

Lappeenranta University of Technology
LUT School of Energy Systems
Electrical Engineering
Master's Thesis
2018

Mikko Nykyri

DATA ANALYTICS FOR PREDICTIVE MAINTENANCE IN A PULP MILL — CASE ELECTRIC MOTORS

Examiners: Professor Pertti Silventoinen
D.Sc Mikko Kuisma
Supervisor: Kari Kerkelä

Abstract

Mikko Nykyri

Data analytics for predictive maintenance in a pulp mill — case electric motors

Master's Thesis

Lappeenranta 2018

48 pages

Examiners: Professor Pertti Silventoinen

D.Sc. Mikko Kuisma

Supervisor: Kari Kerkelä

Keywords: IoT, predictive maintenance, machine learning, data analytics

Industry is going through the fourth industrial revolution, as sensors and devices in industrial sites are being connected to the Internet. The collected data can be refined with machine learning and data analytics to create new value for businesses and industries, to improve productivity and production efficiency. UPM, a Finnish forest industry company, wants to create an IoT system for the electric motor predictive maintenance needs of the pulp mills. The mill's production efficiency increases when unplanned production halts can be avoided by using predictions created with machine learning methods.

The capabilities of data analytics were studied by generating machine learning models. The model detects and predicts overloads on a pump motor. Additionally, a model for pulp production amount prediction was also generated. The models were created using UPM Kaukas pulp mill process data gathered from automation system. The models were able to provide predictions with accuracy scores up to 98 per cent. Models were created both locally and on Microsoft Azure cloud service.

The technology for monitoring and predicting the functioning of a pulp mill exists, and by connecting the data from all data sources a mill-wide IoT system can be built. The system can be implemented in a cloud service, for example into Microsoft Azure by using the set of tools provided by the cloud service. The development of the system requires wide-scale knowledge about data analytics and process engineering, because the system requires all of the parts of process equipment to be modeled and all of the data sources to be connected. The system allows the monitoring of the mill as a whole and predicting fault situations therefore increasing the production efficiency of the mill. The methods presented in this thesis form a foundation for creating this mill-wide IoT system.

Tiivistelmä

Mikko Nykyri

Data-analytiikka sellutehtaan ennakoivassa kunnonvalvonnassa — case sähkömoottorit

Diplomityö

Lappeenranta 2018

48 sivua

Työn tarkastajat: Professori Pertti Silventoinen
 TkT Mikko Kuisma

Työn ohjaaja: Kari Kerkelä

Avainsanat: IoT, ennakoiva kunnossapito, koneoppiminen, data-analytiikka

Teollisuudessa on meneillään neljäs teollinen vallankumous, jossa anturit ja laitteet teollisuusympäristöissä yhdistetään Internetiin. Kerätystä datasta voidaan jalostaa koneoppivilla menetelmillä ja data-analytiikalla uutta arvoa yrityksille, millä voidaan parantaa teollisuuslaitosten tuotettavuutta sekä tuotantotehokkuutta. UPM, suomalainen metsäteollisuusyhtiö, haluaa rakentaa IoT-järjestelmän sellutehtaidensa sähkömoottoreiden ennakoivaan kunnonvalvonnan tarpeisiin. Tehtaan tuotantotehokkuus kasvaa, kun tehtaan suunnittelemattomia alasajoja voidaan estää koneoppivilla menetelmillä tehtyjen ennustaiden avulla.

Data-analytiikan mahdollisuuksia tutkittiin luomalla koneoppivilla menetelmillä malli. Malli tunnistaa ja ennustaa sähkömoottorin ylikuormatilanteita. Lisäksi luotiin malli sellun tuotantomäärän ennustamiseen. Mallit luotiin käyttämällä automaatiojärjestelmästä saatua prosessidataa UPM:n Kaukaan sellutehtaalta. Mallit saavuttivat ennusteissaan jopa 98 prosentin tarkkuuden. Mallintaminen tehtiin sekä lokaalisti että Microsoft Azure -pilvipalvelussa.

Tekniikka sellutehtaan toiminnan seuraamiseen ja ennustamiseen on olemassa, ja yhdistämällä datat eri tietolähteistä voidaan rakentaa koko sellutehtaan laajuinen IoT-järjestelmä esimerkiksi Microsoft Azure -pilvipalveluun käyttämällä pilvialustan tarjoamia työkaluja. Sovelluksen kehittäminen vaatii monialaista osaamista data-analytiikasta ja prosessista, sillä järjestelmän toiminta vaatii tehtaan kaikkien prosessilaitteiden mallintamisen ja eri järjestelmien datojen yhdistämisen. Järjestelmän avulla sellutehtaan toimintaa voidaan tarkkailla, ja ennustamalla vikatilanteita, tehtaan toimintaa voidaan tehostaa. Tässä työssä esitellyt menetelmät luovat pohjan koko tehtaanlaajuiseen IoT-järjestelmän kehittämiselle.

Preface

This master's thesis was written in the laboratory of applied electronics of Lappeenranta University of Technology. As a part of a research project with LUT and UPM, the subject of this thesis was provided by UPM and the research was a collaboration between the two.

From Lappeenranta University of Technology I would like to thank professor Pertti Silventoinen and Dr. Mikko Kuisma for examining my thesis and providing me with guidance to finish my work. I would also like to thank all the people in LUT office room 6405 for the amazing work environment, office humor and LaTeX pro tips. A word of gratitude to professor Jukka Hallikas and Dr. Kari Korpela is also in place, for organizing the research project. From UPM I would like to thank Tero Junkkari and Kari Kerkelä for the subject for this thesis and guidance throughout summer of 2018.

Perhaps the biggest thanks belong to my friends and family for their support throughout my life. For all of my friends here in Lappeenranta and especially everyone in my student association Sätky, I want to thank you for all the unforgettable moments. It seems like it was only yesterday when I walked through the university front doors for the first time.

Mikko Nykyri
October 23, 2018
Lappeenranta, Finland

Contents

Abstract

Preface

Contents

Nomenclature	7
1 Introduction	8
1.1 Research problem	9
1.2 Goals and research questions	9
1.3 Methods and material	10
2 IoT, machine learning & data analytics	11
2.1 IoT architecture	11
2.2 Security	13
2.3 Machine learning & data analytics	14
3 Industrial IoT in UPM Kaukas pulp mill	17
3.1 Kaukas pulp mill overview	17
3.2 Current IoT system in UPM Kaukas pulp mill	18
3.3 Earlier work: Performance prediction of birch pulp line	19
4 Analysis of data gathered from electric motors in predictive maintenance	21
4.1 Data analytics tools used	22
4.2 Structure of electric motor data in UPM Kaukas pulp mill	23
4.3 Prediction of pump motor overloading	23
4.4 Future prediction of overloads in pump motor	27
4.5 Prediction of production amount in oxygen phase	29
4.6 Summary of model based analysis	33
5 IoT application for UPM pulp mills	35
5.1 System overview	35
5.2 Proposed application	36
5.3 Live demo	39
5.4 Challenges & considerations	40
6 Conclusion & discussion	42
References	45

Nomenclature

API	Application Programming Interface
IEEE	Institute of Electrical and Electronics Engineers
IIoT	Industrial Internet of Things
IoT	Internet of Things
LUT	Lappeenranta University of Technology
MLPC	Multi-layer Perceptron Classifier
MLPR	Multi-layer Perceptron Regressor
SVR	Support Vector Regression
avg	Average
max	Maximum
pred.	Prediction

1 Introduction

When discussing industrial revolution, people often think about the first industrial revolution that took place between 18th and 19th centuries when steam engine changed industry for ever. In reality, industry has gone through multiple revolutions throughout history. After the rise of the steam engine, in the end of the 19th century, began the second industrial revolution. Mass production became more and more common when electric power and assembly lines became available. The third revolution took place in the 1970s, when industry automation systems emerged and took over industry as electronics and semiconductor technology became affordable enough.

Today we are experiencing the fourth industrial revolution. Internet of Things (IoT) is shifting towards industry applications. IoT, connecting things to Internet, allows us to collect data from different locations. Gathering raw data, however, is not new technology and data may not be that valuable in its raw form. Yet, by refining gathered data, new information can be created and that can be valuable. Present day industrial automation systems mainly visualize data for the factory operators and the full value of the collected data is not utilized. There is huge potential in the data - the driving force in digitalization of industry is to harness this potential, to improve productivity. Information refined from industrial data can yield savings for the operator. For example, if a motor fault can be predicted by analyzing data, the downtime caused by unexpected faults can be minimized.

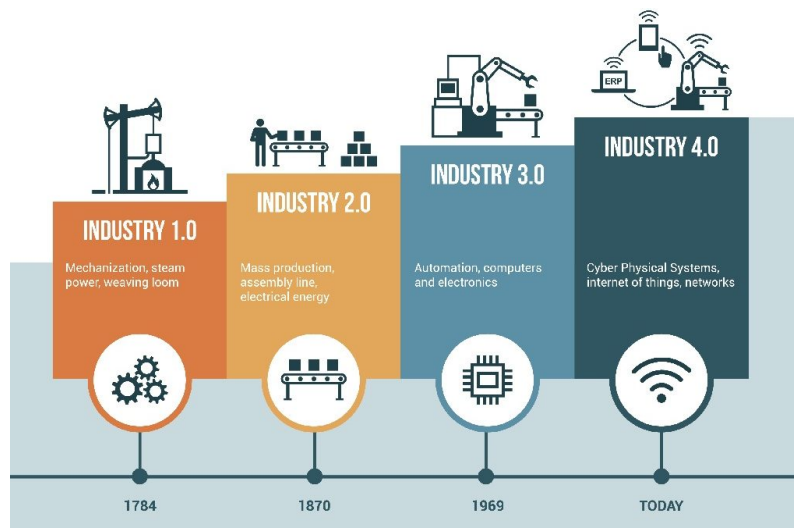


Figure 1.1: Industry has gone through four revolutions throughout history, starting from the 18th and 19th centuries to the rise of industrial IoT today. (Cline, 2017)

Malfunctioning process equipment and machinery cause significant financial losses in factories, for example, in pulp mills. Unexpected production halts cause the loss of tonnes of produced pulp. UPM, a major Finnish forest industry company, has showed interest in building an IoT system for predictive maintenance use to be deployed in their pulp

mills. IoT system means, in this thesis, the connections of things and devices to each other and the Internet (for example, sensors and the cloud). Deployment of Industrial IoT (IIoT) systems is a new industry itself around the globe, and IIoT applications are not in a wide-spread production use.

1.1 Research problem

Data is collected in multiple processes in UPM pulp mills. However, the collected data is not utilized in, for example, predictive maintenance, even though IoT technology allows new applications to gain new value from the data. Because of the complexity of industrial systems a wide spectrum of different devices, there are no ready solutions for UPM to be deployed on their pulp mills.

There is huge amount of devices, for example electric motors, in UPM pulp mills. Data is gathered in different systems, and there are no connections between these systems. The data needs to be handled manually, which is a significant job. With data analytics and digitalization, a machine learning model could be deployed to monitor the data, and detect and predict anomalies. However, there is no pre-made model for this application.

1.2 Goals and research questions

The goal of this thesis is to study how data gathered from electric motors from a pulp mill can be refined to gain information on process performance and maintenance needs. In this thesis, the connections between data are studied, and the draft of a real-time system is proposed.

This master's thesis is a collaboration between Lappeenranta University of Technology (LUT) and UPM, and it is a part of a LUT project *Digital Supply Chain – Systemic Value Transforming within Industry Internet*, of which LUT School of Energy Systems, LUT School of Business and Management and UPM are a part of. One of the goals of this research project is to use, test and develop IoT applications and systems in businesses. The environment and studied IoT system in this thesis is one example use case of such an IoT system.

The most crucial research problems of this thesis are:

- How can gathered data in a pulp mill be refined to produce new information on electric motor performance and maintenance needs?
- What are the different data sources in UPM pulp mills and how can they be connected together in a pulp mill IoT system?
- What kind of IoT application could be suitable for UPM's needs?

Along with the problems listed above, this thesis covers an overview of the current state of IoT cloud services and their data analytics capabilities, and emerging IoT technologies and common issues, such as cybersecurity and blockchain.

1.3 Methods and material

This thesis consists of literal research and analysis of electric motor data from UPM Kaukas pulp mill as a case study. Suitable motor case selection is based on, for example, malfunction data.

