehicle prognostics; for example, finding a potentially faulty vehicle that deviates from the rest of its fleet.

Prognostics is the prediction of the health status of monitored equipment [3]. According to [14], there are three types of approaches for prognostics: model-based, expert-system, and data-driven . The model-based approach is difficult to implement for complex systems. It attempts to use mathematical models, which are highly coupled to the equipment and domain, to represent the behavior of the equipment. The expert-systems approach requires experts' knowledge to build the system. Such systems are built by defining failure modes and identifying degradation patterns of the equipment. Such a system is also tightly coupled to the equipment and domain. The data-driven approach mainly uses current and historical sensor data [8] and works well with complex systems [3]. Such a system is more applicable to many domains since it is purely data-driven and is not coupled to specific equipment. Predictive maintenance lowers costs by preventing failures, unscheduled maintenance, and downtime, and by ensuring the replacement of failing parts is done only when needed [12]. To lower these costs reliably, sensor selection must be conducted with care. Expert knowledge of critical components, data-driven entropy-based approaches, or a combination of both methods are examples of sensor selection strategies [12][14][17][3].

[17] implemented a predictive maintenance system for a bus fleet, named COSMO, which implements a data-driven sensor selection strategy. Some of their older work is found in [5], [16], and [18]. Their system has three phases: initialization, deviation detection, and fault diagnosis. They focus on working with less critical equipment (that is, affordable equipment that can/will break) to learn and acquire knowledge after the system goes into production and select sensors based on entropy and stability. They run experiments with autoencoders, histograms, and linear relations as their models. Their cloud server performs vehicle anomaly detection, and using a vehicle service record database

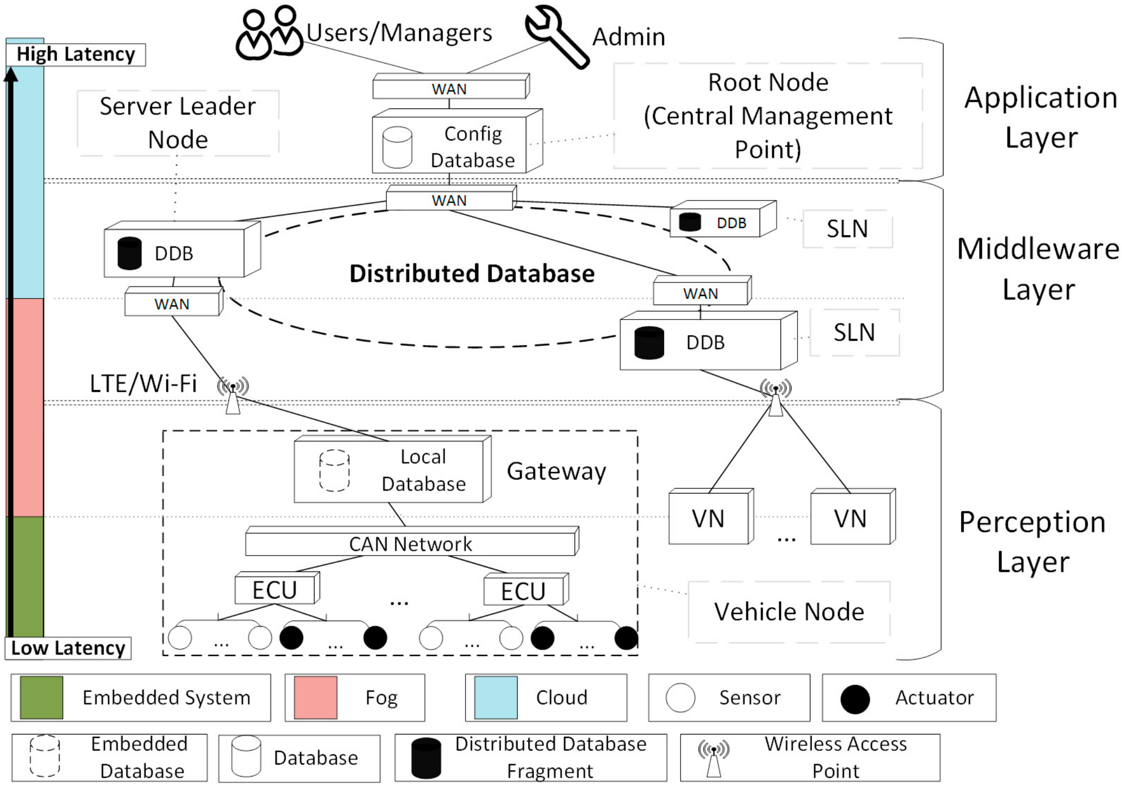

Fig. 1. Predictive maintenance fleet management system architecture-overview diagram.

