# Intelligent Information Management System for Decision Support: Application in a lift manufacturer's shop floor

T. Vafeiadis[1], A. Nizamis[1], K. Apostolou[2], V. Charisi[2], I.N. Metaxa[2], T. Mastos[3], D. Ioannidis[1], A. Papadopoulos[3], D. Tzovaras[1]
[1]Information Technologies Institute (ITI), CERTH Thessaloniki, Greece
E-mail: {thanvaf, anizami, djoannid, dimitrios.tzovaras}@iti.gr
[2]ATLANTIS Engineering SA, Greece
Email: {apostolou, charisi, metaxa}@abe.gr
[3]KLEEMANN HELLAS SA, Kilkis Industrial Area, Kilkis, Greece
Email: {t.mastos, ag.papadopoulos}@kleemannlifts.com

*Abstract*— **Intelligent systems and applications on manufacturing domain aim to improve decision-making capabilities, ease complex decision problems, offer predictions related to maintenance activities and provide cost savings to companies. In order to support the aforementioned functionalities, the intelligent prediction and decision support systems are based on machine learning and signal processing techniques, AI algorithms, IoT devices, data mining and modeling techniques, rules and fuzzy logic systems, and advance visualizations. In this paper, we introduce an intelligent information management system that aims to provide predictive maintenance and enhance decision support in a leading lift manufacturer. The proposed solution is a decision support system equipped with analytic tools, IoT sensors and visualizations. The system supports the full cycle of polishing procedures of the lift manufacturer, as it starts from predictive maintenance during the polishing machines' operation and ends in the scrap metals' removal after the operation. Both the intelligent information system and the scenario of its usage in the lift manufacturer's shop floor are presented in this work.**

***Keywords—intelligent system, data analytics, decision support, Industry 4.0, visual analytics***

## I. INTRODUCTION

The proposed intelligent system has been applied in a lift manufacturer's shop floor and aims to support a full cycle from production to scrap metal transportation. More specifically, the intelligent system focuses on the prediction of failures for the polishing machine and the optimization of the scrap metal collection process through the real-time notification of bins' fill levels and the suggestion of optimal routes within the shop-floor. The scenario is based on an integrated information management system that contains both software and hardware components. The introduced scenario starts from the piston production line in which the polishing service is one of the core activities.

The introduced work comes to add functionalities for collection and analysis of data from the production, towards predicting failures and avoiding congestion and delays within the factory. The end to end process is unified in one intuitive tool, overcoming the challenge of fragmented information at different parts of the factory and process.

The remainder of this paper is organized as follows. In Section II, the different data sources and types of IoT sensors are discussed. Data analytics tools, which enhance the functionalities of the Decision Support System (DSS), are described in Section III, while the DSS itself with its Finite State Machine structure and its Data Streaming Process is presented in Section IV. The application scenario at the lift manufacturer's shop floor is provided in Section V. Finally, we draw our conclusions with a brief outlook on future work in Section VI.

## II. DATA SOURCES – IOT SENSORS

The proposed Intelligent Information Management System is a completely web-based tool and its high-level architecture is available in Fig.1. The tool's functionality is based on both historical and live data, which are part of Shop Floor Sensors and Connectivity building block. The part of historical data is covered by the lift manufacturer's CMMS system, which is connected to the introduced system and provides data related to previous machines' failure and maintenance activities. Newly deployed sensors on both production line's polishing machine and scrap metal bins cover the part of live data.

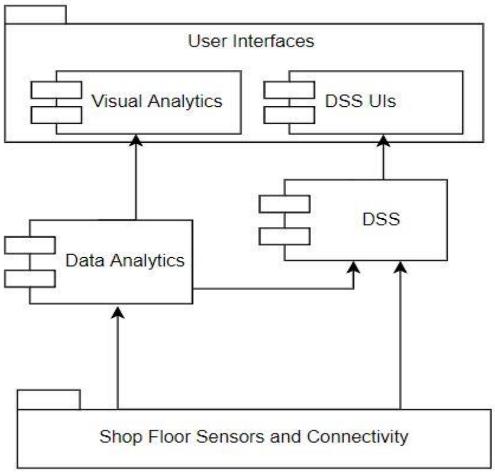


Fig. 1. High-level architecture.