2 IoT, machine learning & data analytics

Internet of Things is defined by the Institute of Electrical and Electronics Engineers (IEEE) as "A network of items - each embedded with sensors - which are connected to the Internet" (IEEE, 2015). This joint between the physical and digital world has been studied widely, and now it is taking ground in the industrial world as Industrial IoT, when industrial assets and equipment are connected to the Internet, enterprise-level information systems, business processes and to the people operating or using them (Diab et al., 2017).

With IoT, devices and things are able to collect and share data, which allows for example monitoring and remote sensing of these devices. When the collected data is combined with data analytics and machine-to-machine communication, new information can be created to improve productivity and therefore gain business value. For example, machine learning tools combined with sensor data can be used to predict failures beforehand. (Kanawaday and Sane, 2017)

2.1 IoT architecture

The architecture of a general IoT system can be divided into three layers: perception layer, network layer and application layer (Swamy et al., 2017). The layers lie on top of each other, and are connected to the adjacent layer. In layman's terms, the layers represent the hardware, the connections and the brains of the IoT system. In this thesis, the solutions presented and studied mainly affect the network and application layers of the IoT system. The structure of IoT layers is presented in Figure 2.1.

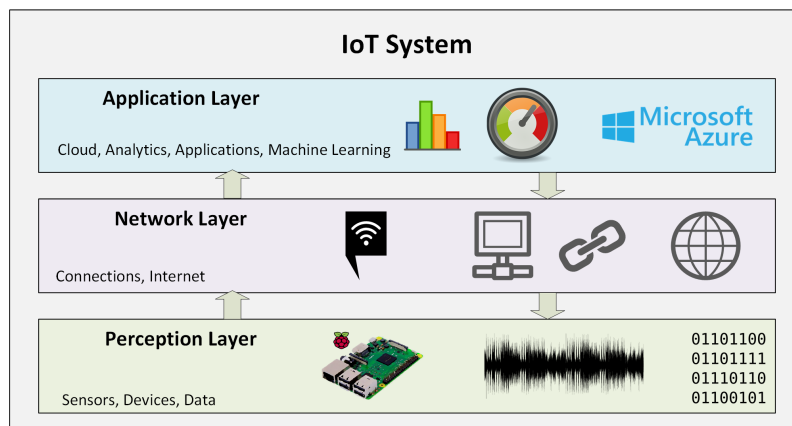


Figure 2.1: The three layers of IoT systems: Perception layer, network layer and application layer. The layers lay on top of each other and are connected to the adjacent layers.

The perception layer, also called the perception and control layer, is the lowest layer in the IoT architecture. It consists of things, systems or processes, which collect data. Instances in the perception layer can be for example sensors and automatic recognition

tools. (l. Zhong et al., 2017) Real-world examples of such sensors can be for example IoT devices such as Raspberry Pis, motor current sensors and environmental sensing equipment, such as infrared movement sensors.

The network layer, also called the transmission layer, is the middle layer of the IoT architecture. It consists of systems which transfer data gathered in the perception layer to the application layer. (l. Zhong et al., 2017) The network layer links systems into each other and brings the Internet onto things. Examples of systems on the network layer are Bluetooth, Wi-Fi, (serial) bus and radio transmission.

The application layer is the top layer of the IoT architecture. The application layer contains all the smart features of the IoT system, and therefore realizes the IoT application. The application layer also works as the link between the IoT system and its user, for example a monitoring person or a smart device. (l. Zhong et al., 2017) An example of an application layer component is a cloud, which is possibly the most common and well-known part of an IoT system.

The cloud is perhaps the most crucial component of an IoT system. It is part of the application layer, and usually contains all the smart features of an IoT system. The cloud can be a on-site server or an off-site server, which are often offered as a platform- or a system-as-a-service model by many technology companies. These cloud systems offer many services such as cloud computing, machine learning, data analytics, data storage and data visualization. The biggest IoT cloud providers as of June 2018 are Microsoft Azure, IBM Cloud, Amazon Web Services and Google Cloud Platform (srgresearch.com, 2018). The largest provider by customer base is Amazon, with 33 per cent of the market share (srgresearch.com, 2018).

Table 2.1: Market shares of major IoT cloud platform providers (srgresearch.com, 2018)

Platform	Market share
Amazon Web Services	33 %
Microsoft Azure	13 %
IBM Cloud	8 %
Google Cloud Platform	6%

Depending on the IoT solution, data processing and calculations can be done in the cloud or in the IoT device itself. On early IoT systems, all the processing was done in the cloud or a centralized server. However, in recent times, IoT systems have distributed some or all of the computing tasks to the IoT devices, to cut down the amount of data transmitted between the sensor and the cloud. (Mihai et al., 2018) This technology is called edge computing, and it has become possible solution due to the increasing computational power of IoT devices and the increasing amount of data to be sent over the network layer.

2.2 Security

The security of any IoT system is a matter which should not be overlooked in any application. Cybercriminals are interested both in data and the devices connected to the Internet. Unprotected devices are vulnerable to attackers, which can tamper with the firmware of IoT devices. The tampering can lead into theft of data or harnessing of the device for botnet or spy use. Devices can be easily protected, though. With a proper firewall, any unauthorized connection attempts can be blocked away. Even the smallest of devices can be targeted. Earlier in Digital Supply Chain research project, a single unprotected Raspberry Pi was attacked (Nykyri et al., 2017).

The data transactions are most vulnerable when in the network layer. The Internet is an open space and data packets can be intercepted by criminals or even by competitors. The packets should therefore be encrypted with a strong enough hash key. The theft of information and/or technology can lead into significant financial losses. The production status of a pulp mill or any factory is usually classified information. If the production predictions leaked into the public, competition strength would weaken and it could also affect the stock trade of the company or its stakeholders and partners.

One possible solution for IoT security issues is a technology called blockchain. Blockchain technology is based on an open, network-distributed ledger which records all transactions. All transactions are grouped into blocks, where each block is linked to the previous block. The blocks therefore form a chain, and their integrity and verification is validated by the network of nodes. (Gatteschi et al., 2018) The ledger of blocks is copied to all of the nodes on the network and the data is matched on random intervals, on average every ten minutes. This shared nature of the ledger prevents tampering of the data, since it is impossible to change data which is distributed. (Korpela et al., 2017) Blockchain technology can therefore be used to secure transactions between endpoints. The principle of a blockchain is presented in Figure 2.2.

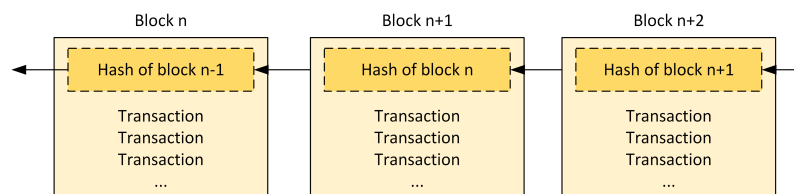


Figure 2.2: The principle of blockchain. Each block contains list of transactions performed and a hash to the previous block. (Christidis and Devetsikiotis, 2016)

Blockchain technology is secure, and it is globally used for example in cryptocurrency transactions. Other applications include public records like land or property titles, vehicle registrations and birth certificates. Blockchain can be used also for identification purposes like for example passports and identity cards, different assets like copyrights and various other kinds of documents. (Gatteschi et al., 2018)

2.3 Machine learning & data analytics

IoT systems allow refining the gathered data into some kind of new information, which provides new value for the IoT system operator. To gain new information, the process which is sensed needs to be modeled. This process of modeling and extraction of knowledge from data sets is called machine learning (Bakshi and Bakshi, 2018). This can be done, if a sufficient amount of data is available. In addition to the raw data, depending on the use case, detailed description of the data may be required. If, for example, a model predicting faults is desired, to teach the model to detect faults both data of flawless operation and faulty operation is needed. The model can then be trained with the data, and detect the current condition of the sensed device.

Machine learning consists of different algorithms to produce models, find patterns and predict a user-defined target output. (Bakshi and Bakshi, 2018) The development of models can be divided into five steps (Anderson et al., 2017):

1. Data collection
2. Feature extraction and reduction
3. Model creation
4. Model validation
5. Deployment

As mentioned above, a crucial requirement for modeling is data, which needs to be collected before any analytics can be put into action. After a sufficient amount of data has been gathered, in the feature extraction and reduction phase, new data columns are generated based on the collected data and/or some columns are dismissed as unimportant (Anderson et al., 2017). An example of a new generated data can be a true/false flag based on an existing data column, for example a flag whether a value is greater than a pre-defined reference value. Before the model is created with a machine learning algorithm in the model creation phase, the data set is usually split into training and testing data sets. Only the training set is used to create a model, and the testing set is later used to test how well the created model works. If the model is deemed accurate enough, it can be deployed into use and for example real-time data can be fed into the model. The workflow of machine learning model creation is presented in Figure 2.3.

An algorithm is simply put a set of instructions to solve a problem. Machine learning algorithms are a collection of these instructions, which solve a prediction problems. There is a vast amount of such algorithms, which work in different kinds of problems. Machine learning algorithms can be divided into supervised and unsupervised ones (Anderson et al., 2017). In supervised learning, the model learning happens with input-output pairs, and the target output is pre-known (Sasikala et al., 2017). An example of supervised learning is an image recognition task, for example a model

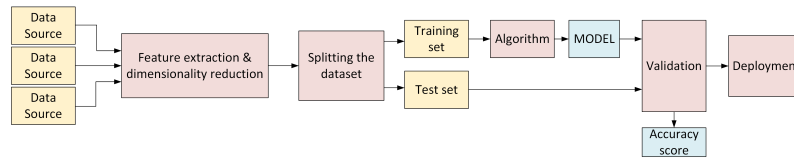


Figure 2.3: The workflow of machine learning modeling. Data is pre-processed, split into training and testing sets and a model is created based on the training set. The model is then tested with the test set.

detecting whether a parking lot is full or empty. In unsupervised learning, the output is not known beforehand, and new information to be extracted is modeled with just the raw data (Sasikala et al., 2017). An example of such model is dividing a group of electric motors to a number of sub-groups based on data, for example age and fault history.

Most common tasks for machine learning include classification, regression and clustering. In classification, the target is to predict a class from a pre-defined set of possible classes. (Sasikala et al., 2017) For example, with classification, the model can predict whether an electric motor is faulty or not. Also, the previously mentioned parking lot image recognition task is a classification task. The classes do not have to be binary - the model can be trained to classify to more classes than two. With regression, one can predict continuous values rather than selecting from a group of available classes (Bakshi and Bakshi, 2018). Clustering means dividing the input data set into groups, and is an example of unsupervised learning (Bakshi and Bakshi, 2018). Examples of machine learning algorithms are listed in Table 2.2. The algorithms below are grouped with the task for which the algorithm is suited for. It is to be noticed that the amount of available algorithms vastly exceed the amount of the ones listed below.

Table 2.2: Examples of machine learning algorithms, grouped with the task for which the algorithm is suitable for.

Task	Algorithm
Classification	Random Forest
	Logistic Regression
	Neural networks
	Gradient Boosting
Regression	Support Vector Machines
	Decision Trees
	Neural Networks
Clustering	K-Means
	Mean-Shift

The models created with machine learning algorithms can be deployed into IIoT use to create new value from existing data. In Figure 2.4, the principle of an IoT system using

cloud-based data analytics and predictive modeling is described. The data flows from sensors (perception layer) to cloud platform (application layer) via Internet (network layer). The data, once in the cloud, is then processed, and the flow of the data is fed to a model which has been previously created with data collected from the production site. In the upcoming chapters, the modeling and deployment of a model in UPM pulp mill environment is presented.

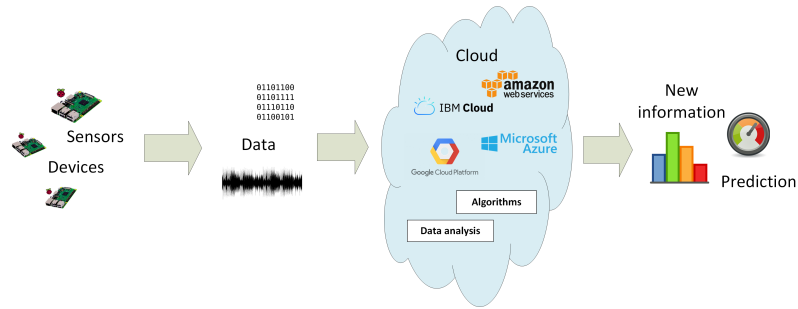


Figure 2.4: The principle of an IoT system where data is gathered from sensors and uploaded to cloud where new information is refined.