(VSRDB), it performs repair diagnosis. However, the COSMO approach has a few drawbacks. In their experiments they used an ill-formatted VSRDB. In their results using the autoencoder model, there were repairs made when no deviations were detected. They suspect the repairs were done too early and believe it was not a no-fault-found (NFF) occurrence. They also mention that a repair may not have been linked to an actual fault. This suggests it is not clear if some faults actually occurred, which means a deviation may not be linked to an actual fault, or a NFF may have occurred. They also mention their sensor selection method is not guaranteed to select optimal sensors for fault detection.

3. Predictive maintenance system architecture

The present work proposes an IoT architecture designed to support fleet management. The architecture (see Figure 1) is inspired by the architecture proposed in [7]. It is divided into three layers (the perception layer, the middleware layer, and the application layer), which are flavors of the IoT layers mentioned in section 2. The perception layer abstracts the fog and embedded systems. It performs sensing, lightweight storage, networking, and machine learning. It provides the interface to low-level nodes. The middleware layer abstracts the fog and the cloud and generally performs more heavy-duty storage, networking, and machine learning compared to the perception layer. It provides the interface to perception-layer nodes. The application layer is similar to the IoT application layer mentioned in section 2.

Found in the perception layer, the vehicle node (VN) represents the vehicle. The VN has a J1939 network and a gateway. By reading J1939 traffic, the VN's gateway can perform sensor data acquisition, aggregation, and lightweight data analytics. Data is stored using an embedded database, since it is lightweight. The gateway interfaces the J1939 network to the fleet system. Each VN has a software administrative agent installed on the gateway, allowing administrators to remotely update the VN's software. Since the VN is mobile, it must have network connection via wireless technologies such as Wi-Fi and LTE. MQTT is used as the machine-to-machine protocol for VN-to-fleet communication.

The server leader node (SLN) is found in the middleware layer. It can be found in the fog (a resource-constrained cloud) or the cloud (see Figure 1). It is responsible for managing VNs of a geographical region. The SLN has a MQTT broker, which provides its VN fleet with the ability to communicate with each other and the SLN itself. Each SLN stores fragments of the fleet system's distributed database (DDB). A region's fleet data can be stored regionally in its SLN's DDB fragment, which provides data locality and therefore increases availability [19][11]. By design, the SLN can perform fleet-wide data analytics.

Found in the application layer, the root node (RN) is responsible for managing the entire fleet system. It is the fleet system's central access point, provides the interface for IoT applications, connects fleet nodes to the system, and enables fleet-wide administration. The RN has a MQTT broker installed to allow fleet system access requests to be made by fleet nodes, which use the RN's fleet configuration database. It has an admin server and admin agent installed, which allow administrators to remotely configure the software of any node in the system.

4. Semi-supervised sensor feature selection — ICOSMO

The present work attempts to improve the sensor feature selection performed in COSMO [17], by using a semi-supervised machine learning approach, which is currently under development. A few definitions are necessary.

a) **Sensor Class**: a J1939 sensor definition, defined as a PGN-SPN pair

b) **Sensor Instance**: a physical J1939 sensor, which may or may not be installed on a vehicle

c) **COSMO Sensor**: a sensor class that has been chosen as a selected feature in the unsupervised deviation detection model of the COSMO approach.

