KTH
VETENSKAP
OCH KONST

The Use of Machine Learning in Industrial Quality Control

Thesis by
Erik Granstedt Möller

for the degree of
Master of Science in Engineering

Stockholm, Sweden, 2017

Abstract

This study investigates the possibilities and the limitations for utilizing machine learning algorithms in industrial quality control. This has been done by a number of steps: Quality improvement practitioners have been interviewed about the difficulties and problems they encounter. Machine learning has been studied through literature review to understand what kind of problems machine learning may solve and what would be needed in terms of resources. Then quality control issues have been paired with machine learning solutions and discussed in terms of resource requirements. There has also been a construction of a machine learning algorithm for solving a specific quality control problem for Scania, acoustic deviation detection, where the resources and usefulness were studied first hand. The readiness of machine learning as a tool in industry is discussed. Finally, a model has been synthesized for evaluating the feasibility of machine learning projects in quality control.

Sammanfattning

Denna studie har undersökt möjligheterna och begränsningarna för att använda maskininlärning i industriell kvalitetskontroll. Detta har gjorts genom ett antal delmoment: Utövare inom kvalitetsförbättringar har intervjuats om de svårigheter och problem de möter. Maskininlärning har studerats genom litteraturstudie för att förstå vad för slags problem maskininlärning kan lösa och vilka resurser som behövs. Kvalitetsproblem har parats ihop med maskininlärningslösningar, och diskuteras med hänsyn till resurskrav och genomförbarhet. Vidare har en maskininlärningsalgoritm konstruerats för att lösa ett specifikt kvalitetsproblem åt Scania, detektering av ljudkvalitet, där resursbehov och användbarhet har studerats på nära håll. Det diskuteras hur redo maskininlärning är att användas som ett verktyg i industrin. Slutligen har en modell sammanställts för att utvärdera genomförbarheten av projekt i maskininlärning inom kvalitetskontroll.

Contents

1. Introduction

1.1 Background

Automation is the use of control systems to operate machines and processes. This has historically had a big influence on industrial production, with important milestones such as the automatic control of the steam engine, continuous chemical production, and industry robots. As information technology has developed, computers and utilization of information has had a greater importance in automation. Modern industries often use multiple IT systems for tasks such as process control, tracing products, and handling capacity. As companies accumulate larger amounts of data, the field of automating data analysis has grown; Computer programs using so called machine learning algorithms are being developed to make better predictions and decisions from data.

Quality Control is the task of assuring that the products produced reach a certain standard, either set by the company or by the customers. The field developed rapidly during the second half of the 20th century and is today an integral part of most manufacturing companies. It serves multiple purposes: By measuring what is important to a product or component, bad components can be excluded before they are delivered to the customer. Furthermore, processes can be controlled so that quality deviations occur less frequently, which results in higher quality and lower costs.

The development within Machine Learning has been rapid the last few years, and has especially been applied in IT for improvements in search engines, speech recognition, and online marketing. There is untapped potential in manufacturing industry using this kind of software based automation, and it is not obvious how it can be used, where it is useful, or how to implement it.

1.2 Problem statement

There is an uncertainty in how machine learning algorithms can improve industrial quality control in terms of improved quality and reduced costs, and an uncertainty about the requirements to implement these algorithms.

1.3 Purpose

The purpose of this study is to investigate the possibilities and limitations machine learning has on industrial quality control.

1.4 Research question

Of what use can machine learning be for industrial quality control, and what limitations are there in terms of resources, reliability and precision?

- What kind of machine learning is suitable for what kind of problems?
- What resources are needed in terms of time, money, competence, and computational resources?
- What result may this lead to regarding quality and cost?

1.5 Expected contribution

The study is expected to contribute with a procedure for decision about implementation of machine learning in quality control, by creating a model of what is required and how it will contribute to a successful implementation.

1.6 Delimitations

The construction of a machine learning program will be limited to known algorithms and configuration, as I do not seek to develop new, but focus on the aspects of application. I choose to limit the width of the scope to study one particular quality issue (engine sound detection), although conceptually broadening my findings to further quality issues. This is because of the depth required to construct and train a machine learning algorithm. There are applications where using multiple algorithms together may be advantageous. For the same reason, I choose to exclude this from my study.

There could be potential in matching all kinds of errors with audio, even errors that do not create hearable noise, to investigate whether a computer can hear errors which humans cannot. That is, however, beyond this scope.

The use of machine learning will be limited by quality control and quality engineering. The use of machine learning for optimization of production parameters is briefly introduced but not studied.

1.7 Method

First, quality control problems were identified through theory study, participation in testing at Scania as well as interviews with quality personnel at Scania. These have been matched up with potential machine learning algorithms. The participation included working in the final testing department at Scania and observing the problems they face. Interviews were conducted semi-structured to gain the interviewees perspective of quality control.

Second, a machine learning algorithm has been developed for Quality Control at Scania, where the problem at hand was to detect engine sound anomalies. The resources required for this in terms of time, cost, competence and computational resources has been documented and analyzed. As part of the preparation, quantitative sound data has been collected.

Third, a model has been proposed to generalize the findings of resource requirements to other type of quality control problems as well. It has been done by analyzing the situational factors affecting each step in the development of the machine learning algorithm, and connecting them with resource requirements.

The study has been inductive, where the starting point has been the empirical results, which then together with theory has been combined to form new theories. The machine learning algorithm was based on quantitative data including audio data and quality data from a large number of engines. The problems and use for machine learning in quality control has been based on qualitative interviews.

2. Industrial Quality Control

Quality Control (QC) is the practice of ensuring and improving quality of a company’s products and services. For consumers, the quality of a product is often considered to be the “fitness for use” of the product (Montgomery, 2013). This can be attained by a good design, and by conformance to that design. In QC, the focus is on the latter: to make the products conform to the intended design by reducing variability in processes and products. From here on, quality refers to conformance to design.

The costs associated with quality can be categorized into prevention costs, appraisal costs, internal failure costs, and external failure costs (Montgomery, 2013). Generally, the total quality costs are reduced by increasing prevention and appraisal costs, since that reduces failure costs even more. Besides reducing costs, high quality can be of strategic importance since it may increase the business reputation among customers and employees, which can strengthen the company’s market position and competitiveness.

The three major methods of quality control are Acceptance Sampling, Statistical Process Control, and Experimental Design (Fountoulaki, et al., 2011) When testing is expensive, time consuming and or destructive, there are advantages with testing only a sample of products to draw conclusions about the whole batch. This is called Acceptance Sampling.

Process control is somewhat different. It sets out to monitor and identify the cause of variation in a process. By noting the distribution of product measurements when a process is well functioning, later measurements can be compared to this. By calculating the probability that each measurement or group of measurements is a sample from that same distribution of a well functioning process, one can determine if there is something wrong with the production process. The goal is not only to find defective parts, but to detect when something is wrong with the process, to be able to correct it as soon as possible. There are many well used tools to visualize the process, and to structure the analysis of what may be wrong, including Cause-and-Effect Diagram, Control Charts, Scatter Plots and Histograms.

Experimental design is mainly practiced when setting up a production process or introducing a new product. The method uses statistical experiments to find good parameters for the production process. It is done by making changes in the input parameters and observing the output of the production (Fountoulaki, et al., 2011). It can also be used to compare different production lines, production techniques, or to evaluate the functionality of design features.

Most studies within the field of quality control concerns process control. Many of these set out to improve common tools for specific purposes.

Yeong et al (2017) exemplify the use of Coefficient of Variation in control charts, as well as show how it can be used more accurately in a variable parameters chart. Coefficient of Variation is the ratio of standard deviation to the mean of a process, and can be a better indicator of process behavior than standard deviation and mean separately, if the standard deviation is a linear function of the process mean.

Teoh et al (2016) write about how adaptive control charts are used to enhance the sensitivity in assignable cause variations. The adaptive control chart examined is the variable sample size chart, where there are warning zones between in-control and out-of-control which signals to increase the sample size. They show how to optimize the sample size based on median run length.

When control charts are set up on a new process, the first step is to measure the process variation to determine mean and standard deviation to control. Diko et al (2017) derive new charting constants to determine how many measurements must be done in this initial phase, both analytically and numerically.

The capability of a process is its ability to produce parts within tolerance, i.e. requirements on the product. By reducing the variability in the production process, more parts will be produced within tolerance, and the scrap rate can be reduced. For products comprising a vast number of components, this is especially important, since one faulty component may be enough to make the product not function properly, and so the number of faulty final products is much higher than for each component (Montgomery, 2013).

Some resent studies in the field of process capability include Ganji & Gildeh (2016) who investigate how to index capability when the specifications are asymmetric: the upper and lower tolerances are not on equal distance from the nominal value. Weusten & Tummers (2016) study how to set preliminary specification limits based on the performance of the process, to be able to estimate a capability index.

Regarding implementing new tools and systems such as machine learning in quality control, a consideration is change resistance among employees. Sim & Rogers (2008) discuss resistance to implementation of improvement programs, where the main contributing factors are aging workforce and low commitment from management.

3. Machine Learning

Machine learning, a subfield of artificial intelligence, is the science and application of algorithms that use data to make predictions and decisions (Raschka, 2015). Instead of setting up rules manually for analyzing data, machine learning algorithms can numerically approach rules and connections in data. There are three main categories of algorithms for different kind of problems: Supervised learning, unsupervised learning, and reinforcement learning.

3.1 Supervised machine learning

Supervised machine learning solves the type of problems where there is historical input data and corresponding solutions. There are mainly two types: Classification problems, and regression problems (Raschka, 2015). For classification problems, the solution to each instance is a category of a set of predetermined categories. An example of classification problem would be to determine if a photo has a person in it. The categories would be having a person, or not having a person. Another classification problem would be to determine the genre of a song, where the genres would be the predetermined categories. For regression problems, instead the solution is a numeric value on a continuous scale. An example would be to predict the air temperature given the location and time. See figure 1 for a visual comparison.

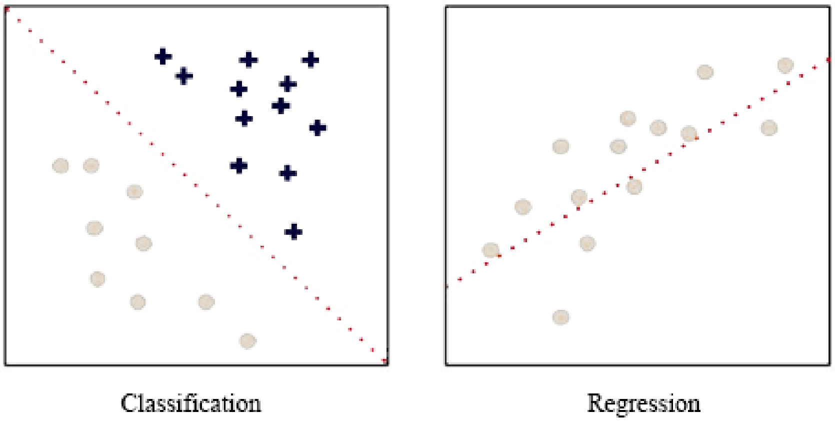

Figure 1 – difference between classification and regression (Oakes, 2016)

There are two phases of a supervised algorithm. The first one is training the algorithm, which tries to create a function from the known input data to the known output data. This is done numerically. During this phase, the algorithm is fed with historical data, and returns the function which maps input to output data. During the second phase, new input data is fed to the algorithm, and it returns categories for classifiers, and numerical values for regression (Raschka, 2015).