In the case of predictive maintenance, vibration sensors have been deployed in the motors of the polishing machine in order to enable for the analytics tools to model the machine's operation and send notifications of abnormal behavior to the users using the notification mechanism of DSS. The vibrometer provides the acceleration of a motor in three $axis(x, y, z)$. The deployed vibrometers are powered by micro-USB wires connected to the polishing machine's power supply, and its communication is carried out via Wi-Fi. Then the data become available to Analytics tools and DSS using

MQTT[1] protocol and a corresponded broker[2]. The data format that is used is the OGC Observations and Measurements (O&M)[3]. In order to provide interoperability and enable the effortlessly extension of the system, the aforementioned format was followed for the description of all exchanged data within the system. Besides the data from different sensors, the Analytics tools output become available to DSS as OGC O&M. The communication of sensors and software components is secured as authentication and authorization services [4]of the message broker are used.

In the case of smart transportation of scrap metals, fill level sensors have been deployed in scrap metal bins which are located in the production line. The sensors enable the monitoring of fill level and trigger notifications from the DSS that inform workers to empty the bins. Light sensors prototypes have been used in order to enable the fill level measurement. The prototypes are enclosed in a plastic case and the complete sensor modules are enclosed in 3d printed cases. The communication is carried out via LoRa[5] wireless network. The data become available to Analytics tools using MQTT protocol, OGC O&M format for data description and security mechanisms as in vibrometers' case.

## III. ANALYSIS OF DATA – INTELLIGENT SYSTEMS

In this section, the data analytics tools of the proposed system are introduced. The tools support different types of algorithms. A probabilities model for CMMS historical data related to machine failures has been implemented for predictive maintenance. Furthermore, for the same purposes a real-time solution based on Eugen values summation of vibrometer data has been developed. An algorithm based on Dijkstra has been designed as well, in order to suggest to the workers the optimal route that the can use in order to empty the scrap metal bins.

### *A. Probablities model for machine predictivce maintenance*

Probability theory is the branch of mathematics concerned with probability. Although there are several different probability interpretations, probability theory treats the concept in a rigorous mathematical manner by expressing it through a set of axioms. Typically, these axioms formalize probability in terms of a probability space, which assigns a measure taking values between 0 and 1, termed the probability measure, to a set of outcomes called the sample space.

In the specified case were a predictive maintenance plan has to be made for a machine in the production line of the lift manufacturer. The approach that was developed is based on the calculation of probabilities of an upcoming event based on no-fault scenarios prior to that event. There are three types of machine faults; namely electrical, hydraulic and mechanical. The analysis is based on the series of these events, per day, in ten years range. In order for the analysis to be more concrete, the series is pre-processed in a way where the three faults are labeled as: $\{1\} \rightarrow$ electrical, $\{2\} \rightarrow$ hydraulic, $\{3\} \rightarrow$ mechanical and the days where no fault occurs as $\{0\}$. The probabilities of a fault to happen are calculated based on several scenarios that point to time ranges prior to that fault where no fault was happened. Thus, $Scenario\ 1$ – provide the probability of an event to happen (no fault, electrical, hydraulic and mechanical) after one (1) day of no-fault. Similar to $Scenario\ 1$, $Scenarios\ 2, 3$ and $4$ provide the probability of an event to happen after $2$, $5$ and $10$ days of no-fault.

In Fig. 2, for $Scenarios\ 1$ and $2$ and after $850$ days, the probabilities are: for $Scenario\ 1$ the probability of no-fault is $0.81$ the probability of an electrical fault is $0.09$, the probability of a hydraulic fault is $0.01$ and the probability of a mechanical fault is $0.09$, while for $Scenario\ 2$ the probabilities has the as much the same values. For $Scenario\ 3$ (Fig. 3) the probability of no-fault is $0.82$ the probability of an electrical fault is $0.1$, the probability of a hydraulic fault is $0.01$ and the probability of a mechanical fault is $0.07$, while for $Scenario\ 4$ the probability of no-fault is $0.84$ the probability of an electrical fault is $0.1$, the probability of a hydraulic fault is $0.01$ and the probability of a mechanical fault drops to $0.05$.