The algorithm proposed in this work is named *Improved Consensus self-organized models* (ICOSMO) and makes the following assumptions: a) VSRDB, with repair records that describe details about faults that were repaired, is accessible (this is not unreasonable, since [17] had access to such a database); b) the repair records in the VSRDB are associated with true faults (that is, the records detailing the fault and the steps taken to repair it are linked to an actual fault, not preventive repairs but instead reactive repairs); c) a document retrieval algorithm exists, which takes a mechanic's repair record as a query, searches its indexed J1939 specification document, and estimates sensor classes involved in the failing/faulty components that the repair fixed (a black box document retrieval algorithm (BBDRA) is used in this work to simulate this assumption); and d) all buses in a fleet are of the same model, and each bus shares similar daily travel routes.

In regard to the drawbacks of COSMO mentioned in section 2, [17] do not provide any evidence to support the absence of NFFs. Assuming NFFs occurred, the present work tries to reduce their occurrences. The following are the possible reasons that COSMO failed to detect the deviations: a) the unsupervised feature selection of COSMO during the initialization phase failed to choose the best sensors needed to detect the fault; b) the best sensors may not even be installed on the bus (or were not accessible by the J1939 logging software); and c) the sensor stream data distribution may have evolved on most vehicles and the machine learning model is out-of-date (that is, chosen sensors are no longer the best choice for anomaly detection). ICOSMO is designed in a data-driven fashion for conducting predictive maintenance, since mostly current and historical sensor data are accessible.

Using its black box document retrieval algorithm (BBDRA), ICOSMO dynamically adjusts COSMO sensors over time, by removing stale sensors and adding candidate sensors from/to the COSMO model, which is done by introducing the following metrics: sensor contribution (SC) and sensor potential contribution (SPC). ICOSMO is separated into three steps:

1. Find the sensor classes involved with a faulty component using the BBDRA.

2. Modify the SC and SPC of sensor instances based on their ability (or potential ability) of finding deviations.

3. Periodically re-organize the COSMO sensors by removing/adding sensor classes from/to COSMO's selected features by analyzing the SC and SPC of the majority of a sensor class' sensor instances.

To verify the performance of ICOSMO, simulations and modeling will be conducted by generating data using the STO J1939 data dumps acquired from the MVP. Experiments will be conducted to compare the performance of ICOSMO to a variety of flavors of COSMO, namely to a) standard COSMO, b) COSMO using a different deviation detection algorithm (instead of the most central pattern method), and c) COSMO without sensor selection (all sensors selected). A variety of metrics will be used to gain insight into each approach's performance, such as fscore, true positive rate, false positive rate, false negative rate, and accuracy. ROC curves will also be used. Once complete, assuming ICOSMO outperforms COSMO, research will be conducted to investigate how on-board expert systems can be combined with ICOSMO, removing the need for the VSRDB. The goal would be to design a fully-autonomous predictive maintenance system using fleet-wide and on-board data analytics.

5. Predictive maintenance system prototype

A minimally viable prototype (MVP) of the architecture mentioned in section 3 has been implemented and is running in a live environment. The MVP does not have ICOSMO (see section 4) implemented yet, since ICOSMO is still currently under development.

The MVP is targeted for creating an IoT predictive maintenance fleet management system for the public transport buses of the Société de Transport de l'Outaouais (STO), Gatineau, Canada. Each bus will have a gateway installed, which reads sensor data and performs lightweight analytics. The goal is to discover novelties and to provide this information to the fleet managers to help them make better maintenance decisions. With each bus equipped with a gateway, fleet-wide data analysis will be possible. This enables the possibility of discovering some novelties that would not be obtainable when only monitoring individual vehicles.

We had many meetings with the STO to gather requirements for the MVP. What we learned is the following: occasionally the 24V bus batteries would fluctuate in voltage. We, therefore, needed an uninterruptible power supply (UPS). A waterproof casing was also required to avoid water damage. The buses used the SAE's J1939 protocol for heavy-duty machinery.

From the requirements, we created the MVP. We installed our hardware in a waterproof plastic container containing the following hardware (see Figure 2): a Raspberry Pi (a mini computer which runs the Linux operating system Raspbian), a UPS PIco, an Adafruit FONA 800 for global system for mobile communications (GSM) network connection, a USB Wi-Fi antenna, a 24V-to-5V converter for power, and a Copperhill J1939 electronic control unit (ECU). Furthermore, the administrative software chosen was MeshCentral, an open source software package for managing a private network of machines.