3 Industrial IoT in UPM Kaukas pulp mill

The pulp mill selected for this study is UPM Kaukas Pulp Mill in Lappeenranta. UPM has three similar pulp mills in Finland: Kaukas, Pietarsaari and Kymi. Kaukas pulp mill was chosen as the case mill for this study, because of the ongoing collaboration between LUT and UPM and its vicinity to LUT. However, all the mills share technology and know-how, and therefore study conducted in Kaukas mill can be considered valid in Kymi and Pietarsaari mills, too.

3.1 Kaukas pulp mill overview

UPM Kaukas industrial complex consists of a pulp mill, a biorefinery, a paper mill and a sawmill. The Lappeenranta-located site is a major employer in the area (approximately 1000 employees), and it is a part of the Finnish UPM corporation with production sites around the world.



Figure 3.1: UPM Kaukas industrial complex is located in the city of Lappeenranta, Finland. It consists of a pulp mill, a biorefinery, a paper mill and a sawmill. (upmbiofore.com, 2017)

In this study, the focus lies on the pulp mill. In UPM Kaukas, the pulp is manufactured using modern sulfate process. Pulp consists mainly of cellulose and hemicellulose, and it is made by dissolving lignin from the wood material. Lignins are complex polymers, which bind the cellulose molecules together in wood materials. Pulp making process in UPM Kaukas pulp mill can be split into five main phases:

1. Chopping the wood into chips
2. Cooking of the chips into pulp using white liquor (sodium hydroxide, sodium sulfide, sodium carbonate, sodium sulfate)
3. Washing

4. Bleaching

5. Drying

Pulp can be manufactured either with either a continuous or batch-based process. In the batch-based process, the reaction chamber is filled with wood chips and cooking chemicals (white liquor). The whole chamber is cooked into pulp, and then it is moved to washing where cooking chemicals and impurities are removed from the pulp, and later on to bleaching. Continuous cook differs from batch cook: the reaction chamber is fed with new wood chips and cooking chemicals continuously and finished pulp is extracted at the same time. In UPM Kaukas pulp mill, batch cook process is used.

3.2 Current IoT system in UPM Kaukas pulp mill

As of June 2018, the UPM Kaukas pulp mill collects process data in Metso DNA system. This system collects data from sensors measuring different quantities of each piece of equipment. The system also generates event and alarm information, if the value of a quantity exceeds a pre-defined limit. General information about each piece of equipment is stored in SAP. This information includes the installation date of the equipment, and any data about any maintenance work done on it. This data is also duplicated into Microsoft Azure cloud platform.

Table 3.1: List of systems where UPM Kaukas Pulp Mill process data is stored as of June 2018

Collected Data	Place of Storage
Process data	Metso DNA
Events and alarms	Metso DNA
Reports on done installations and/or maintenance	SAP, Microsoft Azure

To a certain extent, the existing automation system in UPM Kaukas pulp mill can be considered as an IoT system. The sensors in process equipment represent the perception layer. Local connections in the mill form the network layer and Metso DNA system represents the application layer. There are, however, no connections between SAP and Metso DNA, therefore making the network layer solution insufficient. Any cross-reference of data must be done manually using Microsoft Excel, and the Metso DNA system does not utilize the full potential of the data.

There are plans to integrate all process data to Microsoft Azure, thus implementing a cloud solution. Data analytics on the process data could then be done in Microsoft Azure as an application layer solution. Data analytics can provide new information on the existing data.

3.3 Earlier work: Performance prediction of birch pulp line

Prior to the research conducted during making this thesis, a case study of an IoT application was conducted in LUT. UPM presented a problem with Birch pulp line of the pulp mill. A pump was malfunctioning more often than it should and it required constant replacing. These replacements required a halt in production which caused losses in revenue. According to observations made in UPM, the thickness of the mass pumped by this motor decreases when the pump starts to malfunction. LUT, in collaboration with IBM and UPM began to study whether it was possible to predict the thickness of this mass, and therefore predict the performance of the process thirty minutes into the future.

The source of data in the system consisted of an offline database with Microsoft Excel format. The database contained process data from one year timespan, with data points gathered once a minute. The data itself was collected in the factory with Metso automation system. However, at this point, there were no real-time link to an IoT service provider (such as Microsoft Azure or IBM Cloud).

A model of the process was created using IBM SPSS Modeler software. SPSS Modeler is a data science and analytics tool which uses a graphical user interface (IBM, 2018). The software can create predictive models, and it supports multiple algorithms and analysis capabilities (IBM, 2018). The model created for UPM Kaukas Birch line consisted of pre-selected input variables and their derivatives:

- Temperature (4 measurement points)
- Valve positions (3 measurement points)
- Input mass flow

The model was uploaded to IBM Watson Machine Learning service on IBM Cloud, and the offline database was uploaded to IBM Watson IoT Service, which is a service for managing IoT devices and systems. In the service, Node-RED was deployed. Node-RED is a visual programming tool specializing in connecting hardware, software, application programming interfaces (APIs) and online services (nodered.org, 2018). Node-RED was configured to take data from the offline database, and feed it to the previously created model uploaded in IBM Watson Machine Learning. The Machine Learning Service then uses the model to calculate a prediction based on the passed data, and then returns the prediction back to Node-RED in IBM Watson IoT Service. The architecture of Kaukas Birch Pulp line system is presented in Figure 3.2.

The prediction result along with raw measurement values is visualized with visualization tools of Node-RED. A screenshot of the user interface is presented in Figure 3.3. The values are presented with gauges. Additionally, the measured and predicted performance is projected into a graph over time.

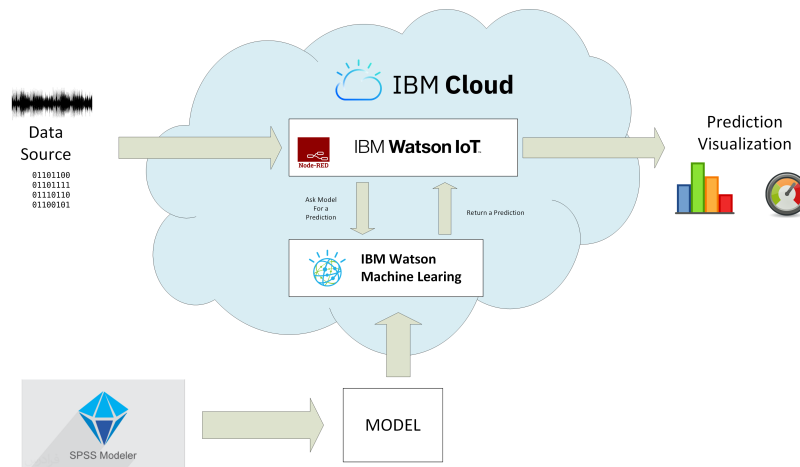


Figure 3.2: The architecture of the predictive IoT system of birch pulp line. The system consists of a data source, pre-made model and visualization.

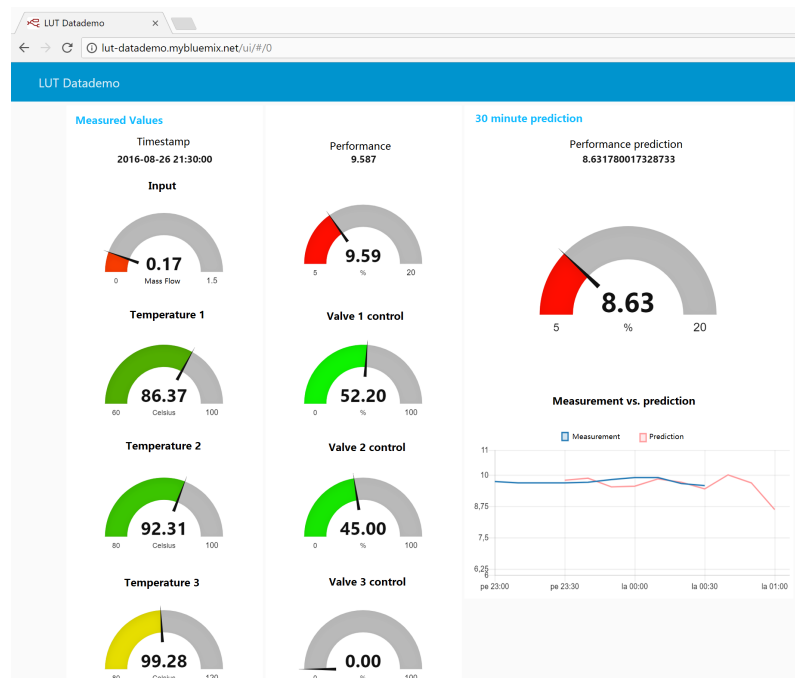


Figure 3.3: A screenshot of the user interface of the birch pulp line performance prediction system. The prediction, along with all measurements are presented with gauges. Additionally, the measured and predicted performance is projected into a graph over time.

The system created managed to predict the pulp thickness fairly well. The Architecture presented here is suitable for both online and offline database usage, and also works as a base for the electric motor data analysis discussed later.

4 Analysis of data gathered from electric motors in predictive maintenance

An electric motor is a crucial piece of equipment in a pulp mill. In UPM Kaukas pulp mill alone there are nearly 4000 electric motors. When taking Kymi and Pietarsaari mills into account, UPM has over 10000 electric motors in its pulp mills. Most of the motors are used for powering pumps to either mix the material or move it from one place to another. The motors in UPM Kaukas come in different sizes and are manufactured by multiple companies, including ABB/Strömberg and Siemens.



Figure 4.1: An electric motor powering a centrifugal pump - a typical application of an electric motor in UPM Kaukas pulp mill. *Picture source: <https://www.sulzer.com/en/shared/products/2017/03/28/12/55/ahlstar-app-t-range>*

Malfunctioning electric motors can cause large-scale production losses in pulp mills. The faults in induction motors can be divided into three categories (Karmakar et al., 2016):

1. Electrical-related faults (unbalanced supply voltage, overcurrent, overvoltage, overload, earth fault, inter-turn short circuit etc.)
2. Mechanical-related faults (broken rotor bar, damaged bearings, rotor winding failure, mass unbalance etc.)
3. Environmental-related faults (moisture, vibration, ambient temperature related issues etc.)

Earlier, research has been conducted on predictive maintenance on electric motors. For example, Fourier analysis on steady-state and especially start-up currents on electric motors have proven possible solutions on detecting faults on electric motors in mining applications (Antonino-Daviu et al., 2017). However, the data collected in UPM Kaukas pulp mill is not sampled on high enough frequency to utilize this method. Instead of

focusing on high-frequency measurements, a machine learning method is implemented in this thesis. There are earlier studies of machine learning methods used in predictive maintenance, for example, neural network has been used to detect abnormal operational conditions of hydroelectric generators (Nadai et al., 2017). Fault detection and diagnosis of rotating mechanical system have been improved by using pattern learning algorithms (Habib et al., 2016).

4.1 Data analytics tools used

There are many different tools for data analysis. In the previous study on birch pulp line, IBM SPSS Modeler was used. In this study, the analysis of the electric motors was performed with Python and its additional libraries Pandas and scikit-learn. Along Python, Microsoft Excel was used for data retrieval. Both Pandas and scikit-learn were obtained using Anaconda, a Python data science platform which consists of a regular Python install, a collection of libraries (including both Pandas and scikit-learn) and Jupyter Notebook and Spyder development environments.

Pandas is an open source Python library designed for data analysis use. Pandas offers data structures and data analysis tools which the plain installation of Python lacks. (Pandas, 2018) Pandas allows the massive amount of Excel-formatted electric motor data to be loaded into Python and processed.

Scikit-learn or sklearn for short is a python library specializing in machine learning. Scikit-learn provides tools for data mining and data analysis. It can perform classification, regression, clustering, dimensionality reduction, modeling and pre-processing of the data. The library is built on NumPy, SciPy and matplotlib, which are scientific libraries for Python. (scikit learn, 2018) Scikit-learn allows to use data structures built with Pandas to create predictive models on electric motor data.

The flow of the data starts from Metso DNA automation system, where it is retrieved to Microsoft Excel with a proprietary add-in. The excel sheets are then loaded into Python using pandas library, and predictive models are created using scikit-learn. The data chain is presented in Figure 4.2.