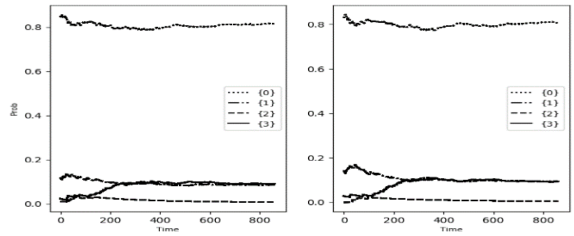


Fig. 2. Calculated probabilities for Scenarios 1 and 2.

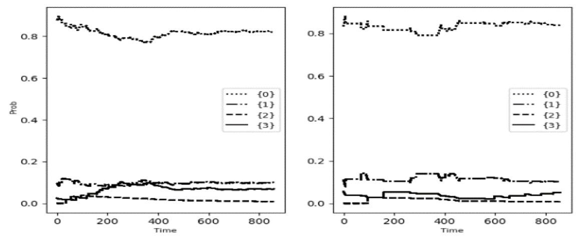


Fig. 3. Calculated probabilities for Scenarios 3 and 4.

The probabilities model output become available, in OGC O&M format and HTTP protocol, to the DSS for visualization in its user interface in order to inform, on a daily basis, the maintenance manager for the probabilities of future faults. Basic authentication is used in order to secure the communication.

### *B. Machine Vibrations Profile and Visual Analytics*

Besides the solution that is based on historical data, a more dynamic solution based on real-time data has been implemented. However, the test machine in lift manufacturer's plant site has no built-in sensors, and it was unable to provide live data for further analysis. Thus, the installation of a vibration sensor, which is introduced in the previous chapter, was considered as the most promising approach in order to get useful data for the detection of instant failures.

The vibration data are samplings of $1344$ samples of accelerations from three $axis(x, y, z)$ at a sampling rate of $1.344\ kHz$. Samplings are sent in average approximately every $1.5$ seconds, in proper motor operation, depending

[1] https://www.rabbitmq.com/
[2] https://mosquitto.org/
[3] https://www.opengeospatial.org/standards/om
[4] https://www.rabbitmq.com/access-control.html
[5] https://lora-alliance.org/

highly on network conditions. Due to WiFi buffer restrictions, every sampling is split into three packets, each corresponding to an axis. These packets contain raw sensor values and are translated in the back-end to accelerations measured at $m/s^2$ before being further analysed.

In [1], a method called Machine Vibration Diagnosis Profile (MVDP) for the real time detection of abnormal vibrations has been proposed. MVDP aims to detect the time point(s) when abnormal vibrations occur from the profile of the eigenvalues summation $es_i$, where $i$ is the time of the recording, and the calculated variance $v_i$ in a sliding window $w$ of fixed size and step one, calculated from the vibration sensor recordings from the three $axis(x, y, z)$. The basic assumption of MVDP is that significant eigenvalue sums with simultaneous significant variations could point out to abnormal vibrations.

The output of Machine Vibration Diagnosis Profile algorithm is visualized on real-time by the visual analytics tool of the proposed system. The tool is web-based and it is developed in AngularJS[6]. The graphical representations are enabled by the use of the ChartJS[7] and D3JS[8] visualization libraries. The visual analytics tool communicates with other system's parts such as data analytics or sensors using MQTT protocol and the communication is secured by the broker's security services. Fig. 4 below depicts the visualization of pilot plant machine vibrations' behavior.