We installed the following (see Figure 3) at the STO garage: a) our gateway onto a hybrid bus; b) a SLN (laptop) for J1939 data network storage; and c) a Huawei E8372 Turbo stick, which is a USB Wi-Fi modem with a LTE data plan, in the STO's garage near the fuel station. This setup allows any of our future equipment to have internet connection when near the fuel station. It also provides the ability to dump (over Wi-Fi) massive amounts of data to our SLN in the garage without going through the limited-bandwidth LTE data-plan for cloud data storage. The RN in the MVP is *meshcentral.com*, a website that hosts administration servers.

Since deployment, the MVP allowed us to acquire J1939 data dumps daily. Approximately 1 GB of uncompressed J1939 data (200 MB when compressed) was acquired daily. This data was stored on the SLN at the STO garage. With the help of the J1939 specification document, there were various data types obtained by analyzing the J1939 data dumps, such as sensor data, the status of equipment, J1939 network routing information, proprietary data, diagnostic trouble codes, etc. Noteworthy sensor readings found in the data dump are listed below:

- wheel speed information: axle angle and relative speed
- vehicle distance information: trip distance, total distance, and remaining distance before running out of fuel
- driver pedal positions
- engine information: speed, torque, and temperature
  - oil information: temperature, pressure, and level
  - coolant information: temperature, pressure, and level
- transmission fluid information: oil temperature and pressure

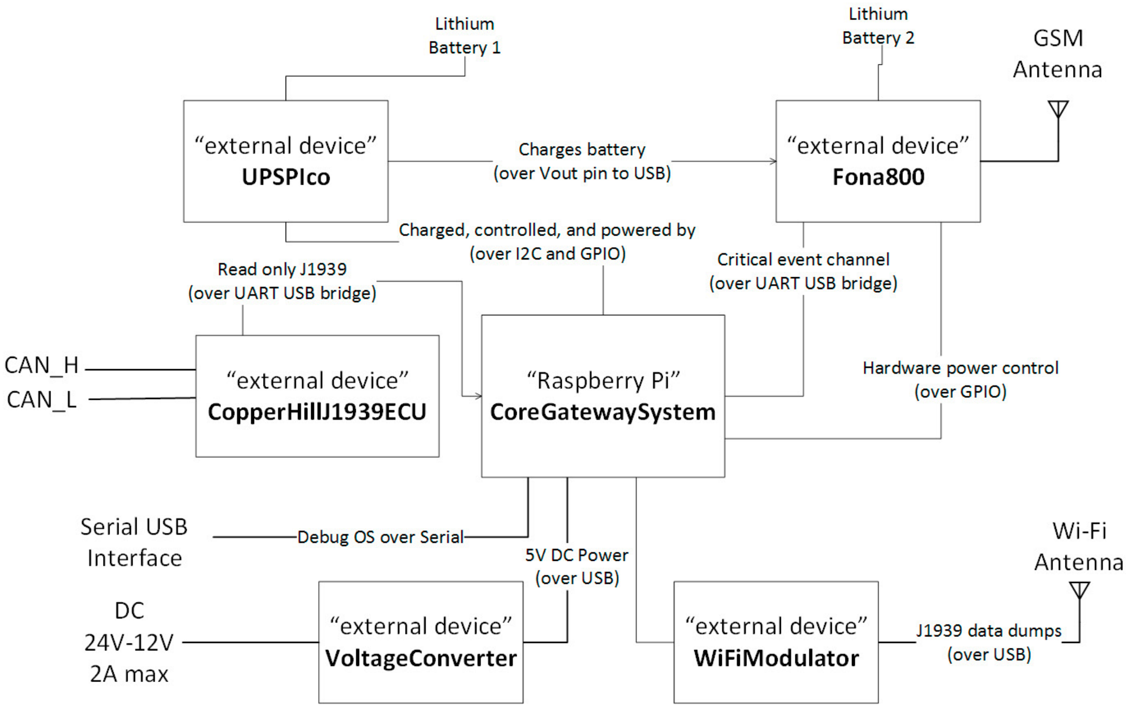

Fig. 2. Hardware SysML[9]-style diagram of gateway in prototype.