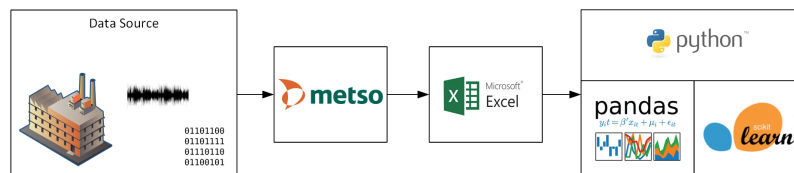


Figure 4.2: The data chain of modeling. The data is first retrieved from Metso DNA automation system using Microsoft Excel. The Excel formatted data is then pre-processed in Python using Pandas, and a prediction model is generated with scikit-learn.

4.2 Structure of electric motor data in UPM Kaukas pulp mill

The process data is collected in Metso DNA automation system. Direct-driven motors provide less data than frequency-controlled motors: from direct-driven motors, the current percentage is the only physical quantity measured from the motor itself. The collected physical quantities of direct-drive motors is presented in Table 4.1, and quantities of frequency controller driven motors is presented in Table 4.2. The additional data provided from the frequency controller motors provide viable information for data analytics, for example the motor temperature.

Event and alarm data is stored in Metso DNA in a consistent form. Every piece of event information contains the event type (usually an alert), start time, end time, time of acknowledgement, priority level, location, location description, event description and amount of triggers. The alerts are prioritized ranging from three to five, five being the most critical. Location and event descriptions tell where the alerting piece of equipment is and what caused the alert.

Table 4.1: Data Collected from direct drive electric motors

Quantity	Unit	Data type
Runtime	h	Float (Discrete)
Amount of start-ups	-	Integer
Current	%	Analog

Table 4.2: Data Collected from frequency converter driven electric motors

Quantity	Unit	Data type
Current	A	Analog
Power	kW	Analog
Rotational speed	rpm	Analog
Torque	Nm	Analog
Temperature	°C	Analog

4.3 Prediction of pump motor overloading

The selection of motors to be studied started with familiarizing oneself with the pulp making process and inspecting the electric motors in Metso DNA automation system. Brown mass handling/oxygen phase sector of pine pulp line was selected for further inspection, because that section contains a large amount of pumps. From the sector, the alarm and event data from different motors was printed out one at a time. Upon inspecting the data, daily overload reports were discovered originating from one electric motor. The motor in question is a 315 kilowatt alternating current motor manufactured

by ABB, powering a medium consistency pump on a buffer container. The current percentage of this motor over time is presented in Figure 4.3. From the figure, the current spikes exceeding 100 % of the rated maximum current can be seen occurring rather frequently. Electric motors are able to withstand overcurrent to some degree. While under overcurrent or overload, the temperature of the coils inside the motor rise. The rising temperature seldom causes instant failures, but does however affect the lifetime of a motor. Rising operation temperatures damage the stator winding insulation and cause mechanical stress fatiguing windings (Ransom and Hamilton, 2013). Therefore, further analytics was decided to be conducted on the motor. If it was possible to make a model capable of detecting and/or predicting these overloads, it would be an example of a useful piece of new information generated with machine learning and analytics.

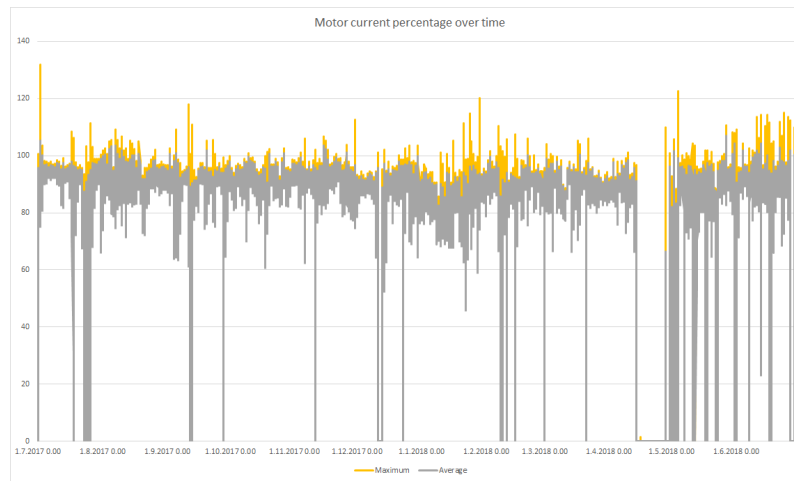


Figure 4.3: The average and maximum current percentage of an electric motor in UPM Kaukas pulp mill. The current spikes exceeding 100 % of the rated maximum current can be seen occurring rather frequently.

Measurements covering one year (1st July 2017 - 1st July 2018) were gathered and the data was split into two parts: the training part and the testing part. The split was set to May 1st 2017, therefore giving 10 months of training data and 2 months of test data. Using the data, a machine learning model was fit to detect overload without using the current measurement itself. The data used for the model were:

- Current measurement of two adjacent pump motors (average and maximum)
- Current measurement of an adjacent filter (hydraulics pump motor and mass eject pump)
- Production of pulp mass on the oxygen sector
- Average thickness of the mass
- Average mass flow

- Maximum mass flows (5 measurement points)
- Inverter frequency of an adjacent pump (average and maximum)
- Average valve positions of two different valves
- Temperature measurement of the overloading motor and an adjacent motor

In addition to measured data from Metso DNA, an additional data column was created for the model. This column was a true/false flag indicating whether the motor was running on overload or not. The column was filled in Excel with simple logic: if the motor current was over 100 percent, the flag in that row was set as 1, otherwise 0. This column was also used as the target value of the prediction.

The machine learning task for this case is classification, because the desired output is whether the motor is under overload or not. The model was generated using random forest classification algorithm. The model was trained to detect whether the motor has been running on overload in the past 10 minutes, without measuring the maximum current. The model accuracy was 99,60 %. The prediction results are presented as a confusion matrix in Table 4.3. The rows represent predictions (pred.) and the columns represent actual values. The ideal result would be that the top right and bottom left corners would be zero, since those are either false positives or false negatives. From the matrix, it can be seen that the model is very good at detecting true negatives: the situations where the motor is not under overload. The motor was under overload in 101 samples and the model managed to predict 67 of those samples, missing one of them. However, the model gave quite a few false positives, 34 of them. The amount of false positives is quite high, when the number of true positives is taken into account.

Table 4.3: Confusion matrix for the prediction. The rows represent actual values, and the columns represent predicted values. The ideal result would be that the top right and bottom left corners would be zero, since those are either false positives or false negatives.

		Actual overload	
		No	Yes
Pred.	No	8682	34
	Yes	1	67

The performance of the model can be presented also with a receiver operating characteristic curve. The steeper the curve, the better the model is detecting true positives. The ideal curve would be shaped like a step function, rising immediately to 1,0. The red line indicates a pure random guess - if the curve falls below the red line, it indicates that the model gives more false positives than true positives. Therefore, the area between the curve and the red line should be as large as possible. The receiver operating characteristic curve of the model is presented in Figure 4.4.

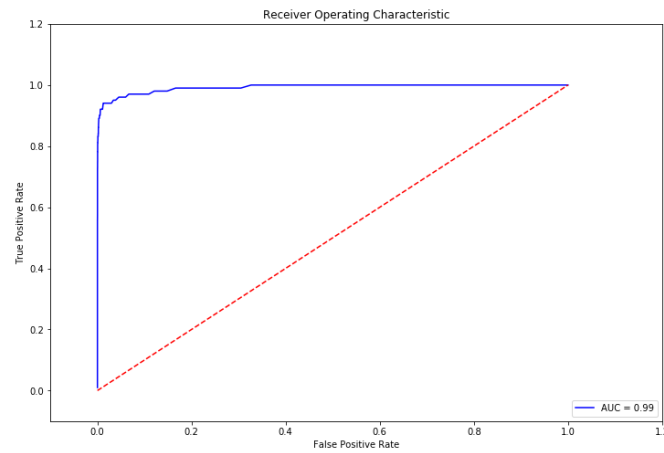


Figure 4.4: The receiver operating characteristic curve of the model. The curve raises very sharply, indicating high accuracy. The red line represents a pure random guess, so the area between the curve and the red line should be as large as possible.

The data columns have different levels of importance in the model - some of the data is more significant than other. The data importances can be plotted as a bar chart. The importances of the data are presented in Figure 4.5. According to the diagram, three of the most important data columns are maximum (max) and average (avg) currents of adjacent motor 2 and maximum mass flow 1. The least important value is the current of filter discharge pump motor.

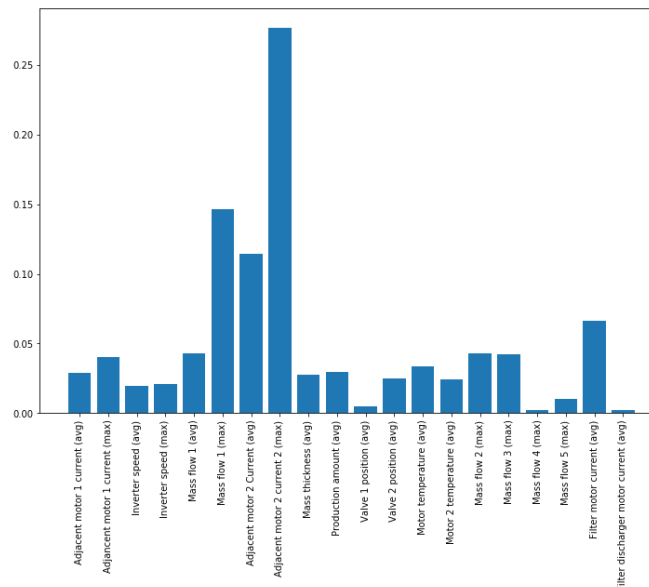


Figure 4.5: The importances of data columns in the model. The most important data for the model is the maximum current of adjacent motor 2, and the least important is mass flow 1.

4.4 Future prediction of overloads in pump motor

The model was later tweaked to try to predict overloads in the motor ten minutes into the future. To accomplish this, a new data column was generated in Microsoft Excel to indicate future overloads. Each row on the column was set 1 if an overload would occur ten minutes from the row's timestamp. This new future overload column was then set as the target for the model. Because this model predicts into the future, the present time current information could be included into the model training dataset.

The modeling to predict the future turned out to be a more challenging task. Therefore, multiple algorithms were tested to determine which would suit the job the best. The classification algorithms tested were random forest, gradient boosting, logistic regression, Multi-layer Perceptron Classifier (MLPC), Gaussian NB and Quadratic discriminant. The highest model accuracy score was MLPC with a score of 98,85 %. This is due to the low amount of false negatives (zero instances). However, the model did not detect any overloads correctly, and all positive results were false positives. Even with slightly lower accuracy percentages, random forest or gradient boosting provides better results. Even though both present false positives and negatives, the models were able to predict overloads. For example, with gradient boosting model, when the model predicts an overload, the prediction is correct 44,66 % of the time. The confusion matrices and model accuracies of all trained models are presented in Tables 4.4a through 4.4f, and the data importance bar charts and receiving operating characteristic curves are presented in Figures 4.6 and 4.7.

Table 4.4: Confusion matrices and accuracy scores for six different classification algorithms.