To evaluate how well a classification algorithm performs, accuracy is commonly used. It is a measure of proportion of correctly classified entries. This is achieved by using input data with known output data, which the algorithm has not been trained on.

3.2 Unsupervised learning

Unsupervised learning is the categorizing of data, without knowing the categories of previous data. One common method is clustering. There are multiple clustering algorithms, but the general principle is to categorize instances based on similarities in data. One example would be segmenting a market into categories based on the customers' buying patterns. Another type of unsupervised learning is anomaly detection, which attempts to detect outliers in the data. It tries to identify whether an observation comes from the same distribution as previous data. This concept is similar to process control in quality practices. The main difference is that in a machine learning algorithm, the distribution is defined by the computer, and can therefore be immensely complex. One approach to this is to manually identify the features in which anomalies may occur, to reduce the dimensions (Ng, 2013). Anomaly detection can be used together with classification algorithms, to identify classes with little data (Dunning & Friedman, 2014). Combinations of multiple algorithms can be powerful (Ko, et al., 2016), however, constructing these is more complex and time consuming, as multiple algorithms must be chosen and integrated. For that reason, this paper will only consider one algorithm at the time.

3.3 Reinforcement learning

Reinforcement learning is fairly different from the other methods. The algorithm attempts to maximize some kind of reward, by acting in an environment and receiving feedback from the environment. The environment is often formulated as a Markov Decision Process, where the environment is in some state, and an action will move the environment into a new state. An example would be a chess playing algorithm. For each state of the board, there are multiple actions to take (moving the pieces), and they all yield new states (the board after the opponent moves). The reward could be the advantage in pieces and position, and the ultimate reward is check mate.

Quality is often concerned with identifying and separating bad parts from good, and that is the case for engine audio as well. For this reason, this theory chapter will expand further on classification algorithms.

3.4 Perceptron

One of the simplest of machine learning classification algorithms, which many others are built upon, is the Perceptron, which mimics a neuron in brains; see figure 2 of perceptron flowchart. Each feature in a sample is multiplied by a weight, and the weights are summed, and put through an activation function. For the perceptron, the activation function is a step function where a threshold number determines which class the sample belongs to. The goal of the training phase is to change the weights such that each sample corresponds to its known class. The weights are updated with the learning function, which takes the difference between the output and the desired output, multiplied by a learning rate (Raschka, 2015).

$$\Delta w_j = n\left(y^{(i)} - \hat{\mathrm{y}}^{(\mathrm{i})}\right)\mathrm{x}_{\mathrm{j}}^{(\mathrm{i})}$$

Where w is used to update the weight vector, such that $w_j \coloneqq w_j + \Delta w_j$, $y^{(i)}$ is the class label, ŷ is the predicted class label, and n is the learning rate. The $x_j^{(i)}$xji is the input value. The weights are updated after each sample, in what is called an online learning. The perceptron will only find a solution if the classes are linearly separable. By changing activation function from the step function, better results can be reached.

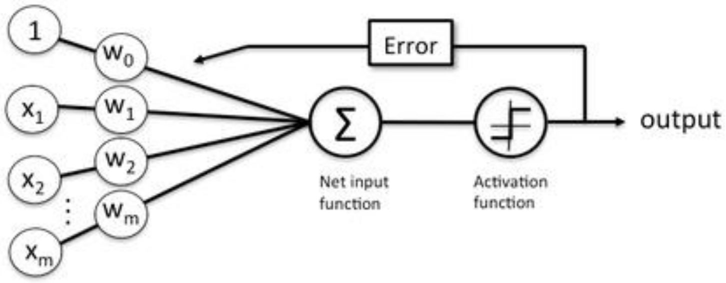

Figure 2 - Perceptron flowchart (Raschka, 2017)

3.5 Neural networks

Neural networks are networks of neurons, using parallel neurons. Deep neural networks also use sequential neurons, creating layers of neurons between input and output, see figure 3. Using parallel neurons allows for different inputs to be weighted differently for their influence on class belonging. Using sequences of neurons allows complex non-linear functions to be created. The outputs of some neurons are simply features for the next neuron downstream. Neural networks have been increasingly popular in many applications because of their ability to model complex functions (Raschka, 2015).

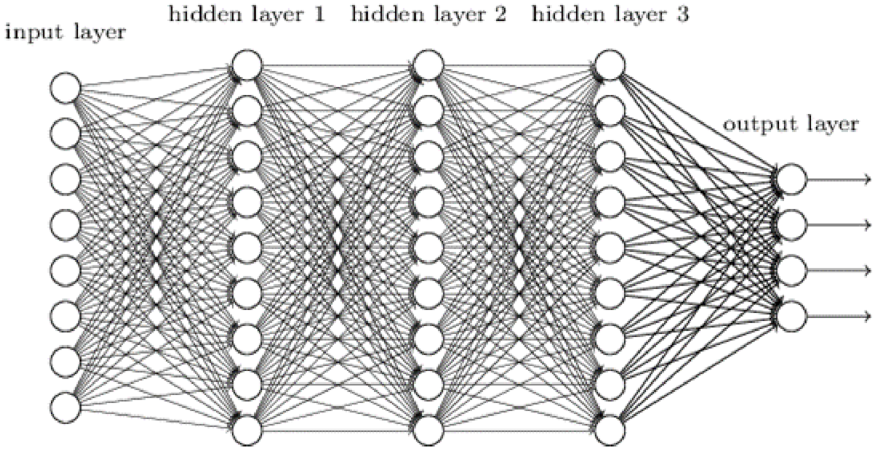

Figure 3 – Deep Neural Network (ODSC Team, 2016)

3.6 Activation function

The activation function can be configured in multiple ways. It has the input of the sum of weights times the feature values, and outputs a value representing the class estimation. Instead of using a step function as in the perceptron, a smoother function can be better to adjust for the distance of the error. A common activation function is the sigmoid function. In contrast to the perceptron, the max and min values are asymptotes, meaning even fairly good estimates will have a small error.

$$f(x) = \frac{1}{1 + e^{-x}}$$

The sigmoid function, a common activation function

3.7 Gradient descent

When building networks of neurons, there are weights for each layer to estimate. This is commonly done through what is called gradient descent. The error is defined as the difference of the estimate and the desired output. To minimize this error, the gradient for each weight is calculated. Then, a step is taken in the opposite direction of the gradient. This requires the activation function to be increasing. The weights are updated and the procedure is repeated.

3.8 Imbalanced data

Having an imbalanced amount of data, such that the classes are significantly different in size, may cause problems. As the algorithm used may favor the larger classes during training, the final algorithm may be bad at detecting the more unusual classes. There are a few ways of dealing with this problem. The first would be to collect more data, to even the classes out. If this is not possible, one could duplicate existing data from small classes, or delete data from larger classes. This is called oversampling and undersampling. A risk with oversampling is overtraining, and a risk with undersampling is that the algorithm will be undertrained. Another way of dealing with this problem is to synthesize samples of the classes that are small. One popular method is Synthetic Minority Over-sampling Technique, which works by creating a synthetic sample somewhere in the feature space between one sample selected at random, and one of its k-nearest neighbors, see figure 4.

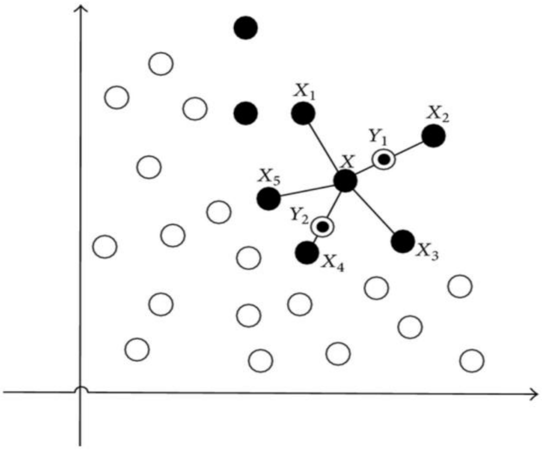

Figure 4 – Synthetic Minority Over-Sampling Technique (SMOTE), where Y_1 and Y_2 are created from combinations of X (Xie, et al., 2015)

3.9 Confusion Matrix

One way of measuring how well an algorithm performs is by using a Confusion Matrix. When studying only two classes of data, each instance can be from either class, and can be estimated to belong to either class. This creates four different scenarios, as shown in table 1. To evaluate an algorithm, using the confusion matrix can be a good tool for understanding what is wrong. For example, it can indicate which class seems to be harder to detect or if the algorithm prefers one class over the other. This is especially interesting in the context of quality, where it is important to control the tendency of type I and type II errors, that is, the risk of classing a bad part as good, and a good part as bad, respectively. Depending on the cost of quality priorities, this may look different for different companies. For high quality products such as trucks, minimizing type I error is much more important than type II error. This is because a type I error means the error is not detected, and so the product may be delivered to the customer, where it will be unsatisfactory. A type II error, in contrast, will be investigated for reparation, but later correctly classified (Montgomery, 2013).

Table 1 – Confusion Matrix with four Scenarios

	Actually positive	**Actually negative**
Predicted positive	Correctly approved	Type I error
Predicted negative	Type II error	Correctly rejected

3.10 Tuning a Neural Network

For a machine learning algorithm to learn how to transform input to output, just showing training data is not enough. A suitable algorithm has to be selected, as discussed. Next step is to build a program which can adjust the weights, to learn. Gradient descent has been mentioned as a method for determining the direction in feature space to step in order to reduce the error. This is a numerical optimization problem in high dimension, and has to be calibrated. One parameter to adjust is the learning rate, which is the step size taken in the direction of the gradient. This affects how fast the program converges, whether it will constantly overshoot the target, and whether the program can get stuck in local minima. Similarly, the batch size, how many of the samples are trained on at once, affects the speed of convergence, as well as possibility to overcome local minima due to noise. Other parameters to tune were L2 regularization which is punishment for overfitting, and number of iterations. There is no right way to do this, and this is done iteratively after each training and evaluation, before training again. See figure 10 in section 8. In connection to this, the network architecture may also be considered a parameter for tuning, such as the node configuration. There have been advancements in how to integrate the parameter tuning and network architecture into the machine learning phase, although this is not in wide use (Cortes, et al., 2017).