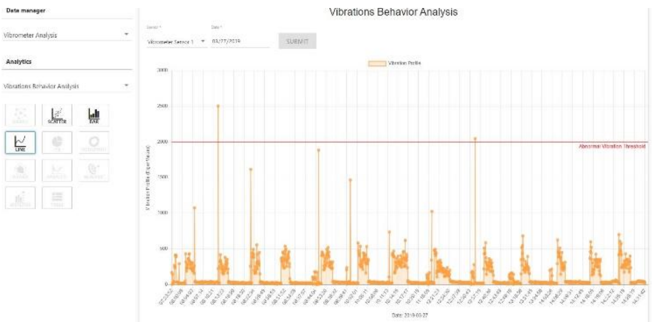


Fig. 4. Visualization of machine vibration diagnosis profile.

By using the above visualization tool, the maintenance manager is able to monitor in real-time the behavior of the polishing machine and be visually informed when the machine's activity surpasses the abnormal vibration threshold. Furthermore, the user is able to choose different types of visualizations such as bar diagrams, scatter or lines.

### C. *Optimal route for scrap metal bins' transportation*

The introduced algorithm proposes the optimal/shortest path that a worker can follow in order to empty a scrap metal bin from production line to an outdoors open top container. Firstly, the algorithm is triggered as soon as a bins' fill level is over 80%. The fill level percentage of a bin is available by the fill level sensor and through MQTT protocol.

A common way to mathematically model and represent road networks, in order to deal with problems such as the shortest path problem, is graphs G that are composed by sets of nodes N and sets of edges E. In graph theory, the shortest path problem is the problem of finding a path between two nodes in a graph such that the sum of the weights of its constituent edges is minimized. There are several works trying to solve this problem, the well-known Dijkstra's algorithm solves the single source shortest path problem in O(V 2) (worst case computational complexity), while there are various implementations of Dijkstra's algorithm that reduce the computational cost [2], [3], [4].

An extension of the Dijkstra's algorithm is the A* search algorithm [5] which achieves better performance by using heuristics to guide its search. Moreover the Bellman-Ford algorithm [6] solves the single-source problem if edge weights may be negative. There are, also, the Floyd-Warshall [7] algorithm, which solves all pairs' shortest paths, and the Johnson's algorithm, which solves the same problem, and may be faster than Floyd-Warshall on sparse graphs. Additional algorithms and associated evaluations may be found in [3].

Dijkstra's algorithm is decided to be used in the specified use case. As mentioned above, Dijkstra's algorithm finds a shortest path tree from a single source node, by building a set of nodes that have the minimum distance from the source. The source node graph has the following:

- vertices, or nodes, denoted in the algorithm by u or v
- weighted edges that connect two nodes: $(u, v)$ denotes an edge and $w(u, v)$ denotes its weight.

The pseudocode of Dijkstra's algorithm is given below:

```
dist[s] ← O
for all v ∈V-{s}
do dist[v]← ∞
S ← Ø
Q ← V
while Q≠ Ø
do u ← mindistance(Q,dist)
            S ← S∪{u}
for all v ∈ neighbors[u]
            do if dist[v]>dist[u]+w(u,v)
                        then dist[v] ←dist[u]+w(u,v)
return dist
```

For the scrap metal transportation problem in the pilot plant site, the source node of pilot plant is given in Fig. 5.

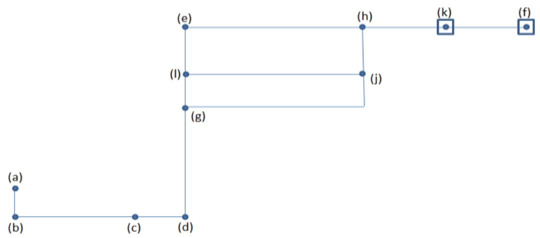


Fig. 5. Source node of pilot plant site.