(a) Random Forest				(b) Gradient Boosting			
		Actual overload				Actual overload	
Pred.		No	Yes	Pred.		No	Yes
	No	8666	89		No	8604	56
	Yes	17	12		Yes	79	45
Accuracy: 98,38 %				Accuracy: 98,83 %			

(c) Logistic Regression

		Actual overload	
		No	Yes
Pred.	No	8447	85
	Yes	236	16

Accuracy: 96,35 %

(d) MLPC

		Actual overload	
		No	Yes
Pred.	No	8683	101
	Yes	0	0

Accuracy: 98,85 %

(e) Gaussian NB

		Actual overload	
		No	Yes
Pred.	No	1241	1
	Yes	7442	100

Accuracy: 15,27 %

(f) Quadratic Discriminant

		Actual overload	
		No	Yes
Pred.	No	1088	0
	Yes	7595	101

Accuracy: 13,54 %

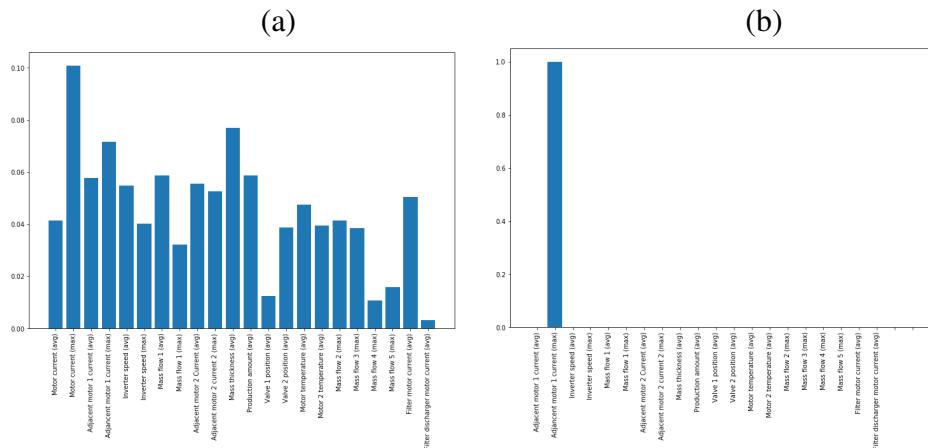


Figure 4.6: Feature importances of random forest and gradient boosting classifiers. Gradient boosting relies only on one input variable, while random forest utilizes more variables.

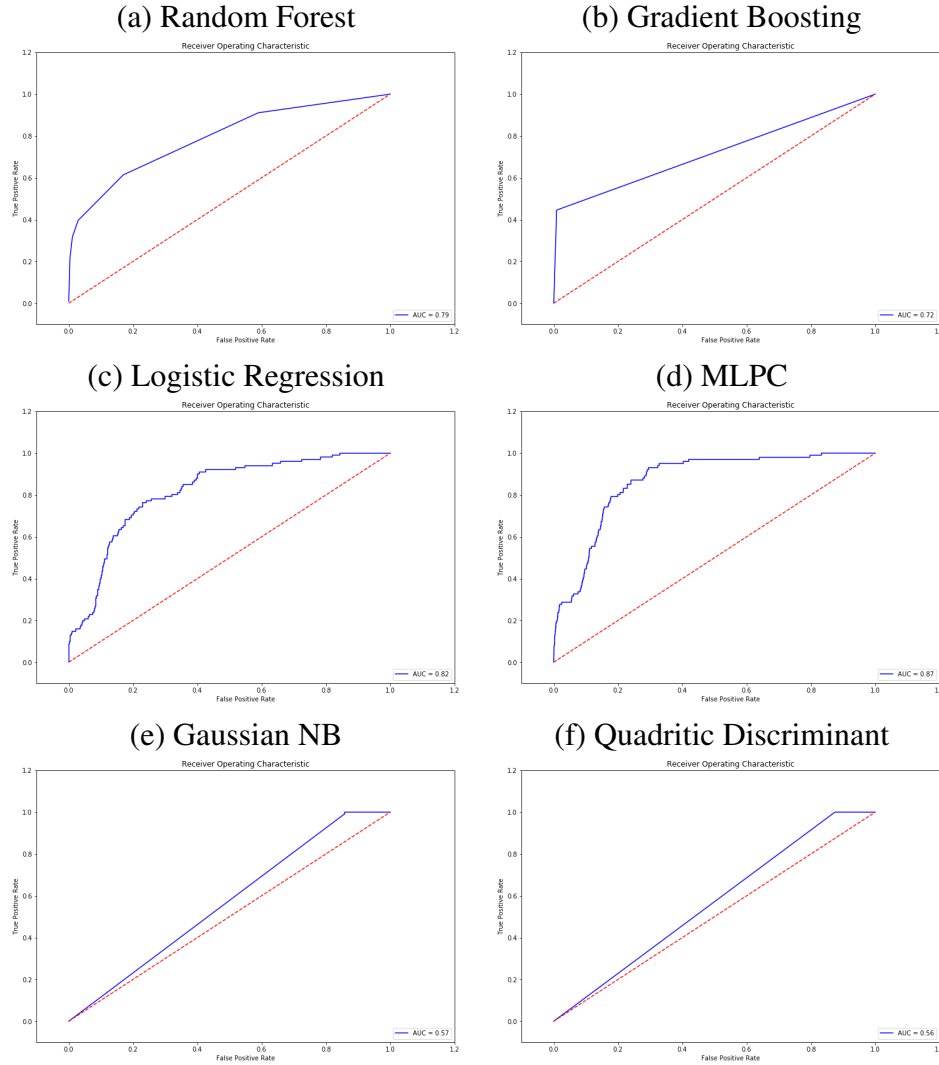


Figure 4.7: Receiving operating characteristic curves of six different classification algorithms. Random forest (a) and Gradient Boosting (b) give the best results.

4.5 Prediction of production amount in oxygen phase

Along with the overload predictions, predicting the production of pulp mass in the oxygen phase of the pulp making process was also studied. The training dataset was the same as with the motor overload analysis, but the production value was set as the target column rather than as input data. Also, the training period was extended to June the 15th, thus setting the test set to 15 days. Predicting a continuous value is a regression task, so six different regression algorithms were selected. Models were trained with Linear and nonlinear Support vector regression (SVR), Decision tree, Extra Trees, Nearest neighbors and Multi-layer Perceptron Regressor (MLPR) algorithms. Just as with the overload analysis, the prediction of present time production model was trained first. The predicted and actual production amounts over time are presented in Figure 4.8. The best

regressor for this case was MLPR with accuracy score of 95,16 %.

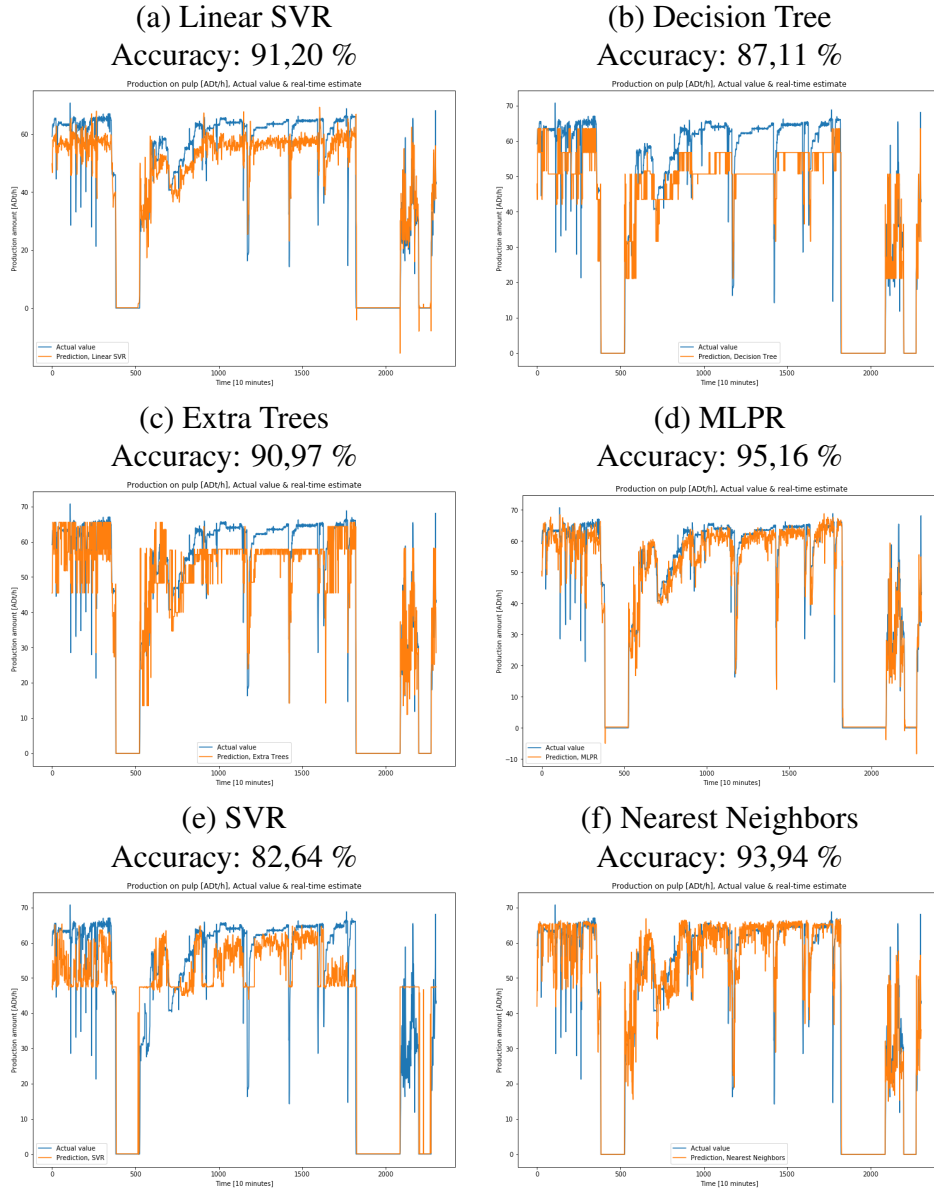


Figure 4.8: Actual and real-time estimate of pulp production with six different regression algorithms. The best result is obtained with (d) MLPR.

The model was then trained to predict the production amount into the future. The present time production amount was put back to the set of input data. The predicted and actual production amounts over time are presented in Figure 4.9. All models managed to predict the production quite well, with MLPR still being the most accurate with an accuracy score of 97,76 %.

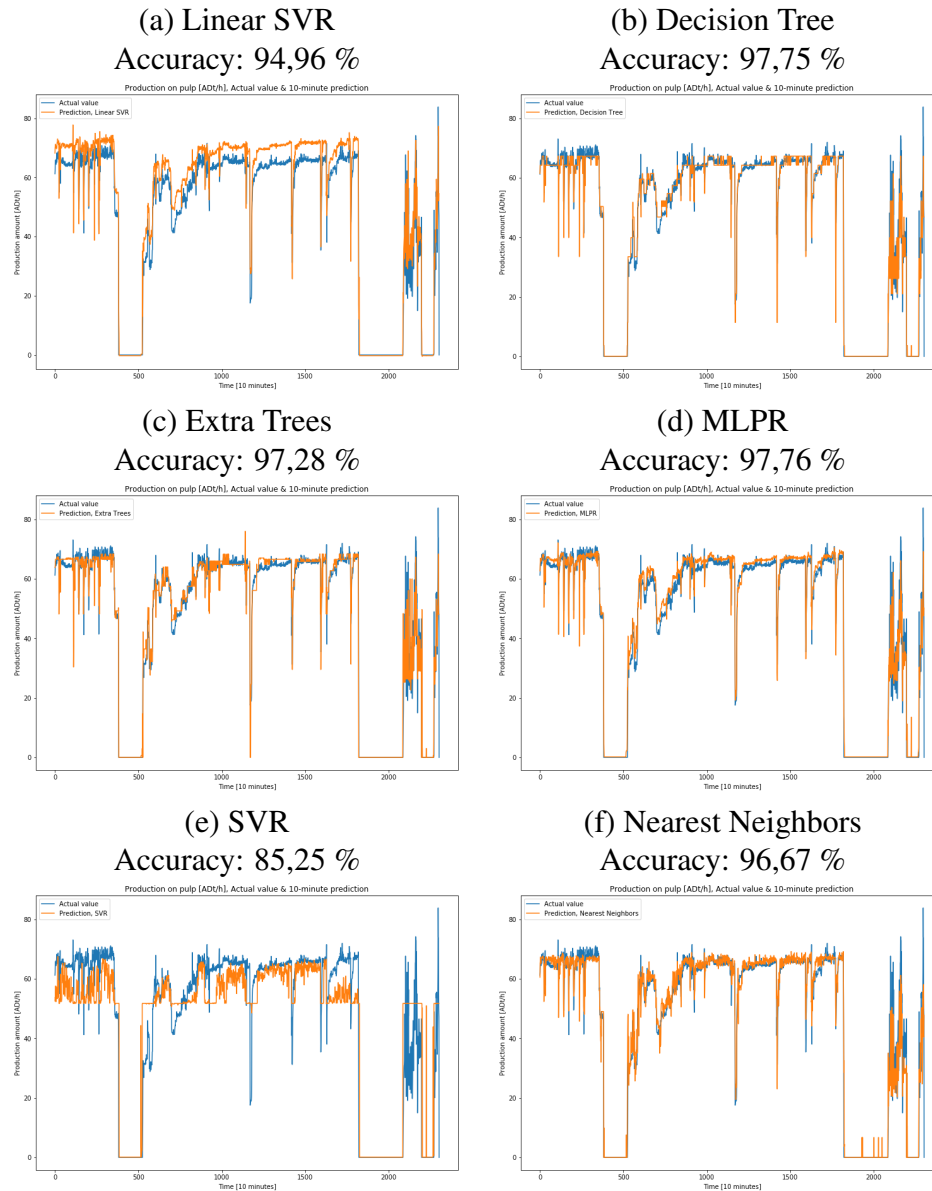


Figure 4.9: Actual and 10-minute prediction of pulp production with six different regression algorithms. The best result is obtained with (d) MLPR.