3.11 Convolutional Neural Networks

Convolutional Neural Networks (CNN's) are networks developed for image sources of input. Instead of looking for patterns between any input features, the CNN will look for patterns in features that are close by. This requires that the input data is not just one vector, but organized in a two-dimensional coordinate system, such as an image with pixels along x- and y-dimensions. This allows the CNN to identify nearby features in both x- and y directions. A common task is to classify a photo based on the objects in the photo. Since an object can appear at any x- and y-position, the CNN approach is to try all x- and y-positions. This is done by only looking at a small part of the image at the time, and determining whether the object is there. The multiple layers in a CNN build up small features such as edges or patterns, and further downstream combinations of small features to images that reminds of objects, see figure 5.

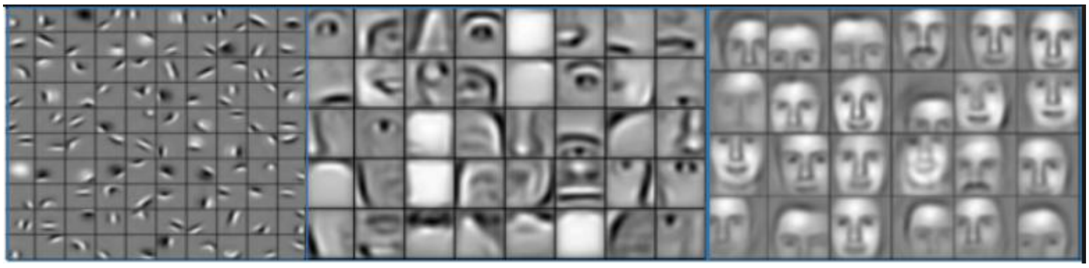

Figure 5 – Convolutional Neural Network filters in three levels detecting faces (Strong Analytics, 2016)

3.12 Recurrent Neural Network

Recurrent Neural Networks (RNN’s) are neural networks where the nodes are not only connected from input towards output, but where some nodes connect back to nodes upstream, creating cycles. See figure 6. This enables networks to process sequences of input vectors, and consider previous vector inputs as well as the current. This makes them especially useful for data which is related to previous data, such as words in sentences, stock prices over time, or sound over time. One well known version is the LSTM network, where the memory is especially good at considering long-term dependencies (Britz, 2015).

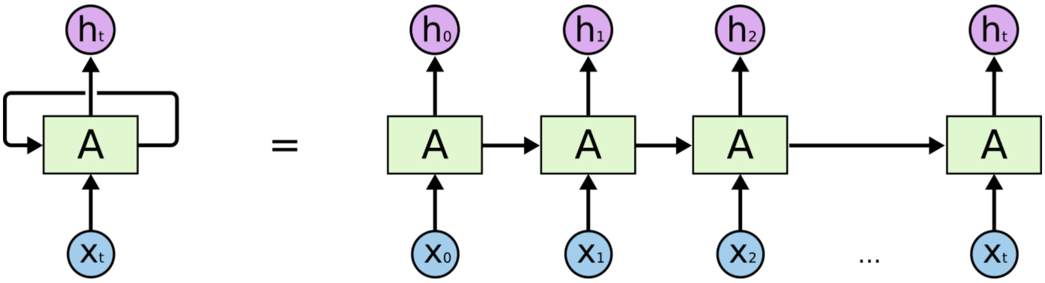

Figure 6 – Recurrent Neural Network, where X_i is input vector, h_i is output, and A is node structure. (Olah, 2015)

3.13 Predicting sample size requirements

In classification, the larger data set available for training, the higher is the expected outcome of accuracy. Accuracy can then be described as a function of sample size. How this function looks is interesting to know because in a typical application there is a required accuracy from the beginning, which directs how much data has to be collected, and or if the project is feasible. Figueroa et al (2012) showed that this can be modeled as the inverse power law, using only a few test runs of different sample sizes, to predict the accuracy on larger sample sizes.

$$Y(x) = f(X; a, b, c) = (1 - a) - b * x^c$$

The variables a, b, and c, depend on the specific dataset and algorithm, and have to be found anew for each problem. This is done by a least squares approximation from a number of test runs. In their study, they used sample sizes between 53 and 280 to predict accuracy for greater sample sizes.

4. Machine Learning and Quality Control

Machine learning as a tool has been applied in many different environments. Quality control has to some extent been examined as well, specifically process control.

One field that has been studied is control chart patterns such as shifts, trends, and cycles. Detecting patterns is important to minimizing variation by identifying the root cause and adjusting the process accordingly. Guh & Shiue (2005) found that decision trees could be used to identify control chart patterns. Wang, et al. (2008) also managed to recognize anomaly types using decision trees. Gauri & Chakraborty (2007) trained a deep neural network to find control chart patterns and features.

Other studies set out to determine whether machine learning can be used to classify processes in or out of control. Smith (1994) used a neural network for X-bar and R charts to determine if a process was out of control. For large shifts in mean, she managed to detect equally well with neural networks as with regular control limits, but for large shifts, neural networks outperformed control limits. Shao & Chiu (1999) trained a neural network to identify different assignable causes, in an attempt to integrate statistical process control with feedback control of a few parameters. Most of these studies use simulated data.

Pacella & Semeraro (2007) highlight the problem that many quality characteristics are correlated, and use a recurrent neural network to monitor quality on autocorrelated process data. Low et al (2003) also consider autocorrelated data, but specifically focus on variance out of control in their proposed neural network procedure.

There have been some practical implementations as well. There are a few successful studies published on quality of fruit detected with visual machine learning algorithms (Pandey, et al., 2013) (Sa, et al., 2016). Plastic injection molding has also been studied, with production parameters as inputs (Ribeiro, 2005) (Tellaeche & Arana, 2013).

Optimizing process parameters with machine learning can be considered experimental design. The field of diagnosis in artificial intelligence concerns determining whether a system is behaving correctly, and if not, what the reason for that is. These are very similar in the sense that both model the production system, but the main difference seem to be that diagnosis focuses on cause analysis, while experimental design focuses on finding the optimum. Machine learning algorithms are especially useful to model complex and unknown functions. Often in chemical industry, there is good understanding for the processes already, and for that reason, machine learning may not be needed. Quality is often ensured with a feedback controller such as a Model Predictive Control. Machine learning to understand the process has successfully been used in assembly, although the requirements of product and production data are very high. Bosch held a competition in 2016 where they shared production data with an online community of data scientist, who tried to classify manufacturing failures (Bosch, 2016). Out of 1373 contestants, the best submission reached a Mathew Correlation Coefficient of 0.52. The winners used XGBoost and Random Forests for their algorithm.

5. Audio processing

Audio processing is a subfield of signal processing, which concern analysis and modification of signals. This field is vast and in this section I will cover the audio processing that is commonly used as preprocessing for machine learning purposes.

5.1 Audio Sampling

There are multiple steps of signal processing needed to make sense of audio data in a way that humans do. A digital audio recording in its rawest form (.WAV) is a vector of air pressure measurements, equally spaced in time. As we want the machine learning algorithm to treat the data similar to how humans comprehend sound, we want these measurements represented in frequencies.

5.2 Frequency representations

For static sound, sound that does not shift pitch over time, a Discrete Fourier Transform is appropriate. This is a transform that will approximate the discrete values with cosine functions at given frequencies, see figure 7.

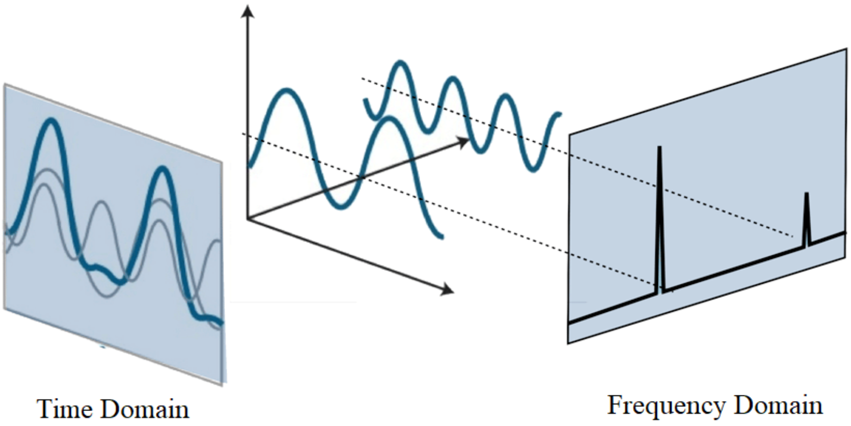

Figure 7 – Fourier transform from time domain to frequency domain (Elster, 2017)

For sound that changes over time, however, this frequency approximation will not be representative. A Short Time Fourier Transform (STFT) is instead when the signal is divided into short time sequences, called windows, and the Fourier Transform is performed on them, under the approximation that the frequency is not changed inside the window. A naïve approach to the window is the rectangular window function, which is a vector of 1's for a sequence of values, and 0 for all others. This is then multiplied with the signal to create a windowed signal. This creates clear cuts at the end of the window. One problem with this approach is that the Fourier Transform consists of continuous cosine curves, creating ripple frequency artifacts at the end of the window as the function may be discontinuous there, see figure 8. For this reason, there are more sophisticated window functions, such as triangular or bell shaped, which give less ripples in frequencies. So besides representing the actual signal, window functions will introduce misrepresentations to the Fourier Transform. (Müller, 2015)

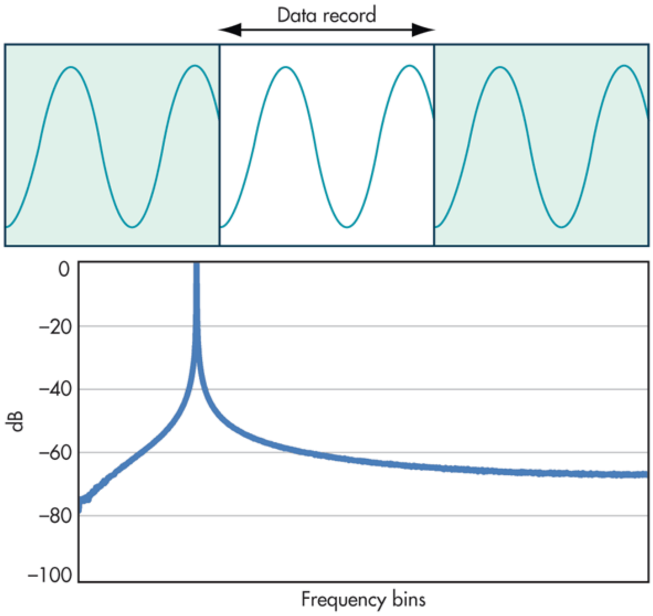

Figure 8 – Rectangle window function with frequency artifacts in the frequency domain (Müller, 2015)

There is a tradeoff between accuracy in time and in frequency. When using a longer window function, the artifacts at the end of the window will be less apparent, while information on frequency changes during the window is lost. A shorter window gives a better view of frequency changes, but creates more artifacts that will ripple through the frequencies (Bradford, 2007). For speech recognition, a typical window size is 20 to 40 ms, with 50% overlap between frames. A Hamming window is often used. The power spectral density is the power of the signal for each frequency, as a function of frequency (measured in Watts/Hz). This is (proportional to) the square of the amplitude of the signal.