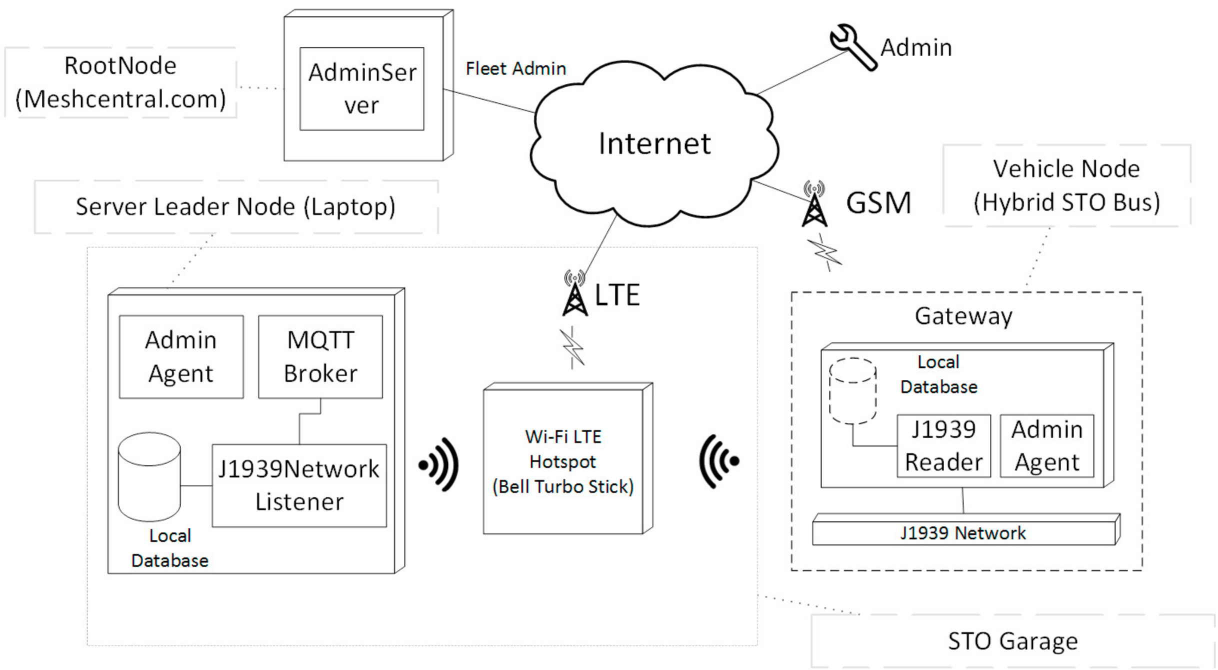

Fig. 3. Simple diagram of the MVP architecture deployed at the STO garage.

6. Conclusion

The present work provides a novel IoT architecture for predictive maintenance. A semi-supervised machine learning approach is proposed for improving the sensor feature selection of the COSMO approach [17], which we name ICOSMO. A prototype of the architecture has been installed at a garage of the STO, Gatineau, Canada. Currently only a single bus is equipped with a gateway, but we plan to expand the MVP and equip many more buses with gateways. This MVP is the foundation of our predictive maintenance machine learning and data analytic research. By using the J1939 data acquired from the buses, we will train machine learning algorithms, and once complete these algorithms will be deployed to the MVP. Future work will include completing the implementation of ICOSMO and running experiments.

References

[1] Amarasinghe, M., Kottegoda, S., Arachchi, A.L., Muramudalige, S., Bandara, H.M.N.D., Azeez, A., 2015. Cloud-based driver monitoring and vehicle diagnostic with obd2 telematics. 2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer) doi:10.1109/icter.2015.7377695.

[2] Bonomi, F., Milito, R., Natarajan, P., Zhu, J., 2014. Big data and internet of things: A roadmap for smart environments. volume 546.

[3] Braden, D.R., Harvey, D.M., 2014. A prognostic and data fusion based approach to validating automotive electronics. SAE Technical Paper Series doi:10.4271/2014-01-0724.