When the forecast was advanced further to one hour, the model performance dropped slightly. Decision Tree, Extra Trees and SVR models produce rather distorted curves. MLPR, Linear SVR and Nearest Neighbors produce still quite accurate curves, MLPR being yet again the most accurate with a score of 90,74 %. The predicted and actual production amounts over time are presented if Figure 4.10.

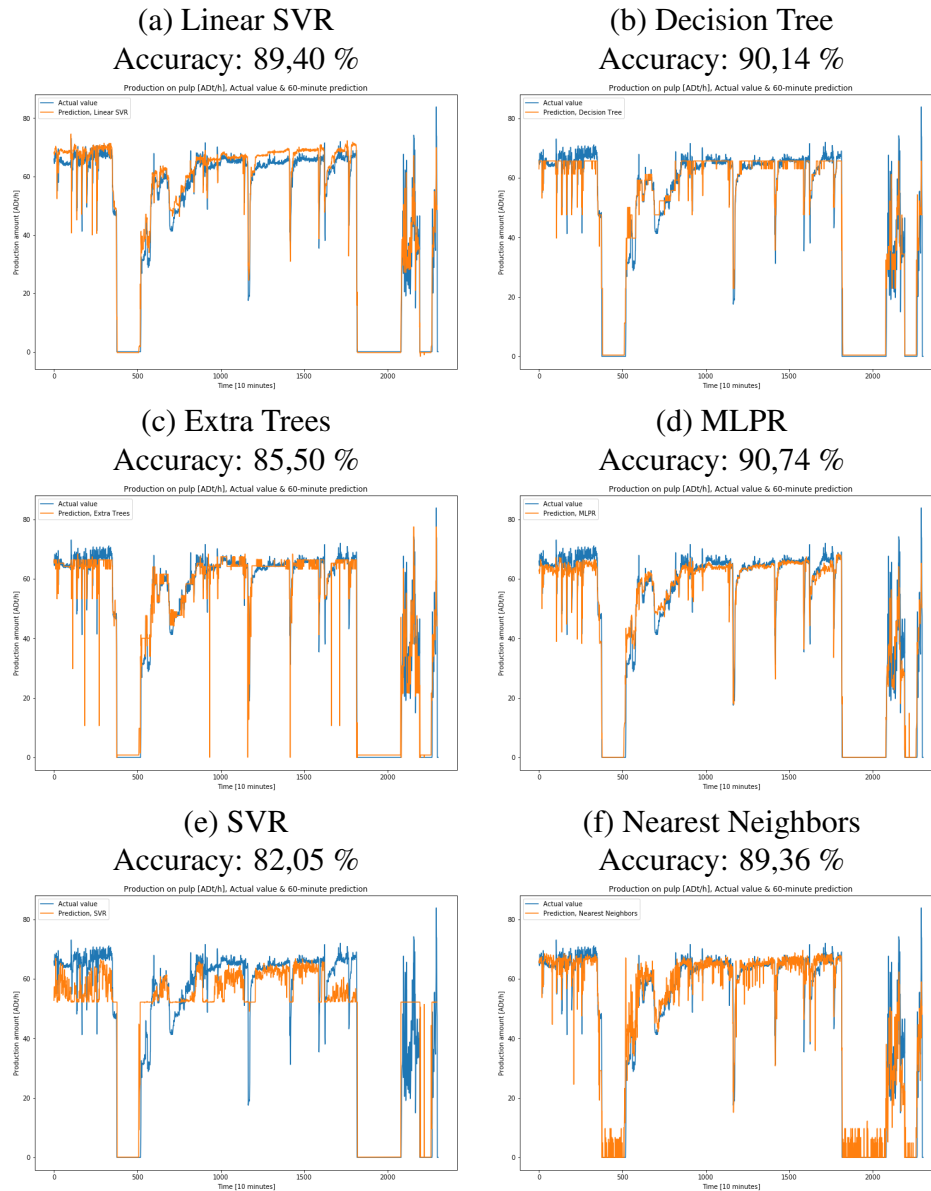


Figure 4.10: Actual and 60-minute prediction of pulp production with six different regression algorithms. The best result is obtained with (d) MLPR.

When the forecast was advanced even further to 12 hours, the model performance dropped significantly. The Linear SVR and MLPR models still manage to predict the shape of the production curve, but the amounts lower than in reality. The decision tree and extra trees model predictions are closer to the real amount, but the shape is heavily distorted and very inaccurate in some points. Nearest neighbors model accuracy drops significantly. The highest accuracy score is still with MLPR, with 38,34 %. However, decision tree produces values closer to reality, but with the penalty of heavily distorted curve. The predicted and actual production amounts over time are presented in Figure 4.11.

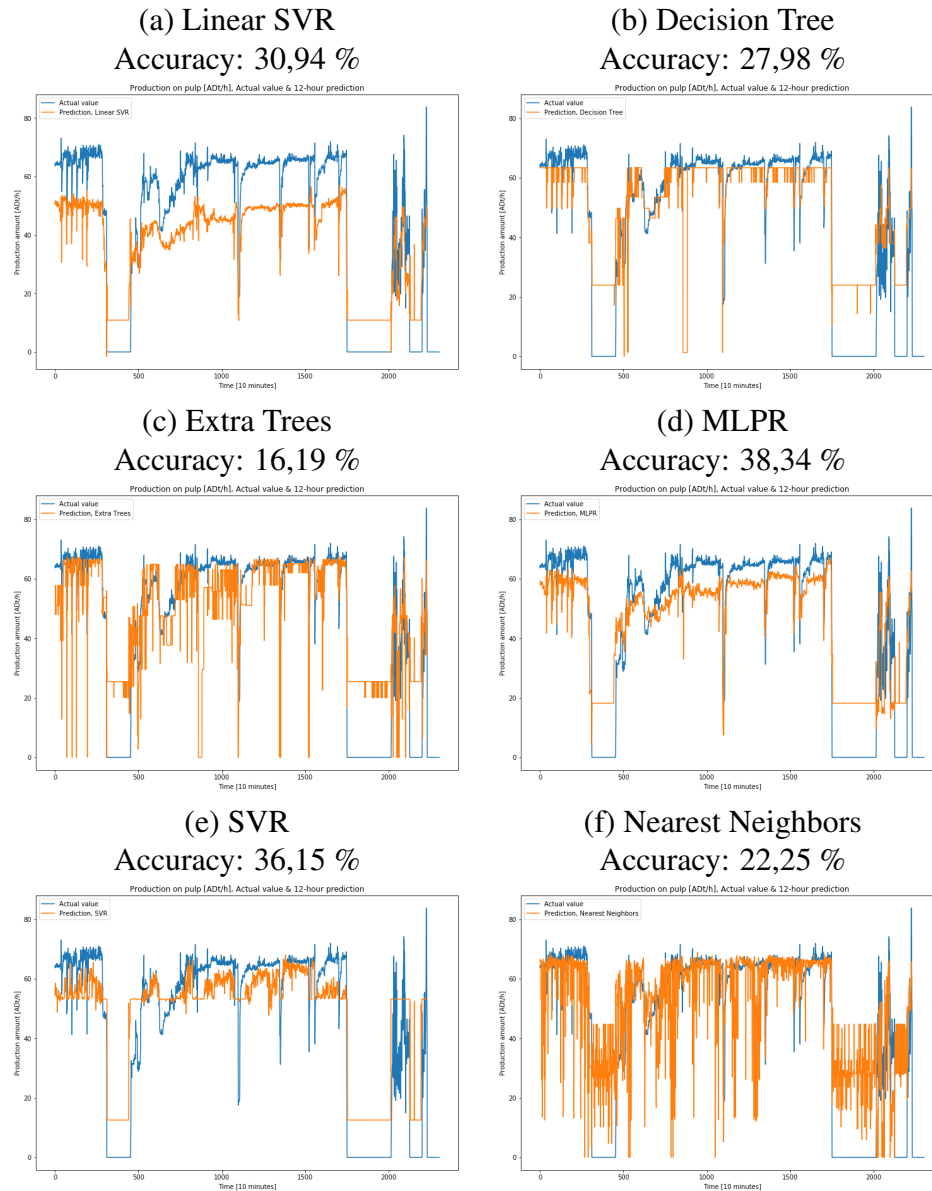


Figure 4.11: Actual and 12-hour prediction of pulp production with six different regression algorithms. The best result is obtained with (d) MLPR.

From the figures above, it can be seen that the production trend can be predicted to some extent. The curve form is heavily dependent on the used algorithm - decision tree and extra trees give more accurate percentages on some cases, but the curve is significantly more distorted.

4.6 Summary of model based analysis

There are a lot of different tools available for modeling, and the usage of these methods require knowledge of data analytics and process engineering (Nadai et al., 2017). There

is a wide range of different machine learning tasks and algorithms, and the most suitable should be selected for deployment. The two predictions presented above act as examples of both classification and regression tasks - the overload detection is purely a binary prediction of whether there is an overload or not, and the pulp production estimate predicts a continuous value. The correct task for the machine learning problem is therefore defined by what the desired output is. The best algorithm for each job depends purely on the data - for example, on some cases neural network might perform best, and on other cases random forest might.

With data analytics and machine learning models, predictions can be made to detect overloads and forecast pulp production amount. The models presented here prove that data analytics can create new value in a pulp mill environment. The methods and tools used for modeling the two example cases can be generalized to be used in the rest of the mill also. The rest of the mill's motors and other pieces of process equipment can be modeled in the same way. Because the automation system stores history data, the necessary data for modeling already exists. Therefore, there is no need for data gathering before modeling.

5 IoT application for UPM pulp mills

In chapter 4 it was discussed how new information can be refined from existing data collected from the electric motors on a pulp mill. The new information refined from the data is valuable information. In this chapter, a draft of an IoT system combining the data and new information is proposed.

5.1 System overview

The purpose of the IoT application is to bring new information for UPM pulp mill operators. It is not enough that the models exist - the full value of a predictive IoT system is harnessed only when the system is put to a real-time use. The system was requested to have the following features:

- Presentation of real-time data from motors on the mill
- Prediction of malfunctions or anomalies (such as overload) in motors
- Combining data from Metso DNA and SAP under one application
- Presentation of production estimates in real time (bonus feature)
- Self-learning: the application would itself learn the patterns of data and generate alerts (bonus feature)

The primary focus of the proposed application is to give operators information on motors which need further inspection. Therefore, instead of focusing on a single electric motor on a pulp mill, the system should be extended to take all motors into account and monitor the whole mill continuously. This requires all of the motors and the entire mill to be modeled, though. This would eventually create the mill's digital twin. Digital twins are virtual representations of real-life things, environments and systems, which form a virtual factory, where the operators can for example optimize production, adapt the product, manipulate production parameters and perform experiments on a simulated environment (Vachálek et al., 2017). These representations combine the physical industrial environment and sensor data. Digital twins consist of three parts: the physical entities (factories, production sites etc.), virtual models and data connecting the physical and virtual worlds together (Qi and Tao, 2018).

The primary source of data is still Metso DNA automation system. The data needs to flow from Metso DNA to the machine learning model and/or user interface seamlessly. This requires the removal of Microsoft Excel and its add-ins, since Excel is not suitable for data flow control. The flow of the data is presented in Figure 5.1. The data flows from Metso DNA to the model, and the result of the model is then presented in the user interface. If the data is to be presented just as a real-time information, it must also bypass the model and be directly connected to the user interface. The maintenance data from SAP or other data from any external databases should flow directly to user interface. If

the data is utilized in the prediction process, a connection to the machine learning model needs to be made.