5.3 Human imitating analysis

The Mel Frequency Cepstrum Coefficients (MFCC) is a procedure to map the frequency spectrum into a scale that better represents how humans perceive sound (Davis & Mermelstein, 1980). Out of all the frequencies we can perceive, humans are better at perceiving differences in lower frequencies than in higher frequencies. This is often done in preprocessing of speech. Speech recognition is a more common audio classification task than engine classification. One may ask if this step is necessary for engines, since the noise could be in frequencies where humans cannot detect them. If there are engines that make noises humans cannot detect, those instances would not have been classified as noise, but as good sounding engines. For this reason, it makes sense to investigate the sound humans hear. There could be potential in matching all kinds of errors with audio, to investigate whether a computer can hear other errors as well. That is, however, beyond the scope of this paper.

The Mel filter will sum up the powers over a number of frequency bins. For higher frequencies, more frequencies are included, see figure 9. The logarithm of this power is then taken to represent the human sensitivity to volume.

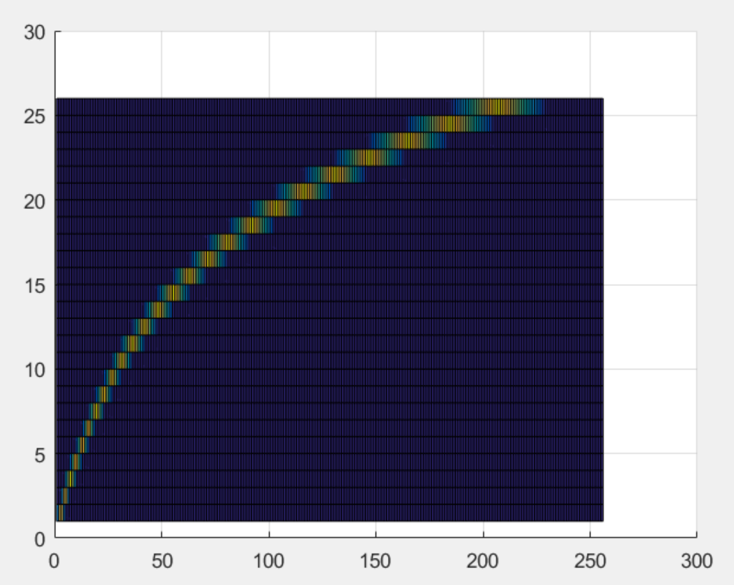

Figure 9 – 256 frequency bins are mapped into 26 Mel bins

For deep neural networks, this has been used as the only preprocessing successfully, sometimes referred to as fbank (Hinton, et al., 2012). To gain the MFCC, a DFT is further conducted on the fbank to identify features of speech. Finally, this is normalized, by subtracting the mean of each coefficient from each frame (Hinton, et al., 2012). This is also discussed by Tóth (2013).

5.4 Machine learning from audio

Machine learning from audio is a field which has been primarily focused on speech recognition. In speech recognition, the task is to interpret the sound of spoken words as words and sentences (Hinton, et al., 2012); (Abdel-Hamid, et al., 2014). Studies have also been done on bird whistles (Ross, 2006) and urban sound identification (Piczak, 2015), as well as music genre classification (Sigtia & Dixon, 2014).

Sound data is often preprocessed before an algorithm is trained on it. As described by Velik (2008), neural networks cannot learn to conduct Fourier transforms. For parallel computing purposes, however, a network with predetermined weights in its initial layers can be used to conduct a Fourier transform (Velik, 2008).

As an activation function, the rectifier function, $f(x) = max(0, x)$, is a good choice in deep neural networks for sound classification (Tóth, 2013). First, the backpropagation works better on rectifier functions than the popular sigmoid, because of the avoided gradient vanishing effect since the rectifier is linear (Tóth, 2013). Second, many weighs are set to 0 due to negative activity values, which increases the performance of the network as well (Maas, et al., 2013). Third, it is faster to compute since the exponential does not have to be calculated (Tóth, 2013).

Tóth (2013) also states that it has been popular to train neural networks with only one hidden layer, but recently it has shifted to using deeper networks. The demonstrator used 2000 neurons per hidden layer, and 1-5 hidden layers, on the TIMIT dataset of recorded sentences, a well known database for benchmarking in speech recognition. Furthermore, dropout reduces the level of overfitting in deep neural networks, in supervised learning tasks such as speech recognition (Srivastava, et al., 2014). They also use the TIMIT dataset for demonstration.

Besides DNN’s, CNN’s have also been successful at classifying sound. Hinton et al (2012) state that an advantage with CNN’s are their “temporal invariance”, that is, their ability to detect a certain sound regardless of when it appears in time. Furthermore, they argue that there is an advantage with some invariance in the frequency dimension as well, due to variation in a spoken voice.

Recurrent Neural Networks have also had success in speech recognition (Graves, et al., 2013). They are suitable since they have a memory, and can for that reason consider many previous inputs in time. In the case of audio, each frequency window is processed as a separate input, which makes them possible to handle input sequences of different length, something typically not possible for DNN’s or CNN’s. Furthermore, because of the depth in time, recurrent neural networks can learn immensely complex dynamic patterns, such as spoken words, changing tone 10’s of times per second (Sutskever, et al., 2013)

6. Current testing situation

During my participation at the Engine Test department at Scania, I received a good understanding of what they do and what problems they face. After the engines have been assembled, they are tested on how well they function. The engines arrive, one at the time, to a test cell where it is will be tested. Before its arrival, it has been prepared with a number of lids etc to make the testing go fast. The test bed operator receives the engine and locks it in position. Hoses and tubes are connected with all the necessities to run the engine. The operator leaves the test bed room and starts the engine from their control panel. An artificial load is put on the engine, simulating pulling the load of a truck. The power and torque are measured, together with temperature, rpm and other information. The engine goes through a program of different speeds, depending on which type of engine it is. All the measured values appear on the operator's screen in real time, and all measurements have tolerances. When a tolerance is overstepped, the number is highlighted in red for the operator to spot. The operator also listens to the engine sound through headphones, to detect anomalies. When the engine stops, the operator prints a deviation report for the measured parameters during the test cycle. The operator then turns the lights out in the test cell and searches for leakage from the engine using a flashlight. There is fluorescent material in all the liquids making them easy to spot in the dark. If there is anything wrong with the engine, a repairer is called upon to take the case forward. Most of the measurements are automated through sensors, and compared with tolerances automatically. The major tasks for the test bed operators are checking for leakage, listening to the engine for anomalies, handling of the engine and its setup, and supervising the test run in case something would go wrong.

Checking for leakage requires walking around the engine, and lighting up with a flashlight one part at the time. The input data here is visual, and the output is leakage or not, which is a classification problem. This could potentially be solved with machine learning, but would require a change of setup, including cameras, and different lighting. The problem is not trivial however, since the leakage can be small and may only be seen from a certain angle. Furthermore, some leakage which comes from the test bed equipment could be accepted, and so the operator or algorithm would have to be able to tell the difference. The algorithm potential is image recognition classification, such as a convolutional neural network.

Handling the engines and setting up the test bed for each engine is a manual task. It could hypothetically be automated, but requires lots of gripping and moving through space, and must be flexible enough to adapt when new engine types are introduced. Input is an engine and equipment at certain positions, and output is an engine setup for testing. This seems as if it is not suitable for machine learning. Industrial robots have been trained with reinforcement learning, so that they are rewarded when they act correctly. It seems that it is too critical to get this right every time, it might as well be done manually.

Supervising the test run means looking at the engine through a window, making sure all the hoses and tubes stay connected through the test run and looking at the control panel for out-of-tolerance measurements. It also includes looking out for the unexpected, which is why this seems difficult to do with machine learning. Unsupervised learning such as anomaly detection could be used on the visuals of the cell. One problem with this would be the vast amount of data that would be the input, in the form of real time video.

7. Interviews

Three interviews have been conducted, with somewhat different focus and scope. The first interview presented below is closely related to quality control and quality engineering, and did for that reason contribute more to the reflections of the potential of machine learning in quality control. The second interview widens the perspective of quality in an organization, and examines the applications in field quality. The third examines the market for machine learning services, as well as contribute with insights into broader applications in industry and sound analysis.

7.1 Interview with Johansson

I interviewed Krister Johansson at Scania, who works with error proofing in production. They faced many types of issues. Most of them consisted of detecting defects and originated from previous defects.

They had had problems with drilling a hole at a certain location of parts. To detect if the hole was not drilled, they had set up a vision system. The system was simple and it was programmed to detect holes in a visual interface. If a hole was not detected, the conveyor belt would be stopped and an operator would have to approve or remove the part. (Johansson, 2017) This type of problem could work with a machine learning algorithm as well, but does not seem practical. Detecting a hole is such an easy task that a probe or any kind of sensor should be able to do it.

When a defect arises, there are often many defective products, since the machines are configured to produce with low variation. The problem for the quality engineers is then to trace down how the defect arises, to correct it. This requires good knowledge of every step in the process (Johansson, 2017). This could be compared with a medical doctor, diagnosing a patient. This analogy is interesting because machine learning algorithms have been used to aid doctors making accurate medical diagnoses (Al-Shayea, 2011); (Li & Zhou, 2007). The input for this algorithm would be the end quality of the product, and the output would be what process is behaving incorrectly and causes the defect. The problem with this would be the large amounts of input parameters there may be, together with the small amount of historical data on defects, and large number of defects possible. An alternative would be to teach the algorithm which machines are in contact with which parts of the components. However, then it would not be machine learning, as the algorithm would not learn the possible defects from experience, but from a list handed to it. Johansson did sound interested in a system like this when I discussed it with him.

Perhaps diagnosis with machine learning can be done on a work station level. The more often quality is measured, the easier it is to connect input to output. If quality is measured right after a machining operation, this may well be subject to machine learning and diagnostics that the operator normally does. Just by going through what machine settings the operator changes, it may be possible to learn from this to apply either as recommendations or as automatic adjustments, as a control system.