The scrap metal bins inside the pilot plant are placed on (a), (c), (g) and (j) dots on Fig. 5 while the scrap metal open top containers are placed in nodes (k) and (f). The actual distances (in meters) between the nodes are given in the Table I below:

TABLE I. DISTANCE MATRIX (IN METERS) BETWEEN THE NODES

| | (a) | (b) | (c) | (d) | (e) | (f) | (g) | (h) | (j) | (k) | (l) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **(a)** | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **(b)** | 20 | 0 | 60 | 80 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

[6] https://angularjs.org/
[7] https://www.chartjs.org
[8] https://d3js.org/

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **(c)** | 0 | 60 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **(d)** | 0 | 80 | 20 | 0 | 140 | 0 | 70 | 0 | 0 | 0 | 0 |
| **(e)** | 0 | 0 | 0 | 140 | 0 | 120 | 70 | 60 | 0 | 100 | 0 |
| **(f)** | 0 | 0 | 0 | 0 | 120 | 0 | 0 | 60 | 0 | 20 | 0 |
| **(g)** | 0 | 0 | 0 | 70 | 70 | 0 | 0 | 0 | 90 | 0 | 0 |
| **(h)** | 0 | 0 | 0 | 0 | 60 | 60 | 0 | 0 | 40 | 40 | 0 |
| **(j)** | 0 | 0 | 0 | 0 | 0 | 0 | 90 | 40 | 0 | 0 | 60 |
| **(k)** | 0 | 0 | 0 | 0 | 100 | 20 | 0 | 40 | 0 | 0 | 0 |
| **(l)** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 60 | 0 | 0 |

The distance matrix is the main input of the Dijkstra's algorithm, and thus it has to be very precise and accurate. The measurements in pilot plant site took place with the use of GPS. After the application of proposed algorithm on distance matrix, the optimal paths from scrap metal bins to scrap metal open top containers are given in Table II:

TABLE II. OPTIMAL PATH FROM POINT A TO POINT B

| Point A | Point B | Route |
|---|---|---|
| (a) | (f) | (a) → (b) → (d) → (e) → (f) |
| (c) | (f) | (c) → (d) → (e) → (f) |
| (g) | (k) | (g) → (e) → (k) |
| (j) | (k) | (j) → (h) → (k) |

## IV. EXPLOITATION OF DATA – INTELLIGENT APPLICATIONS

### A. *Decision Support System*

Intelligent applications for industrial environments need to find ways to use heterogeneous data for a variety of purposes. Data acquisition, processing and analysis facilitates the decision making process for manufacturing. The proposed DSS comes in place to add value to the aforementioned data and it is used by the industrial partner for the maintenance operation, and specifically the predictive maintenance process. However, it has the potential to be further exploited for other processes on the shop-floor. The DSS is a fully web-based application that runs in real time and provides suggestions to the manufacturer's decision makers.

The developed system is a combination of a model-driven and data-driven decision support system. An architectural description of the system consists of its four main sub-components and the external components from which the DSS receives raw or processed data, which then uses for implementing the rule engine instance for the industrial partner's case.

The main DSS sub-components (see Fig. 6) are: i) the rule engine, ii) the stream processing sub-component, which handles all external system data connections and extracts the necessary information in the data to be used in the rule engine, iii) the data persistence sub-component which is responsible for the storage of inbound and outbound data for internal use, and iv) the HMI (Human-Machine Interface) for user friendly experience. The Analytics tools provide the input data to the DSS in OGC O&M format via either HTTP or MQTT communication protocols.