[4] Budakoti, J., 2018. An IoT Gateway Middleware for interoperability in SDN Managed Internet of Things. Ph.D. thesis. Carleton University.

[5] Byttner, S., Nowaczyk, S., Prytz, R., Rognvaldsson, T., 2013. A field test with self-organized modeling for knowledge discovery in a fleet of city buses. 2013 IEEE International Conference on Mechatronics and Automation doi:10.1109/icma.2013.6618034.

[6] Datta, S.K., Bonnet, C., Nikaein, N., 2014. An iot gateway centric architecture to provide novel m2m services, in: Internet of Things (WF-IoT), 2014 IEEE World Forum on, IEEE. pp. 514–519.

[7] Ding, B., Mends, D., Kiringa, I., Yeap, T., . Iot architecture with hybrid communication modes for fleet management, in: Technical report, Faculty of Engineering, University of Ottawa, Ottawa, Ontario, Canada.

[8] Furch, J., Turo, T., Krobot, Z., Stastny, J., 2017. Using telemetry for maintenance of special military vehicles, in: International Conference on Modelling and Simulation for Autonomous Systems, Springer. pp. 392–401.

[9] Gomaa, H., 2016. Real-time software design for embedded systems. Cambridge University Press.

[10] Kargupta, H., Sarkar, K., Gilligan, M., 2010. Minefleet®: an overview of a widely adopted distributed vehicle performance data mining system, in: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM. pp. 37–46.

[11] Killeen, P., Parvizimosaed, A., 2018. An ahp-based evaluation of real-time stream processing technologies in iot, in: Course report, Faculty of Engineering, CSI5311, University of Ottawa, Ottawa, Ontario, Canada, Winter 2018.

[12] Kumar, S., Dolev, E., Pecht, M., 2010. Parameter selection for health monitoring of electronic products. Microelectronics Reliability 50, 161168. doi:10.1016/j.microrel.2009.09.016.

[13] Libelium, 2018. Lorawan coverage for latam and asia-pacific on its iot sensor platform. URL: http://www.libelium.com/libelium-expands-lorawan-coverage-for-latam-and-asia-pacific-on-its-iot-sensor-platform/?utm_source=NewsletterLB&utm_medium=Email&utm_campaign=NLB-301018.

[14] Lui, L., Wang, S., Liu, D., Zhang, Y., Peng, Y., 2015. Entropy-based sensor selection for condition monitoring and prognostics of aircraft engine. Microelectronics Reliability 55, 20922096. doi:doi.org/10.1016/j.microrel.2015.06.076.

[15] Obaidat, M.S., Nicopolitidis, P., 2016. Smart cities and homes: Key enabling technologies. Morgan Kaufmann.

[16] Prytz, R., Nowaczyk, S., Rgnvaldsson, T., Byttner, S., 2015. Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data. Engineering Applications of Artificial Intelligence 41, 139150. doi:10.1016/j.engappai.2015.02.009.

[17] Rögnvaldsson, T., Nowaczyk, S., Byttner, S., Prytz, R., Svensson, M., 2018. Self-monitoring for maintenance of vehicle fleets. Data Mining and Knowledge Discovery 32, 344–384. URL: https://doi.org/10.1007/s10618-017-0538-6, doi:10.1007/s10618-017-0538-6.

[18] Svensson, M., Byttner, S., Rognvaldsson, T., 2008. Self-organizing maps for automatic fault detection in a vehicle cooling system. 2008 4th International IEEE Conference Intelligent Systems doi:10.1109/is.2008.4670481.

[19] Vaquero, L.M., Rodero-Merino, L., 2014. Finding your way in the fog: Towards a comprehensive definition of fog computing. ACM SIGCOMM Computer Communication Review 44, 27–32.

[20] Zhang, Y., Gantt, G.W., Rychlinski, M.J., Edwards, R.M., Correia, J.J., Wolf, C.E., 2009. Connected vehicle diagnostics and prognostics, concept, and initial practice. IEEE Transactions on Reliability 58, 286–294.

[21] Zimek, A., Schubert, E., Kriegel, H.P., 2012. A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining 5, 363387. doi:10.1002/sam.11161.