Many inspection tasks are automated, especially those measuring dimensions. Some surfaces are inspected manually for surface defects. There were also camera systems for surface defect detection. One system was using machine learning, and was trained with pictures of good and bad components. The software was made by an external firm, while the integration with the product line and operator was done by Scania. Since surface defects are difficult to measure, it has traditionally been done manually. Programmed vision systems can be used as well, but are not as flexible for different kinds of defects as systems trained with machine learning. The time to train the network did not seem to take more than a few hundred images, while the installation with PLC and interface for the operator took the main part of the implementation time and effort. The competence required was not in machine learning at all, that is essentially what the external firm is providing, but in operations. The system requires a server with a high-end Graphics Processing Unit (GPU) for consumers, to process the images in real time. It also requires standard industrial cameras with resolution good enough do detect small defects. Although the software license was somewhat expensive, and the server required some setup, that can be used for more than one inspection. For that reason, Johansson sees great opportunities with this system for future inspection needs. (Johansson, 2017)

Another of the problems they faced was detecting gaps between two components after assembly. This was done manually for a time before a camera solution was tested. In this camera solution, many cameras took photos of the assembled components. The images were then processed in external software to produce an estimate of the gap between the components. There was a tolerance which this measurement was compared to, to determine whether the assembly passed (Johansson, 2017). This is a special kind of problem where dimensions are determined by photographs, which often are considered inexact. It required weeks of configuration with overlapping manual checks to make sure the camera solution was good enough, and that it was trustworthy. I was under the impression that the system did not use machine learning, but rather was trained through a vision interface. This problem could probably be approached with machine learning, using image recognition software for regression. However, this seems unnecessarily complicated considering that the problem is linear. It would require time to build the system and knowledge of machine learning algorithms, as well as computational resources such as a server. For the same problems, I believe simpler sensors could be advantageous, such as photoelectric or laser sensors.

In a modular product range like Scania's, variations of parts can come together to form multiple subassemblies and final products. A problem that may arise is the assembly of incorrect parts, creating an undesired subassembly. To avoid this, sensors can control what part is being used. Scania uses bar codes on the parts with information on what model it is, and a bar code scanner to detect it (Johansson, 2017). This is a simple solution which does not set out to detect features to dimensionally different parts. This requires that each part has the correct bar code on them. An alternative would be to use multiple sensors to detect features which are different for different part models. This may be complicated and require lots of sensors, as well as being inflexible for new part introductions. Image recognition software could be of use for this purpose. Training a machine learning algorithm to classify different parts should not be too difficult, since the differences between different models are much greater than the variation within, assumed all model differences are visible from the outside. However, a system like this would require computational resources and training time. The simple system of bar codes may well be just as effective as long as they are correct and readable.

When having problems with the process, understanding what is happening is essential to finding the problem. As some machines are operated with closed doors, it may be difficult to get a good understanding for what is going wrong. By inserting cameras inside the machines, and displays on the outside, the operator can watch what is happening and notice if the machine is behaving out of the ordinary (Johansson, 2017). This is a simple yet effective solution that allows control over the process by the operator. An obvious alternative is to have glass doors on the machine. However, cameras can give a better view, since it is possible to position them in multiple angles. Machine learning in this situation would be about detecting what is not going right. Vision systems would require video, since machining is done over time. This is computationally very heavy and it would be difficult to detect anomalies, considering the variations in appearance due to cutting fluids. Many machines have sensors for their own control system. This could perhaps be relevant input for machine learning, as discussed above. For gaining initial control, however, machine learning does not seem to contribute.

I asked Johansson about reliability for new systems, including their machine learning system for surface defects. To make sure they are reliable, all new systems are run in parallel with previous systems, such as manual quality control. The most important aspect of a new system for Scania is that is does not approve any defective parts. The whole configuration is then between warning about too many defective parts, and just the right ones. If the system warns too often, instead, the operators feel that they cannot trust it. For that reason, the decision to start using a new system is made by the technician when they consider the system reliable (Johansson, 2017). However, machines could be considered more reliable than people. Some defects, such as surface defects, are hard to measure and compare, and it is not easy to have a guideline for what to pass and what to scrap. For this reason, it is often up to the operator, and different operators may take different decisions on borderline cases. Furthermore, human errors occur in inspection, and can be affected by mood and focus. In the quality gates, work stations in the production line where inspections are made, Scania has focused on improving ergonomics and reducing distractions in attempts to improve quality inspection (Johansson, 2017).

7.2 Interview with Snellman

I also interviewed Isolde Snellman, who works with Quality Information at Scania. She develops tools for statistics about field quality, and makes customized quality reports on demand. The internal customers are mostly field quality engineers, following up on quality issues the end users have experienced.

The problems the field quality engineers are facing include detection of deviation in quality, deciding whether to start a quality case, tracking down the root cause, develop a solution, and deciding on market action. (Snellman, 2017)

Detecting deviations may seem like something suitable for machine learning. However, what to detect here is the increase in the ratio of faulty components per all of those components. Since it seems fairly clear what to look for, there can hardly be a need for machine learning.

Machine learning can potentially be useful for root cause analysis. Association rule learning is a method for finding relationships in large datasets. If a breakdown in one specific component more often occurs in trucks that come with another component, this can be learnt as a rule and displayed (Snellman, 2017). It would then be up to the engineer to examine the causality of the relationship. The data is readily available for the Quality Information team. Developing this kind of system also requires high competence in machine learning, which is available as well. It will take time to develop, tune, and implement it. Since there are systems for the same purpose today, the value of the new system would only be the additional insights. When I asked Snellman about transitions between systems, she though running both old and new systems in parallel for a while would let the users test both on the same data, and discover for themselves the advantages of the new system. One idea was to introduce the system to some users and train them in it, which later would show their colleagues (Snellman, 2017).

Besides regular quality improvements, Snellman sees great possibilities with machine learning, including for forecasting, report classification, and even for truck driving feedback.

7.3 Interview with Strömbäck

The final interview was conducted with Henrik Strömbäck, IT architect at IBM. He told me about a few projects they have done in predictive maintenance. By collecting data about breakdowns, noise and vibrations, they had built systems that could predict breakdowns, which then could be prevented by maintenance.

Many of their customers are not sure how this new technology can be used and what kind of results they can expect. IBM and their competitors contribute with knowledge in the field, as well as experience from similar industrial applications they have done before (Strömbäck, 2017).

There seems to be two main types of products, the first being standard applications such as image classification and speech recognition, which are offered ready to use as a cloud service. The other type is custom made products, which requires designing a machine learning algorithm adapted for that specific task (Strömbäck, 2017).

Strömbäck’s view on reliability is that customers realize that custom made machine learning implementations are experimental initially, and that it may take some adjustments before a system is ready to be used by a large number of users on a daily basis.

8. Development of acoustic classifier

8.1 Iteration and resource observations

According to Kotsiantis (2007), the flowchart (figure 10) describes the steps and orders in making a machine learning classification algorithm work. As I set out to do this myself, the resources needed for each step have been considered.

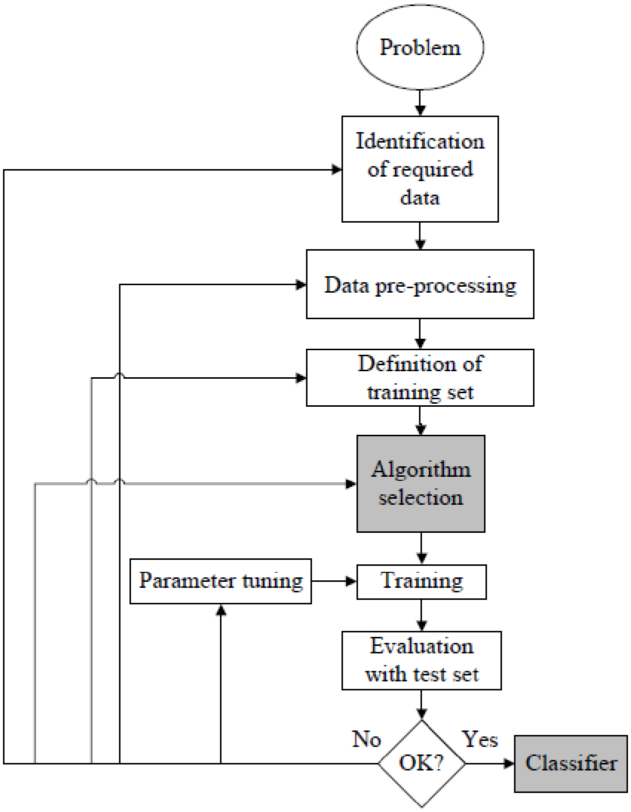

Figure 10 – Flowchart of classification construction procedure (Kotsiantis, 2007)

The process of developing a supervised machine learning algorithm is not as straight forward as it may seem. It is an iterative process, not only in the sense of numerical computations, but in the design phase as well. The further down you get in the flowchart, the more times you may have to iterate to get it right. There are guidelines and practices on how to preprocess data and tips for network architectures, but there are no right answers. The resource requirements were studies throughout the project including:

- Time – to do/wait
- Time – to calculate
- Knowledge – Machine Learning, subject

- Cost
- Computational resources
- Storage of data

8.2 Identifying of the problem

Identification of the problem may seem like a trivial part, but formulating a problem which can be solved by a machine is no obvious task for someone unfamiliar with machine learning. One must consider what the available data is, such as if there is input data with matching output data or not. Then one needs to consider the output, if it is a classifier, a regression, or a decision to interact. My project was to automate detection of bad sounding engines. The problem is to separate bad sound from good sound; a classification problem with two classes. There was previous data to train the algorithm on, and so I could use supervised learning. An initial idea was to have the algorithm separate different types of engine errors from each other, and so even this early step had to be revisited in the process. The identification of the problem is not too time consuming, as it is often fairly clear when a project is initialized.

8.3 Identifying required data

Identification of required data includes figuring out what data is needed, finding what data is collected and what needs to be collected, collect data, finding who owns the data, and being granted access to that data. In my case, there was data collected for a time period, but to gain more data, I had to wait for more engines to be tested. To gain access to the data, I had to wait for the owner to grant me access, as well as talk to a number of employees to figure out in which internal systems the data I was interested in was stored. Generally, collecting data can be very time consuming, depending on the production rate and amount of data required. The amount of data needed depends on how complex the data is. Identifying which data is required does not require too much knowledge of machine learning, although it does require knowledge of the task and of the organization, which someone working with the problem typically has. Depending on the amount of data, storage needs to be considered as well.

This step was revisited multiple times in the process, when trying out smaller datasets of less internal variation in the classes to easier distinguish them.

8.4 Data preprocessing

Data preprocessing requires knowledge and computational resources. After having collected all the data, I integrated different data sets to create one set of the data I needed. I then built a program that would cut the sound files to make the sounds comparable. This was done through visually identifying a frequency span where the difference in rpm was easily detected, through a spectrogram, see figure 11. The amplitude for each point in time in this frequency span was averaged, and the points in time where the amplitude average changed the most were used to identify the changes in rpm. This cutting reduced the data, making it easier to work with, but may have lead to that sound classified as bad was cut off from some files.