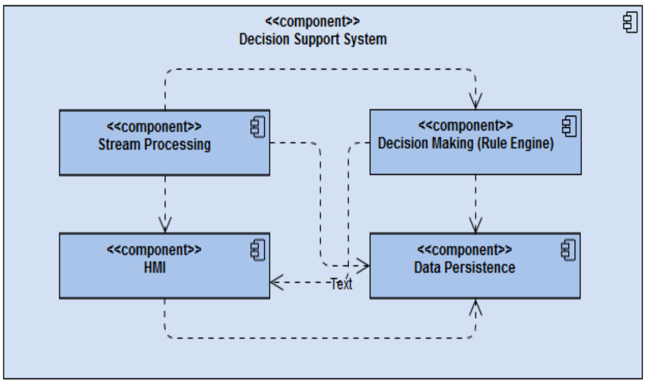


Fig. 6. System architecture for the Decision Support System

Various algorithms have been considered for the DSS implementation. Classification trees, genetic algorithms, support vector machines and Naïve Bayes algorithm were examined during the research and development phase of the system. Their complexity was suitable for large-scale decision support systems, where a large number of suggestions should be provided in very short time periods. In the manufacturing environments studied in this work, the DSS needs simpler and more accurate algorithms to perform, because the problems encountered are of specific nature and often appear after a large time period has passed. Training the system with real data from the shop floor has been the focus of our work.

Finite state machine and non – deterministic state machines were also tested during development. Both methods provided more accurate results with significant less resources than all the previous ones. Consequently, they were the ones chosen for the DSS implementation and deployment.

### B. *Finite State Machines*

A finite state machine (sometimes called a finite state automaton) is a computation model that can be implemented with hardware or software and can be used to simulate sequential logic and some computer programs. Finite state automata generate regular languages. Finite state machines can be used to model problems in many fields including mathematics, artificial intelligence, games and linguistics [8]. A finite state machine [9] is a tuple D ($Q, \Sigma, \delta, q_0, F$) where:

- Q: is a finite set of states
- Σ: is the input alphabet (any non-empty set of symbols)
- $\delta$: Q×Σ⟶Q: is the transition function
- $q_0$: is the initial state and
- F⊆Q: is the set of final states.

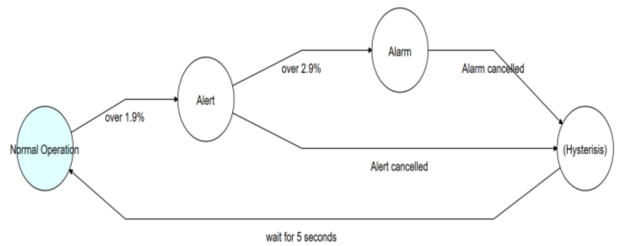


Fig. 7. Finite state machine for the SFT probability of hydraulic failure rule.

The vertices of the finite state machine (see Fig. 7) indicate the states of the machine. The labelled arcs in the graph represent the transitions. The arc labels are words from the finite state machine's language and they are made from the machine's alphabet. Initial state is denoted with the coloured circle and represents the normal operation of the system. All state machines should always find their path back to the initial state, for the manufacturing system to return to operating under normal conditions. The alphabet of the finite state machines, in these cases, is the conditions for each state. Each condition can be a mathematical expression, a regular alphanumeric expression or a combination of both of them. The transitions are defined from the alphabet and they are a subset of it. The function for each transition is evaluated as true or false and when the transition is evaluated as true, the system moves from the initial state to the final state.

The algorithmic steps for the rule engine of the DSS are given in the Table III below:

TABLE III. FINITE STATE MACHINE ALGORITHM USED IN THE RULE ENGINE

| | **Finite State Machine Algorithm for Decision Support Rule Engine** |
|---|---|
| 1 | Define the states for the rule engine |
| 2 | Define the transitions for the rule engine |
| 3 | Define the conditions for each state |
| 4 | Define the actions should be taken for each state |
| 5 | Describe a certain rule for a specific situation |
| 6 | Analyse the rule, discovering the states and transitions needed |
| 7 | Define the transition set which will be used for the rule |
| 8 | Set the values of the conditions |
| 9 | Set the limits of the conditions, which lead to preference for one of the transitions |
| 10 | Apply the rule on sample data to test its application |
| 11 | Apply the rule for real data |
| 12 | Use the DSS suggestion on shop floor problems |
| 13 | Improve rule with constant feeding it with new data |
| 14 | Revise rule when the conditions do not apply to the problem any more |

## C. *Data Streaming Process for the DSS*

Most data are continuous streams and they vary from sensor events, web site activity to results of mathematical analysis or simulations. Usually, they are timestamped at the source and carry the timestamp through the process of acquisition, transformation and exploitation in the system. Predictive maintenance requires the continuous monitoring of the shop floor through a sensor network. In the case studied here, the sensor network consists of vibrometer sensors and fill level sensors. The polishing machine of the lift manufacturer is a critical piece of equipment and it is monitored with two custom made vibrometers.