IoT applications can be made self-learning. As the amount of data grows daily, the model can be trained with more and more data as time passes. Therefore, the model can be re-generated from time to time to gain better results. For example, a script could be configured to train the model once a week.

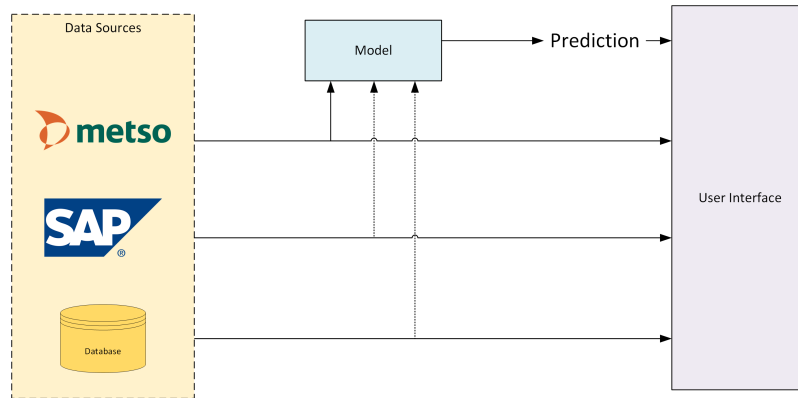


Figure 5.1: The flow of data in the proposed UPM IoT application. The data flows from Metso DNA, SAP and possible other databases/data sources to the user interface. Metso DNA data also flows to the model, generating a prediction, which flows to the user interface. The possible other connections with SAP and other sources and the model are presented with dashed connectors.

5.2 Proposed application

UPM has chosen Microsoft Azure as their IoT cloud provider. Already now UPM uploads event data to Azure cloud storage. Therefore, it is justified to build the application on Azure also. The machine learning algorithms discussed in chapter 4 can be implemented in Azure as well, under Azure Machine Learning Studio, which is a graphical, drag-and-drop tool for testing, developing and deploying predictive analytics on data (Microsoft, 2018b). The same models which were created earlier could be created in the cloud and predictions can be created in the same way as with scikit-learn.

Microsoft Azure allows all data streams to be connected and then visualized. The structure of the proposed system is layer-based, starting with mill overview and then going to more specific position-based information. The top level could feature an overview of the mill, with a "top ten" list of motors to be checked, sorted with the fault priority. This list could be refreshed, for example, once every day, and the motors to be checked are determined by machine learning models. Along the list, a crude map of the mill could be presented with markers set where the motor listed is located. With this

kind of top level, the mill operators can get valuable information with a single look of the system. The next level could be a part of the mill, for example the oxygen phase in which the overloading motor discussed in chapter 4 is located. This layer could present a more detailed map or process chart of the mill sector in question, with the markers again over the "top ten" listed motor. The deepest layer would be about the motor itself, with detailed information about current status. The user interface would present all of the measurements from Metso DNA, and recent SAP-based information about maintenance work done on the motor. The sketches for layers are presented in Figures 5.2 through 5.4.

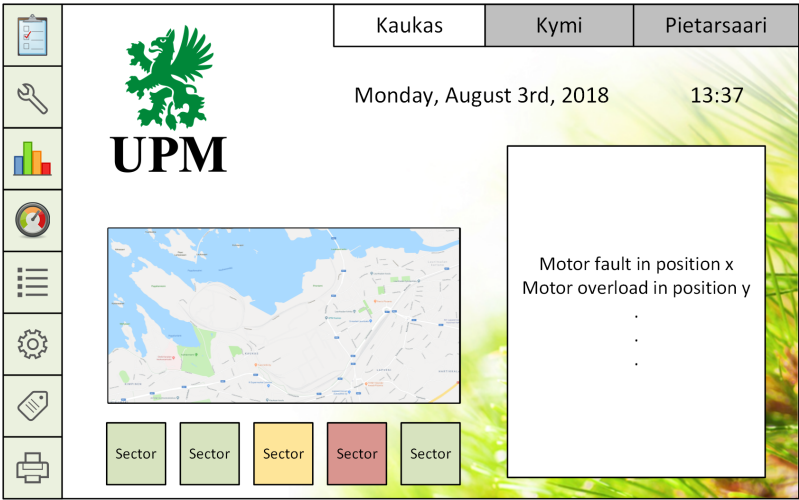


Figure 5.2: The top layer of the proposed IoT application user interface. The layer shows a crude map of the mill and its sectors, and a "top ten" -list of motors that need further inspection. *Map source: Google*

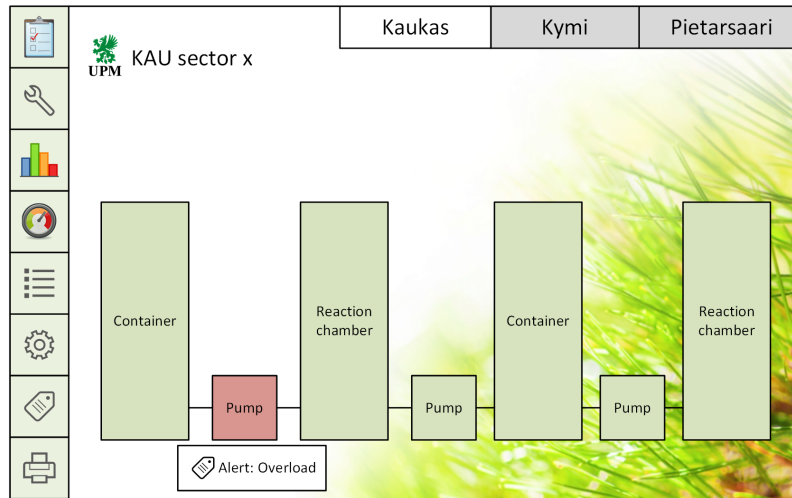


Figure 5.3: The middle layer or the proposed IoT application user interface. The layer shows a more detailed map of a sector of the mill, with tags on the motors which need inspection.

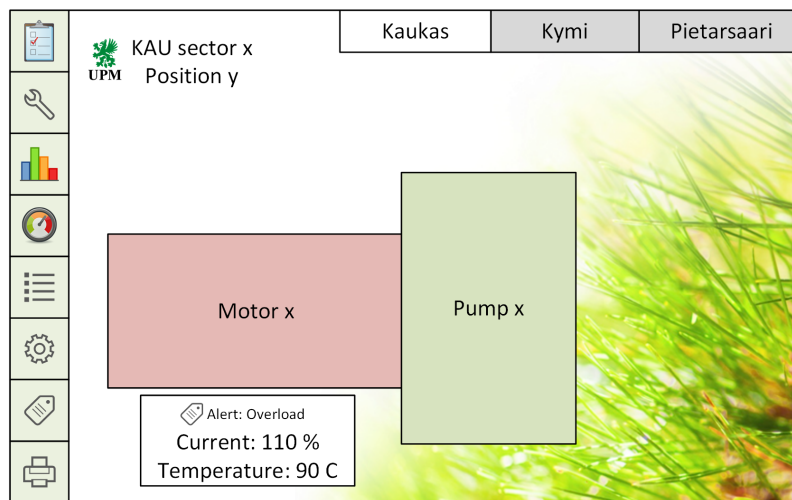


Figure 5.4: The bottom layer or the proposed IoT application user interface. The layer shows a single motor position, and detailed information on the alert, along with real-time sensor data from Metso DNA.

In addition to this layer-based approach, the mill could be represented also with a tree-like structure. The mill would be visualized in text form with collapsible sections, just like in the file manager in Microsoft Windows. The sections would be the same as the layers in the layer-based approach. This type of design is presented in Figure 5.5.

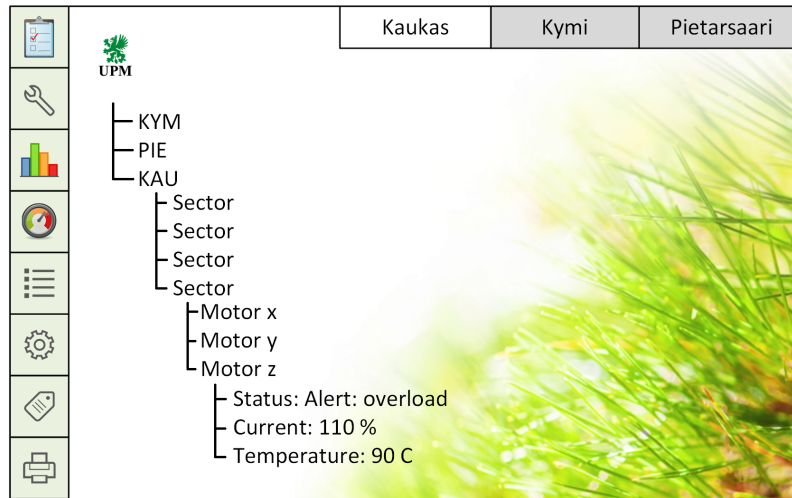


Figure 5.5: The alternative user interface of the proposed IoT application. The mill is presented in a tree-like structure, just like in file manager of Microsoft Windows.

5.3 Live demo

To test the proposed system, a live demo with Microsoft Azure was built. The demo was planned to feature real-time predictions on motor overloads, which was discussed in chapter 4. The system would act as a proof-of-concept, and be the foundation to build a mill-wide IoT system later.

Since there is no direct link between Metso automation and Microsoft Azure, to utilize the process data, a link must be built. An experimental link with a rather low bandwidth was built to connect live data to cloud. The data in Azure was stored in a Azure Blob storage, which is a feature in Azure allowing the storage of unstructured data, for example various different files (Technopedia, 2018). Besides Blob storage, the data is stored also on an SQL database. Due to the nature of the proof-of-concept demo, only the data which is needed for motor overload prediction was to be sent to Azure. If further modeling is required, it has to be done with Microsoft Excel and its add-ins because this link only sent the necessary data for the model. The link therefore is not enough for a mill-wide solution.

The model for overload prediction was created on Azure Machine Learning Studio using the same data set which was used with scikit-learn. The excel file was converted into a comma-separated file format and then uploaded into Azure. A screenshot of Azure Machine Learning Studio is presented in Figure 5.6. The screenshot presents the generation of a model for the overload prediction.

The Machine Learning Studio generated a model which was deployed as a web service. This web service can be activated using an API call - by passing input arguments to the model, it returns a prediction and a probability for it. It can also read values directly from a SQL database. The app can be configured to read the recent values from a database

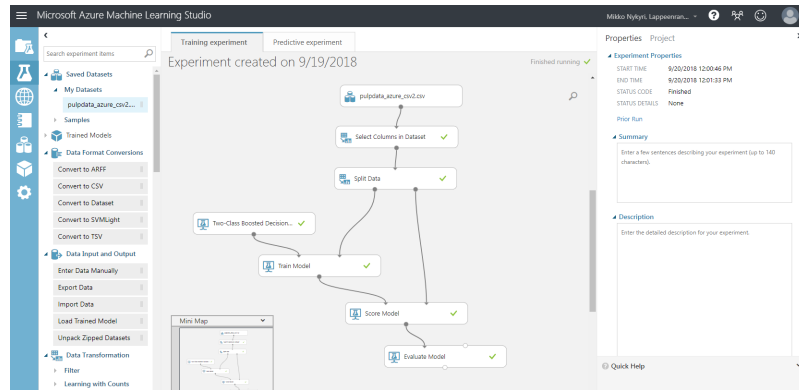


Figure 5.6: A screenshot of Microsoft Azure Machine Learning Studio, which is a graphical tool for creating machine learning models.

where the automation data from Metso DNA was uploaded with a simple SQL command. The collected newest values can then be used to generate predictions with the pre-made model. The prediction result can then be later saved to a Blob storage or SQL database.

The visualization of the data can be made with, for example, Microsoft Power BI, which is a tool used to generate graphical business or industry reports. Power BI is capable to connect to external databases, for example Azure databases to get presentable data. The reports generated with Power BI can be shared withing UPM organization and it can be accessed from multiple devices. An alternative solution for visualizing the result would be the construction of a web application running on Azure, but Power BI is preferred due to the previous experience on the environment in UPM.