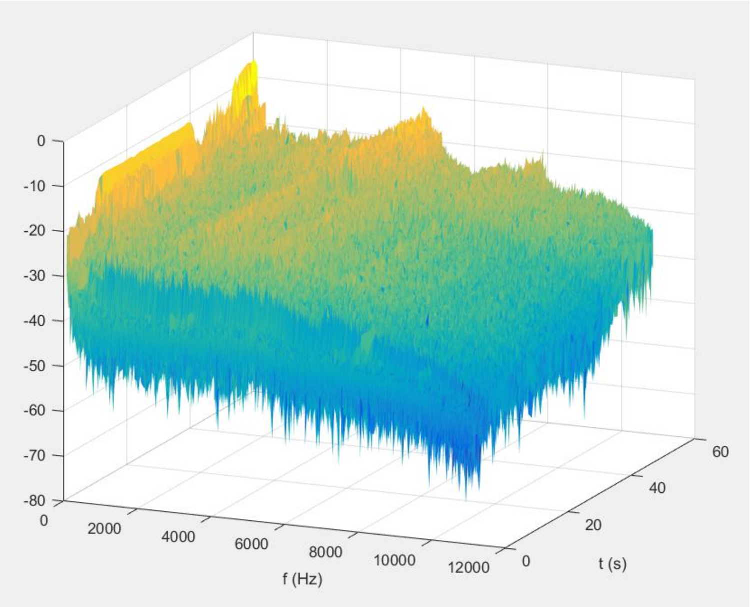

Figure 11 – Spectrogram of an engine sound

The sound data was then transformed to the frequency domain using a Short Time Fourier Transform, as described in the Theory section. Both window function and window length have been iterated a few times.

In a later iteration, the STFT was transformed to the logged Mel scale. The final Discrete Fourier Transform to receive the MFCC was not performed, as discussed in the theory section.

As a Deep Neural Network algorithm was selected later on the input for the algorithm had to be of equal length. For this reason, the STFTs where cut off which further reduced the data, but also risked that dynamic noise were cut off. The data was later normalized around 0 with a variance of 1, for each feature to make the gradient descent more effective.

When using a Convolutional Neural Network instead, I also had to revisit the preprocessing, and decided to pad the short instances with 0's instead of cutting the longer data short. That has the effect that all data was used, leaving bad sound at the end of the sound files detectable. I also avoided cutting the data short based on rpm and frequency, and so the log Mel frequency representations of the full sound recordings were used.

This stage requires knowledge about how the data needs to look for an algorithm to handle it, as well as specific knowledge of the kind of data. It also requires active time, and depending on the dimensions and size of the data, preprocessing may require some computational resources as well.

8.5 Definition of training set

The training set was chosen based on the small number of instances in the class of bad sounding engines. To not lose any of the data when training, cross validation was used, in such a way that algorithm was run multiple times, with change in which data was test and which was training data. To make sure each of the classes was represented proportionately in the test and training sets, a stratified k-fold cross validation was used, using the same distribution between the classes in both test and training set.

This is not too time consuming and there are common proportions to use. To make the findings general, the training set data should not be handpicked but randomly selected. Having 80% training and 20% testing is common, but using larger amounts of data may make it possible to include as much as 99% of the data in the training set (Raschka, 2015). It only requires some understanding about the tradeoff between training and testing.

8.6 Algorithm selection

Selecting the algorithm is a major part of solving the problem. There are often multiple alternatives that can solve similar problems, and there is often no obvious best alternative. To be able to make a good decision, one should have knowledge of different algorithms, and it may take time to be well informed of the advantages and disadvantages of the different alternatives. For sound classification, some well used algorithms are

- Deep Neural Networks
- Recurrent Neural Networks
- Convolutional Neural Networks

I have first used a Deep Neural Network (DNN) and in a later iteration a Convolutional Neural Network (CNN). See the theory section for a discussion on how they work and what they are good for. The DNN was a first choice because it has showed successful results on speech recognition, while being fairly straight forward to construct. The CNN being a special case of DNN, was later chosen because of its success in image recognition. Perhaps counterintuitive, image data is fairly similar to sound data, considering that the sound after processing is represented in a spectrogram or similar, see Figure 12 of log Mel sound.

Recurrent Neural Networks are good at detecting complex dynamic relationships in time series data, and have been successfully used on the TIMIT dataset (Sutskever, et al., 2013). However, since engine data is not especially dynamic, and since RNN's are considered more difficult to train than other networks (Sutskever, et al., 2013), I chose to focus on the DNN and CNN.

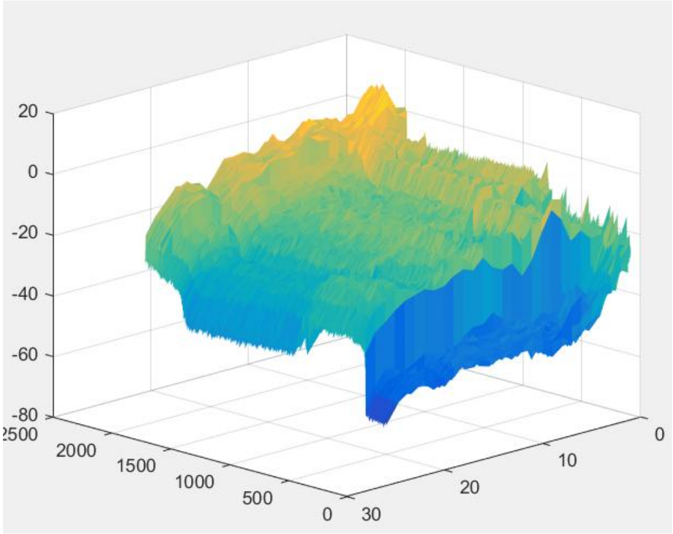

Figure 12 – Spectrogram of an engine sound on the log mel scale

8.7 Training the algorithm

The training phase includes running the algorithm in training mode, feeding it the training data with classified samples. This especially requires time, as it is done so many times. It also requires computational resources. There is a tradeoff between computational power and time to run the training. For large networks, the training can be done on multiple servers at the same time (Zaharia, 2016), and at a GPU to increase the computation speed. Some companies offer servers for short term renting, specifically for large computations, called cloud computing. Since many servers then can be used in parallel, the time can be greatly reduced. The software may also affect how long it takes to train the algorithm. I used Scikit-learn library in Python 3 for DNN, and Keras on backend Theano for CNN, both run on CPU on a PC. Each run took between 1 and 5 hours, and was iterated many times.

8.8 Evaluation with test set

The evaluation of the training is done with the test set. It is fast and easy compared to the training. Good performance measurements should be selected to determine how successful the algorithm is. I used a confusion matrix to be able to separate type I and type II errors. The result can be an indicator on how to proceed with tuning or whether to back further in the process.

8.9 Parameter tuning

Tuning of the parameters is done based on the performance of the test set, performance on the training set, as well as the loss function convergence. This is a systematic exploration since it is repeated many times. It takes some active time to make a plan for how to change parameters depending on result, and to revise that plan. The parameters will also affect training time. This part of the process requires good understanding for how the algorithm works and is the final step of making a successful machine learning program.

9. Results

9.1 Audio classification

From the DNN, the results were unsuccessful, meaning there was no accuracy on the testing set that reached more than slightly higher than a naïve guess. The higher accuracy the algorithm received on the training set, the lower it received on the testing set.

For the CNN, however, partially successful results have been achieved. For the whole dataset, containing a combination of engine models and different engine errors, no accurate results were reached. For dataset containing only one engine type and one type of error, the CNN was able to classify the test set successfully. The network that reached this accuracy was a convolutional neural network in thirteen layers, consisting of convolutional layers, pooling layers, and fully connected layers. Specifics on the network architecture can be found in Appendix I. Some results are showed in table 2 below.

Table 2 – selection of the many test evaluation results

Networks	Type I errors	Type II errors
CNN – type I focus	0.0%	12.5%
CNN – type II focus	20.0%	4.0%
DNN	13.0%	84.90%

9.2 Resource requirements

The resources that were needed for this specific design, is presented in Table 3 below. The values are cumulative, such that the time to do/wait/calculate is a sum of all iterations. As none of the steps required direct costs such as server renting, that resource column has been excluded.

Table 3 – resource requirements

	Time to do/wait	**Time to calculate**	**Knowledge**	**Computational power**	**Storage**
Problem	-	-	Basic ML	-	-
Identification of required data	A few weeks	-	Task & organization	-	100s of GB
Data pre-processing	A few weeks	Minutes	Data & ML	Personal Computer	100s of GB
Definition of training set	-	-	Some ML	-	-
Algorithm selection	Days	-	Good ML	-	-
Training	-	Weeks	-	Personal Computer	10s of GB
Evaluation with test set	-	Minutes	Some ML	-	-
Parameter tuning	Hours	-	Good ML	-	-

In the following section, the tradeoffs between the different types of resources will be discussed.

9.3 Quality problems and potential solutions

In table 4 below are the results of my analysis based on the interviews and literature review, for what machine learning can be used for when it comes to quality. As stated above, I have not studied these as in-depth as engine sound, and for that reason cannot determine their actual feasibility. This is, however, an indicator of what could be studied and or tested further.

Table 4 – quality problems and potential machine learning solutions

Quality Problem	**Machine Learning**	**Feasibility**
Audio anomalies	Classification (CNN)	Feasible
Checking for leakage	Image classification	Potential
Setting up engines	Reinforcement learning	Unnecessarily complicated
Supervising testing	Image classification	Potential
Feature detection	Image classification	Unnecessarily complicated
Cause diagnosis	Multidimensional Regression, association rule learning	Potential
Machine adjustments	Multidimensional Regression	Potential
Surface defects	Image classification	Feasible
Assembly gaps	Image Regression	Unnecessarily complicated
Part identification	Image classification	Unnecessarily complicated
Machine error cause	Image classification	Unnecessarily complicated
Field quality detection	Association rule learning	Unnecessarily complicated
Field quality cause analysis	Association rule learning	Feasible
Predictive maintenance	Classification	Feasible
Control chart pattern detection	Classification	Feasible
Optimizing production parameters	Classification	Potential

For the discussion that led to this, see sections Theory, Current testing situation, and Interviews above.