Data from the vibrometers are streamed to the Visual Analytics Toolkit (cf. section III B), and they are transmitted to the DSS via the MQTT message broker. The historical data of the polishing machine from the CMMS are also used for the analysis and prediction of failure as already discussed in section III A. The DSS uses RX/StreamInsight along with data at rest and stream processing infrastructure, in order to pull or push data from the repositories.

Data is stored in databases or file systems with mass storage capabilities. They are extracted and streamed via network and APIs to the suitable applications to be used. The DSS uses this data in the rule engine for rule creation and machine learning techniques in the historical data for training.

The DSS uses a stateful stream process which is a subset of the general stream process. The computation maintains contextual state. It stores all information derived from the detected events. The stateful stream processing connects the database and the key/value store tables with the event–driven data acquisition protocols. A stateful stream process during compilation and runtime increases application performance, scalability, data consistency and operational simplicity, as stated in [10].

The DSS uses in parallel stream processing and data at rest or batch processing. Stream processing is used in storage, analysis and training, while batch processing is used in the data acquisition from other components, based on MQTT and HTTP protocols (see Table IV). Both architectures are shown below (see Fig. 8)

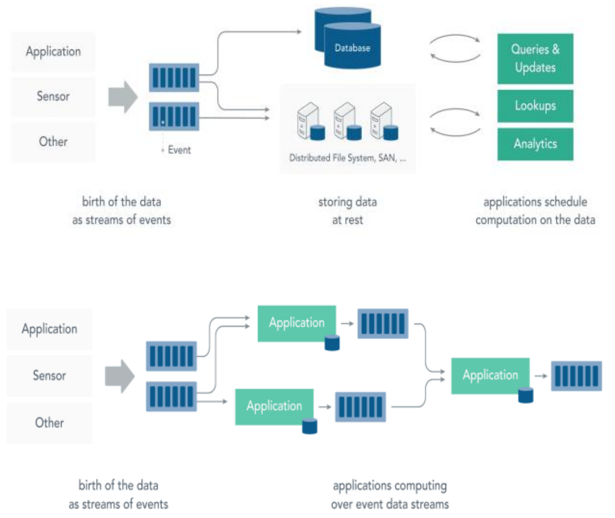


Fig. 8. Data at rest architecture and stream processing infrastructure [10].

TABLE IV. STREAM DATA TYPES

| **Stream** | **Description** | **Type** |
|---|---|---|
| Wireless Sensor Network data | Data from wireless sensors | Sensorial data |
| Vibrometer data | Acceleration data from vibrometer | Sensorial data |
| Analytics data | Monitoring data from machines | Sensorial data |
| Analytics probabilities for the polishing machine | Probabilities of failure for the polishing machine, based on the analytics algorithms | Processed data |
| Monitoring data – result of the Visual Analytics Toolkit | Binary data based on the monitoring of real – time data | Processed data |

# V. MAINTENACE DECISION SUPPORT AND FILL LEVEL MONITORING FOR LIFT MANUFACTURER

In order to better demonstrate the applicability of the proposed system, a scenario has been developed in close cooperation with the lift manufacturer. The scenario is based on two operations performed within the shop-floor i.e. predictive maintenance and scrap metal collection. The scenario takes place in the central factory's piston production line and starts with predictive maintenance. In the piston production line, pistons for hydraulic lifts are produced. The most critical machine of the piston production line is the polishing machine which requires effective maintenance in order to improve availability. The technologies used in the proposed intelligent system allow for predictive maintenance processes of the polishing machine, taking advantage of historical and real-time data. The probabilities for the prediction of different failure modes (electrical, hydraulic, mechanical) together with the outcomes of the Machine Vibrations Profile and Visual Analytics are fed in to the Decision Support System. The Maintenance Manager sets specific rules in the Rule Engine and prescribes actions to be taken, when the predictive maintenance scenario is activated. The personnel at the shop floor receive notifications and guidelines for actions via the DSS.