5.4 Challenges & considerations

To make the system 100 % real-time, the link between Metso DNA and Microsoft Azure should be implemented. The low-bandwidth link built to realize the live demo is not sufficient enough for a mill-wide solution. The current solution of modeling with Microsoft Excel is not suitable for real-time IoT systems. Also the link between maintenance information and Azure needs to be implemented if maintenance information is to be utilized in modeling. Connections between different systems are problematic, and require manual labor unless it has been automatized. A communications standard could solve this: one possible solution to this problem is the usage of technical product information standards and APIs. With this system, all transactions would be formatted in an universal format which is easily processed by machines, thus reducing the need of manual labor.

The problem of manual work in interactions even exceeds the boundaries of UPM pulp mills. Every business transaction is made by hand, since there are no rules how transactions between systems and companies should be made. An analogy to human communication can be made - the situation is like if one person talked Finnish and other

person talked Swedish - they most likely would not understand what the other was saying.

Microsoft has pre-made IoT solutions available to deploy in Azure. These solutions are called Azure IoT Accelerators, and there are four different ones to choose from: remote monitoring, connected factory, predictive maintenance and device simulation. (Microsoft, 2018a) The solutions for remote monitoring and predictive maintenance were tested, but eventually rejected because of their complexity and poor/difficult modifiability. The solutions itself are more suitable for proof-of-concept works, rather than skeletons to build own solutions on.

Since Kaukas mill alone has nearly 4000 electric motors, the implementation of a mill-wide predictive model requires a lot of modeling to be done and hiring data analysts. The need of analysts is pushed even higher, when Kymi and Pietarsaari mills are taken into account. However, investing in prevention of production halts may pay off in the long run. Guaranteeing the continuity of production will save money, if the motors can be repaired beforehand, during planned production halts.

6 Conclusion & discussion

The emerge of Industrial Internet of Things is part of the fourth industrial revolution. The industrial automation systems, which emerged in the 1970s, are now being replaced with sensors connected to each other and the Internet. The collected data can be refined into new information with machine learning and data analytics.

The main results of this thesis are:

- Machine learning models can be created from pulp mill automation data and the models can make predictions, which create new information for mill operators.
- The data for creating the models exists already, because the automation system stores this data from the past.
- The tools for building a mill-wide IoT system exist and building such system requires wide-scale knowledge about data analytics and process engineering.

The data gathered from electric motors in a pulp mill can be used to gain new information about faults and process bottlenecks. As an example, machine learning models were created for electric motor overload detection and prediction and pulp production amount prediction. The two predictions were selected due to their different nature, overload prediction being a classification task and production amount prediction being a regression task. The motor in question was selected for further analysis due to frequent overload alerts originating from it. The predictive model was created using offline excel data to detect overload in real time and predict overload ten minutes into the future. In addition to predictions on possible overload, the information about frequently overloading motors indicate possible bottlenecks in the pulp mill. This might be due to misfitting the motor, or due to the increasing production amount over the years. The motor might have been properly sized in the past, but today, when the mill produces higher amount of pulp, the motor became too small. Further data analytics can give answers to whether or when this motor should be replaced or serviced. The production amount prediction model was trained to predict production amounts in real time, 10 and 60 minutes and 12 hours into the future.

In this thesis, the possibilities of data analytics and machine learning in a pulp mill environment is confirmed with the models generated for the overload detection and prediction. However, overload is not the only anomaly or problem with electric motors in pulp mills. To gain further information on motor maintenance needs, the fault data could be combined with the automation system data. A motor can have multiple models - for example one model for overload prediction, one for overheat, one for vibration etc. The amount of anomalies to detect and predict is limited only by the structure and level of detail on the fault reports. If the reports contain detailed information about each fault, and there are enough instances of such failure with such information, modeling can be possible.

An IoT application draft for UPM pulp mills was proposed. The application would contain machine learning models of all the electric motors on a pulp mill, and it would predict faults and anomalies and present a list of motors to be inspected for the mill operators. A demonstration was made with Microsoft Azure as a proof-of-concept work. The demonstration featured a real-time data flow from a motor to a machine learning model predicting motor overloading. Eventually the application would form the mill's digital twin, which is an virtual representation of the whole pulp mill. This virtual mill, a digital twin of the real mill, could then forecast production amounts and predict faulty components of the system. It could also be used to run simulations in the factory, to experiment between input values to find the best output and improve the mill's productivity.

Although the automation and data gathering in UPM Kaukas works well, getting the data in a large scale from the automation system for later examination is a significant job and requires manual labor with obsolete tools. Database access requires the usage of a proprietary Microsoft Excel add-in, which Excel 2016 considers a security threat, is clumsy and inefficient. The add-in dumps the data in a excel file. Microsoft Excel as a tool for migrating data from an automation system to a data analysis tool, such as Python, is not efficient and optimal. Instead of proprietary Excel add-ins, an API could be implemented to get the data straight to data analysis tool. This solution would serve both offline and online data analysis tools' needs, and would cut the unnecessary middleman between systems. Currently, this Excel layer of the system needs to be operated by a person, although all of this could be automated. This issue was tackled during the research, when a link was established to send selected pieces of data into Microsoft Azure cloud. However, a system-wide solution is not yet built and retrieval of data relies on Excel, because the built link has so low bandwidth that it is incapable of handling data on a mill-wide scale.

Integration of different systems is a hot topic among industries worldwide. Even in UPM mill scale, the integration between different systems is challenging, since the different systems, such as Metso DNA automation system, SAP and Microsoft Azure are not plug-and-play connectable. When taking larger scale into consideration, for example business-to-business transactions, the challenges are present even clearer. A solution for integrating things is universal API:s. Sharing data between companies and systems is crucial - and API:s and standardized communication protocols make it possible to automate these transactions between businesses, machines and people.

Unfortunately, data also brings problems with it. Data is wanted material - IoT systems are common targets for cybercriminals. When connecting any system, device or thing into Internet, proper precautions ought to be made to make sure that the flow of data is safe and certain. New sources of data create new problems. One good example is the ownership of data: if data is shared between stakeholders, it is often not certain who owns the data and who decides, for example how it is kept safely, to whom it is shared and how it is destroyed when necessary. A possible solution for data security is blockchain.

This thesis forms the base for an IoT application to be used in UPM pulp mills. The technology to build a mill-wide IoT system is available, and the next steps to realize the system would be to model more motors and build an application in Azure to display these predictions and values from the mill. In addition to the need for modeling, investments in existing automation systems and networks are required to acquire a higher-bandwidth link between Metso DNA automation system and Microsoft Azure. The investments would allow to bypass excel and have all the data of all of the mill's motors in the cloud. The current solution does not combine the fault data from SAP to the model making process, but this data could also be utilized. The ultimate goal is to integrate all data into one system, from which all of the necessary functions and information can be seen, including sensor data and a list of predictive maintenance predictions across the mill.

References

- Anderson, N., et al. (2017). *The Industrial Internet of Things Volume T3: Analytics Framework*. url: <https://www.iiconsortium.org/industrial-analytics.htm>.
- Antonino-Daviu, J., López, A.Q., Rubbiolo, M., and Climente-Alarcon, V. (2017). Diagnosis of the rotor condition in electric motors operating in mining facilities through the analysis of motor currents. In: *2017 IEEE Industry Applications Society Annual Meeting*, pp. 1–8.
- Bakshi, K. and Bakshi, K. (2018). Considerations for artificial intelligence and machine learning: Approaches and use cases. In: *2018 IEEE Aerospace Conference*, pp. 1–9.
- Christidis, K. and Devetsikiotis, M. (2016). Blockchains and Smart Contracts for the Internet of Things. *IEEE Access*, 4, pp. 2292–2303. ISSN 2169-3536, doi:10.1109/ACCESS.2016.2566339.
- Cline, G. (2017). Industry 4.0 and Industrial IoT in Manufacturing: A Sneak Peek. url: <http://www.aberdeenessentials.com/opspro-essentials/industry-4-0-industrial-iot-manufacturing-sneak-peek/>.
- Diab, W.W., Harper, K.E., Lin, S.W., and Sobel, W. (2017). *Industrial Analytics: The Engine Driving the IIoT Revolution*. url: https://www.iiconsortium.org/pdf/IndustrialAnalytics-the_engine_driving-IIoT_revolution_20170321_FINAL.pdf.
- Gatteschi, V., et al. (2018). To Blockchain or Not to Blockchain: That Is the Question. *IT Professional*, 20(2), pp. 62–74. ISSN 1520-9202, doi:10.1109/MITP.2018.021921652.
- Habib, M.R., Hassan, M.A., Seoud, R.A.A., and Bayoumi, A.M. (2016). Mechanical fault detection and classification using pattern recognition based on bispectrum algorithm. *Lecture Notes in Networks and Systems Advanced Technologies for Sustainable Systems*, p. 147–165. doi:10.1007/978-3-319-48725-0_15.
- IBM (2018). IBM SPSS Modeler Product Overview. url: <https://www.ibm.com/products/spss-modeler>.
- IEEE (2015). Towards a definition of the Internet of Things (IoT). url: <https://iot.ieee.org/>.
- Kanawaday, A. and Sane, A. (2017). Machine learning for predictive maintenance of industrial machines using IoT sensor data. In: *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 87–90.
- Karmakar, S., Chattopadhyay, S., Mitra, M., and Sengupta, S. (2016). *Induction Motor Fault Diagnosis*. SPRINGER.

- Korpela, K., Hallikas, J., and Dahlberg, T. (2017). Digital Supply Chain Transformation toward Blockchain Integration. *Proceedings of the 50th Hawaii International Conference on System Sciences (2017)*. doi:10.24251/hicss.2017.506.
- scikit learn (2018). scikit-learn Overview. url: <http://scikit-learn.org/stable/index.html>.
- Microsoft (2018a). Azure IoT Accelerators Overview. url: <https://azure.microsoft.com/en-us/features/iot-accelerators/>.
- Microsoft (2018b). What is Azure Machine Learning Studio? url: <https://docs.microsoft.com/en-us/azure/machine-learning/studio/what-is-ml-studio>.
- Mihai, V., et al. (2018). Wireless Sensor Network Architecture based on Fog Computing. In: *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 743–747.
- Nadai, N., Melani, A.H.A., Souza, G.F.M., and Nabeta, S.I. (2017). Equipment failure prediction based on neural network analysis incorporating maintainers inspection findings. In: *2017 Annual Reliability and Maintainability Symposium (RAMS)*, pp. 1–7.
- nodered.org (2018). Node-RED Overview. url: <https://nodered.org/>.
- Nykyri, M., Korpinen, K., and Jäppinen, J. (2017). Raspberry Pi -pohjaisen IoT-anturijärjestelmän suunnittelu. url: <http://www.doria.fi/handle/10024/136452>.
- Pandas (2018). Pandas Overview. url: <https://pandas.pydata.org/>.
- Qi, Q. and Tao, F. (2018). Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison. *IEEE Access*, 6, pp. 3585–3593. ISSN 2169-3536, doi:10.1109/ACCESS.2018.2793265.
- Ransom, D.L. and Hamilton, R. (2013). Extending Motor Life With Updated Thermal Model Overload Protection. *IEEE Transactions on Industry Applications*, 49(6), pp. 2471–2477. ISSN 0093-9994, doi:10.1109/TIA.2013.2265375.
- Sasikala, B.S., Biju, V.G., and Prashanth, C.M. (2017). Kappa and accuracy evaluations of machine learning classifiers. In: *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pp. 20–23.
- srgresearch.com (2018). Cloud Growth Rate Increased Again in Q1; Amazon Maintains Market Share Dominance. url: <https://www.srgresearch.com/articles/cloud-growth-rate-increased-again-q1-amazon-maintains-market-share-dominance>.

Swamy, S.N., Jadhav, D., and Kulkarni, N. (2017). Security threats in the application layer in IOT applications. In: *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 477–480.

Technopedia (2018). What is Blob Storage? <https://www.techopedia.com/definition/32166/blob-storage/>.

upmbiofore.com (2017). UPM Kaukas Leads the Way in Promoting the Circular Economy. url: <https://www.upmbiofore.com/upm/upm-kaukas-leads-the-way-in-promoting-the-circular-economy/>.

Vachálek, J., et al. (2017). The digital twin of an industrial production line within the industry 4.0 concept. In: *2017 21st International Conference on Process Control (PC)*, pp. 258–262.

l. Zhong, C., Zhu, Z., and Huang, R.G. (2017). Study on the IOT Architecture and Access Technology. In: *2017 16th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES)*, pp. 113–116.