10. Analysis & discussion

10.1 Audio classifier

Engine sound is complex. In contrast to music and voices, engines are not adapted to make sounds humans can easily detect and distinguish; the sound is merely a byproduct. This is both the reason for this project, and the reason for why the problem is so difficult to solve. The results of the CNN were relatively successful, considering that the algorithm could tell apart all bad engines from most good engines, of one type. This means CNN's are one alternative of how to solve this problem. To make the engine sound classifier more accurate, there are requirements which are not met today. The main issue is insufficient training data. Scania makes large quantities of engines, but very few of them sound bad when tested. This means there is plenty of data in total, but very little concerning bad sounding engines. Furthermore, Scania produces many engine models, which naturally sound differently. Each engine model will also have different sounds for different kind of errors. As this may be difficult to collect, a realistic solution may be to go with other algorithms. One that may seem promising is anomaly detection. As discussed in the theory section, anomaly detection calculates distributions in feature space, to determine whether or not a sample belongs to the population of good samples. The advantage of this would be that it may work with only sound samples of good engines, which is the vast majority. An apparent disadvantage is the inability to separate different kinds of bad engines from each other. However, this is clearly an aspect of lower priority. Furthermore, classifiers rely on categorized data to train on, which requires that the data is correctly categorized. Considering that it is difficult to tell engine sounds apart, there is a high risk that some engines in the training set are incorrectly categorized. This could naturally lead to an algorithm which cannot separate good and bad sounding engines. By using unsupervised techniques, such as anomaly detection, this problem can be avoided.

10.2 Resource model synthesis

The idea here is to create a model for what kind of scenario leads to what kind of resource requirements. There are a few well known tradeoffs in computing, of computational time, memory, and computational resources. Computational resources and memory can both be connected to money, as they can be bought or rented through cloud computing services. The different inputs for this kind of model would be the situation, which can be broken down into

- Solution requirements
    - What requirements there are on accuracy etc
- Data
    - How large the data is
    - How complex the data is in terms of how difficult it is to understand and preprocess, including number of dimensions
- Organization
    - How rigid the organization in terms of change and data availability

Besides requirements for the algorithm, different organizations may have different resources available already, such as in-house competence or computational resources.

Accuracy requirements will affect the algorithm selection, the training and the parameter tuning. A high accuracy requires careful selection of the algorithm, as the algorithm needs to suit the data well. This requires more knowledge of different machine learning algorithms, but also good understanding of the structure of the data. The training is mainly affecting the time, as more advanced algorithms generally require longer computational time. Parameter tuning would also require more time, and knowledge, as this is the final step to achieve a well functioning algorithm.

The data size will naturally affect computational times for data preprocessing, training and testing, as well as storage requirements. The complexity of the data will affect the knowledge and time required for data preprocessing, and the knowledge and time required to select a suiting algorithm. A complex dataset may also increase the effort to select a good training set.

The time it takes to identify the data needed, and being able to access that data can be different in different organizations, depending on how restrictive the IT systems are and how access procedures are designed. This also affects what software is available and allowed which may increase difficulties of preprocessing and training.

The parts of the situation each set requirement for the process steps according with above analysis. Further are these process steps dependent on resources in accordance with the result table. As I remove the process steps, and connect the situation directly to the resources needed, I find the relationship portrayed in the Figure 13 below.

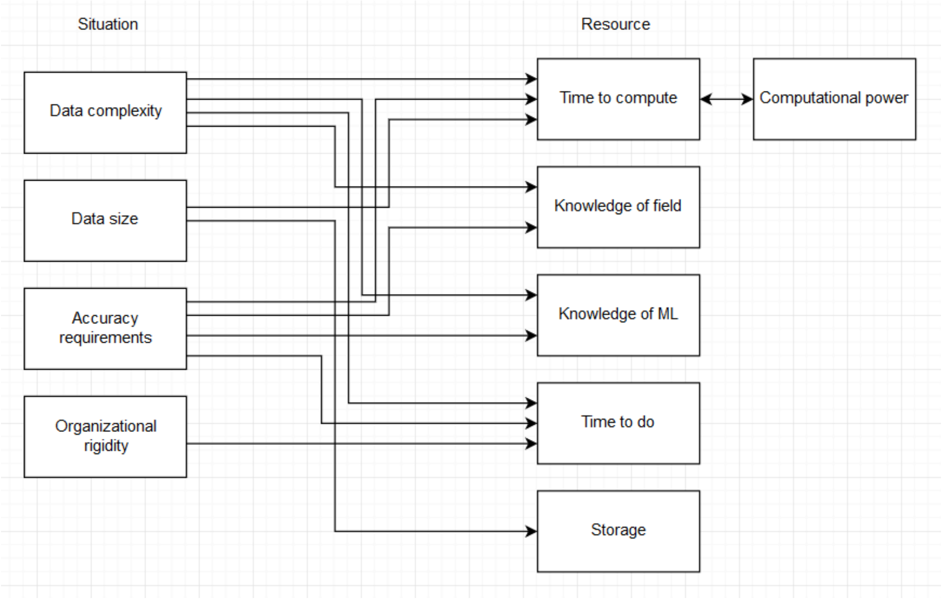

Figure 13 – graph portraying the relationships of situational elements and resources required for successful machine learning design

The idea is that this model should help the understanding of what different machine learning problems need in terms of resources to be solved successfully. By first identifying the situation, the required resources could be understood. The second step would be to compare the available resources with required, and consider acquiring additional resources. It would be useful with this relationship described as some kind of function, for example where a certain measurement of accuracy would require a certain level of skill etc. That is however not possible to determine, and this model does not set out to do so. This should rather be considered an indicative guide. It is important to note that this is based on one demonstrator project and would benefit from being verified by further studies. One key takeaway, regardless of the exact connections, is that if a company increases all of their resources dedicated to machine learning projects, it will become better at solving any machine learning problem.

10.3 Matching problems with machine learning

The problems discussed are by no means exhaustive; there are certainly more quality related problems suitable for machine learning. The feasibility is based on the interviews and my understanding of the problems and machine learning. The point here is not to decide whether or not these can be solved with machine learning, but to come with suggestions on future investigations, and to show that there may be plenty more applications, even in this narrow field of machine learning in quality control.

10.4 Reliability

Reliability, in terms of quality, refers to how rarely a product fails (Montgomery, 2013). The reliability of a quality detection system could similarly be considered reliable if it rarely fails to detect quality. This can be measured in hard numbers such as accuracy or type I and type II errors. However, reliability can also be considered whether the system is trusted. One issue to consider is how to make the people working with the system trust that it does what it is supposed to. Isolde Snellman experienced that users were overconfident in the results, when the system provided results in new applications (Snellman, 2017). This causes a risk that the results are trusted to a higher extent that they should, leading to bad decision making. Krister Johansson also talked about trust, and argued for the importance of eliminating type I errors first. Then, the system should be configured to eliminate type II errors, to increase trust. False alarms were the main issue for distrust in his experience (Johansson, 2017). Naturally, type II errors are easier for operators to detect than type I errors, since type II may require an operator to double check the quality manually for those classed as bad quality. Discovering type I errors, instead, requires a manual check of all good parts, which is a much more comprehensive task. To properly ensure high accuracy and trust, initially running systems in parallel may be a good solution, which will make it possible to study both type I and type II errors and how they decrease to a satisfying level. By first focusing on eliminating type I errors, and then focusing on type II errors, there is a risk of actually increasing type I errors. If type I errors are first eliminated, the class border is fairly close to the good samples in feature space. If later, only type II errors are detected and subject to training, this could move the class border away from the good samples, increasing type I errors.

One clear advantage of a human operator is their ability to act on the unexpected. An algorithm can act according to how it is taught, but is bad at understanding new situations. An example would be if an audio detection system was introduced that can determine what is wrong with the engine, and then it encounters a new error, or a new type of engine, or a broken microphone. How it would act in this situation is difficult to determine, but it would probably not be able to handle the situation satisfactory. A natural solution would have some kind of other-class for anomalies, which would signal an operator to help.

10.5 Readiness of machine learning in industry

As new technologies develop, they may be more or less ready to be applied on a wider scale. This is the case for machine learning as well. Although it has conceptually existed for half a century, it is in recent years machine learning has really taken off, partly due to the advances in hardware. Whether it is ready to be used depends on the requirements for accuracy, and how common the problem at hand is. The Imagenet Large Scale Visual Recognition Challenge is an annual challenge which serves as a benchmark in visual classification, and is a good indicator for how far the technology has reached. Deep neural networks won the first time in 2012 and have been the winners since. In 2015, the algorithms surpassed human accuracy, see figure 14.

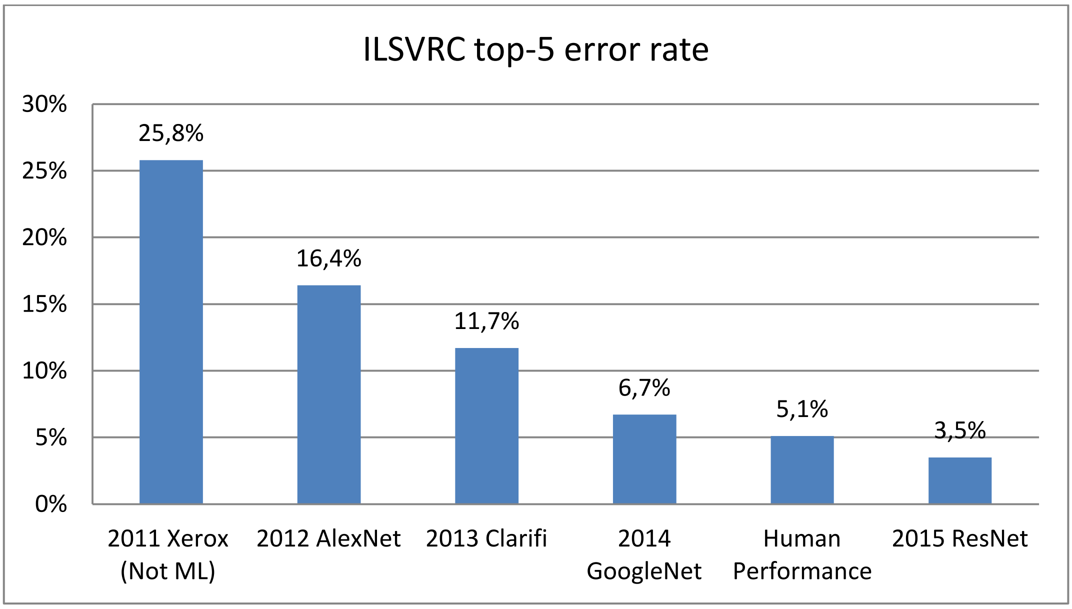

Figure 14 – Error reduction over time due to development in Machine Learning (Stanford Vision Lab, 2017)

Speech recognition is one of the main applications for machine learning on sound, and is therefore one of the most studied. Although speech recognition consists of many steps, including making sentences, one on the first is to recognize the sounds and classify them. As mentioned earlier, the TIMIT dataset is well known for benchmarking. The error rate has decreased over time:

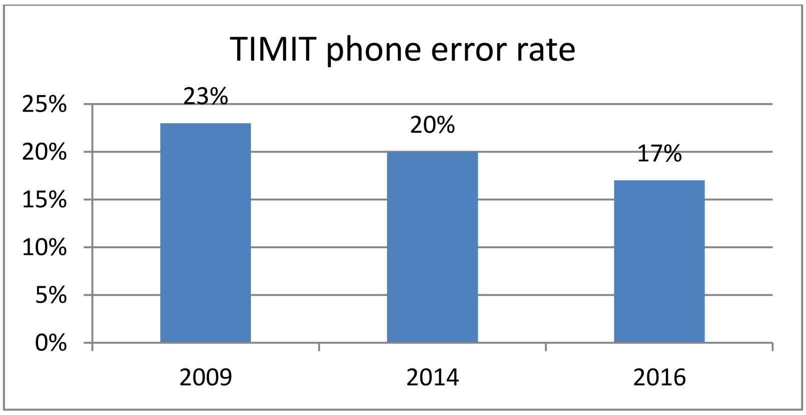

Figure 15 – TIMIT phone error rate development over time (Abdel-rahman, et al., 2009) (Abdel-Hamid, et al., 2014) (Lu, et al., 2016)

The slow increase indicates that classifying sound is difficult, and that there is a long way to go before perfection. In speech recognition, sounds are combined to form words, and words to form sentences. This makes it possible to guess on similar sounds or similar sounding words depending on the other content in the word or sentence. As a high functionality has been reached, the incentives to improve speech recognition further may have decreased.