The demonstration scenario continues with the handling of the scrap metal that is produced after the polishing of the piston. The produced scrap metal, not only from the polishing machine but also from other machines, is collected in bins inside the shop-floor. The removal of the scrap metal after the operation is critical due to the limited availability of the bins. At this stage, early and automated detection of scrap metal fill levels is needed in order to empty the bins before they get full. Every day, forklift drivers pick up tens of scrap metal bins (a), (c), (g) and (j). Then, they transport the collected material to the open top containers (k) and (f) to dispose the scrap metal. The described process can sometimes cause congestion within the factory or delays in other tasks. Hence a shortest path proposition and a timely notification to the personnel is needed. The scenario ends in the suggestion of optimal routes within the shop-floor.

## CONCLUSIONS

The proposed Intelligent Information Management System has been applied and validated in a lift manufacturer's shop-floor. The major contribution of this work is the demonstration of Industry 4.0 developments in a real-world setting. Overall, the benefits gained until now regarding the maintenance part are the real time condition monitoring, the failure predictions estimation and the real-time KPIs tracking. Regarding the scrap metal collection process, the key outcome is the real time notification in order to empty the bins based on fill level monitoring, as well as the optimal path calculation.

In terms of the application of technologies, the experience gained in this work is related to the combination of heterogeneous data sources, the application of multiple data analysis methods, the development of the data streaming process subcomponent for the Decision Support System. Additionally, one of the most valuable lessons learned and challenges faced, was the application of technologies in a real industrial environment and in real operating conditions. The solutions proposed are robust and as complicated as necessary to support real-life situations. The main advantage of the solutions developed in this work is their general nature and transferability to other shop-floors and industrial cases. Future work is intended to experiment on exactly that, the transfer and application of the developed solutions to other production processes and plants.

## ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723145 - COMPOSITION. This paper reflects only the authors views and the Commission is not responsible for any use that may be made of the information it contains.

## REFERENCES

[1] T. Vafeiadis, A. Nizamis, N. Alexopoulos, C. Ntinas, T. Mastos, A. Papadopoulos, D. Ioannidis and D. Tzovaras, "Data analytics and data modeling for predictive maintenance and automated waste management: An integrated (IoT-Industry 4.0) sensor based solution to improve factory procedures", International Conference on Internet of Things and Smart Manufacturing, London, UK, 2018

[2] R.K. Ahuja, T.L. Magnanti, and J. B. Orlin, "Network Flows: Theory Algorithms and Applications", Prentice Hall, 1993

[3] B. Cherkassky, A. Goldberg, T. Radzik, "Shortest paths algorithms: theory and experimental evaluation", Mathematical Programming. Ser. A. 73 (2), 1996, 129–174

[4] F.B. Zhan and C.E. Noon, "Shortest Paths Algorithms: An Evaluation Using Real Road Networks", Transportation Science, 32, 1998, 65–73.

[5] P.E Hart, N. J. Nilsson, B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", IEEE Transactions on Systems Science and Cybernetics SSC4, 4 (2), 1968, 100–107.

[6] R.H. Kenneth, "Discrete Mathematics and Its Applications", 5th Edition. Addison Wesley, ISBN 0-07-119881-4, 2003.

[7] B.J. Jørgen; G. Gutin, "Section 2.3.4: The Bellman-Ford-Moore algorithm", Digraphs: Theory, Algorithms and Applications (First ed.). ISBN 978-1-84800-997-4, 2000.

[8] J.A. Anderson, "Automata Theroty with Modern Applications, Cambridge University Press, 1st edition, Cambridge, 2006.

[9] M. Almeida, N. Moreira, R. Reis, "On the performance automata minimization algorithm", Technical Report Series DCC-2007-03, 2007.

[10] https://www.ververica.com/what-is-stream-processing