Competence in machine learning is sought after in the modern job market, and many competent students end up in large software companies. Some of these companies offer their knowledge as a service to others as well, such as industrial companies. The range of services spans from parallel computing servers, to software licenses, or even pretrained algorithms ready to use in the cloud. For standard tasks like image recognition or speech recognition, these solutions may be excellent approaches, effectively eliminating the need for machine learning competence in-house. For more specific problems, however, these are not enough, as a solution will have to be customized.

10.6 Testing setup

The current test cells have a microphone on the wall. It samples in 22050 samples per second, in stereo sound. The cells are small rooms with closed doors, and from in there you can hear an engine being tested in the next cell. The walls are hard and covered with equipment used for testing engines. To improve the ability to detect anomalous sounds, for both operators and computers, an improvement in this environment could be a good step.

First of all, I recommend sampling in 44100 Hz. A sampling rate is twice the maximum frequency possible to reproduce with frequency analysis. This means 22 kHz sampling only yields about 11 kHz as a maximum frequency to detect. Instead, by using the 44.1 kHz standard sampling rate, the analysis will be able to detect up to 22 kHz, which includes the full hearable spectrum up to 20 kHz.

The bit depth, that is, the resolution of each sample, is 16-bit as of today. This is common in CD quality music recordings. Higher bit depth than that does not sound better, however it can matter when the audio signal is being processed. For that reason, most audio professionals use 24-bit (Connor, 2008). In this application, the audio is being processed in multiple steps, and so I would recommend considering 24-bit of bit depth.

Stereo could be useful if it was recorded from different microphones at different locations in the cell. As of now, the audio signal is very similar, due to the distance between the engine and the microphone, and due to echoes in the test cell. For that reason, half the data is not useful. I would recommend recording in mono for data size reduction.

As sound volume decreases by the square of the distance to the source, having the microphone closer to the engine could yield better recordings, as more direct engine sound would be recorded, and less reflected from the walls. Please note that it is important that the microphone is in the same position, and probably should not be moved back and forth for each engine test. Another way to reduce the reflection from the walls is to introduce sound absorbent materials where possible. A problem with this is that the cells are subject to a rough environment with engine fluids leaking, and already covered with equipment. Perhaps absorbent ceiling panels could be something to further investigate.

Absorbent materials will decrease echoing, and make the direct engine audio signal clearer; however it will not reduce noise from adjacent test cells. Insulating walls, particularly with heavy materials, would reduce this noise. Since this seems to be a rather comprehensive measure, I would suggest starting out with others. Furthermore, when engine testing is done at a lower capacity, one step towards improvement would be to simply use every other test cell, leaving an empty in between. Another approach would be scheduling the test cells so that every other cell set up their engine while the others run the tests. However, the sound from adjacent cells is not a major contributor to noise, in my view.

Good equipment can make a difference as well, in both ends. Microphones can have a huge difference in quality. Some may be adapted for speech or music, but the microphones that should be used in this setting are those which represent sound in the most natural way. There are dust and water resistant microphones for industrial environments available. Likewise, the headphones used by the operators can affect their ability to detect noise. Besides the ability to represent sound while new, speakers lose their ability over time, especially if they are used extensively. Without knowing the quality of the microphones or headphones used at Scania, I would recommend trying high-end products to evaluate the difference.

Storage of audio data is one of the major limitations to training a machine learning algorithm, as discussed. For that reason, I recommend storing more data, but also storing it differently. Today the data reflects what is produced, that is, more of some engine types, and very few with sound anomalies. To better be able to train a machine learning algorithm, sound anomalies must be prioritized. I recommend saving all recordings of anomalous sounds going forward, as well as equal amounts of data from different engine types. I do recommend continuing saving the files in WAV format, as other formats are compressed with loss of information. To reduce the overall space requirements, I would suggest a lossless compression. Besides increasing the potential for machine learning in the future, saving anomalous sound makes a good starting point for training operators at detecting anomalies.

10.7 Ways to integrate with testing

If a machine learning algorithm could be 100% accurate on detecting sound anomalies, the operator would not have to listen to the engines run. Their presence is still required since they are setting up the engines for testing, and check for leakage afterwards. However, operators could potentially service multiple test cells. This is not the case for a near future. Before this will be feasible, more anomalous sounds must be recorded, or other algorithms explored, as discussed above. A first step to integrate this technology with the testing could be through the preprocessing. Instead of listening to the engine sounds, the operators could potentially be analyzing spectrograms. If this makes it easier or harder to detect than listening to sound is hard to say, but it will require training for the operators. Advantages of using this is that two spectrograms can be compared side by side, they can be discussed and analyzed with all information available at the same time, where you do not have to take time to listen again. A potential advantage if this would work out well is that it may be faster than listening. The effect of this is as discussed above, that operators may be able to take care of multiple test cells. Just like the actual sounds, the spectrograms will be easier to interpret and distinguish with suggested setup improvements.

10.8 Quality and cost considerations

The demonstrator would not improve quality control due to its limited scope of engine types, and accuracy issues. However, further developed versions may improve quality control. Improving quality control can lead to better quality, and lower costs. Quality from the production process does not get better directly by detection, however, it increases the ability to control the production processes. Furthermore, quality control detects bad products and prevents them from reaching the customers, which increases quality of products sold. Increasing quality comes with costs, of four main types: prevention costs, appraisal costs, internal failure costs, and external failure costs (Montgomery, 2013). Prevention costs include quality engineering, process design and process control, whereas appraisal costs include product inspection. There is a natural tradeoff between costs of quality, since spending more on prevention and appraisal reduces costs of failure. Where the balance between these is, depends on the production cost of the product as well as on the company's quality aspiration. For example, a manufacturer of plastic toys may not invest heavily in prevention or appraisal, since production costs are so low that a high internal scrap rate is acceptable, and external failures rarely lead to warranty claims. In contrast, a truck manufacturer relies on prevention and appraisal because of the high production costs, leading to expensive scrapping and repairing, and high requirements from customers followed by expensive warranty claims.

Automation is often considered for cost reduction in high output manufacturing, and cost is one consideration in this case as well. Using machine learning algorithms to inspect audio quality could lead to a reduction in personnel requirements, as discussed above. However, I argue a reduction in inspection costs is secondary to the accuracy of the inspection. In line with the explanation above, reducing appraisal costs may lead to an increase in other costs, as a consequence of lower quality. For that reason, inspection accuracy should be of focus when determining whether to implement machine learning, even if it would increase the cost of inspection. Reducing costs while improving accuracy would naturally be the most beneficial, which may well be possible due to this new technology.

10.9 Recommended future research

Other types of preprocessing could be examined in relation to engine sounds. Bradford (2007) suggests multiple methods for static and dynamic sounds and vibrations, and so does Abdel-Hamid et al (2014). Other types of machine learning algorithms should also be examined, such as RNN or Anomaly Detection with statistical distributions. Furthermore, the resource requirements relationship model needs further examination in more machine learning projects to verify, specify, or change it. One of the previously mentioned delimitations was hearable sound. It may be possible for machine learning algorithms to pick up sound details which humans cannot hear, which would have to be classified with quality data other than sound. Field quality of early engine breakdowns could potentially be used for this classification. In relation to this could preprocessing be examined which does not imitate human hearing. Moreover, production parameters could be optimized using machine learning and quality, to model the production processes for improvement purposes. Multiple quality problems feasible to solve with machine learning have been suggested in the results section, which requires further studies.

Reference List

Abdel-Hamid, O. et al., 2014. Convolutional Neural Networks for Speech Recognition. *ACM Transactions on Audio, Speech, and Language Processing.*

Abdel-rahman, M., Dahl, G. & Hinton, G., 2009. *Deep Belief Networks for phone recognition.* Whistler, Canada, NIPS Workshop on Deep Learning for Speech Recognition and Related Applications.

Al-Shayea, Q. K., 2011. Artificial Neural Networks in Medical Diagnosis. *International Journal of Computer Science Issues,* 8(2).

Bosch, 2016. *Kaggle Bosch Production Line Performance.* [Online]
Available at: https://www.kaggle.com/c/bosch-production-line-performance
[Accessed 05 06 2017].

Bradford, S. C., 2007. *Time-frequency analysis of systems with changing dynamic properties,* s.l.: California Institute of Technology.

Britz, D., 2015. *WILDML - Recurrent Neural Networks.* [Online]
Available at: http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/
[Accessed 20 07 2017].

Connor, D., 2008. *The Stereo Bus.* [Online]
Available at: http://thestereobus.com/2008/01/12/sample-rate-and-bitrate-the-guts-of-digital-audio/
[Accessed 05 05 2017].

Cortes, C. et al., 2017. *AdaNet: Adaptive Structural Learning of Artificial Neural Networks.* Sydney, International Conference on Machine Learning.

Davis, S. B. & Mermelstein, P., 1980. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *Transactions of Acoustics, Speech, and Signal Processing.*

Diko, M. D. et al., 2017. Phase II control charts for monitoring dispersion when parameters are estimated. *Quality Engineering.*

Dunning, T. & Friedman, E., 2014. *Practical Machine Learning: A New Look at Anomaly Detection.* s.l.:O'Reilly Media, Inc.

Elster, A. D., 2017. *MRI Questions.* [Online]
Available at: http://mriquestions.com/fourier-transform-ft.html
[Accessed 26 05 2017].

Figueroa, R. L., Zeng-Treitler, Q., Kandula, S. & Ngo, L. H., 2012. Predicting sample size required for classification performance. *BMC Medical Informatics & Decision Making.*

Fountoulaki, A., Karacapilidis, N. & Manatakis, M., 2011. Augmenting statistical quality control with machine learning techniques: an overview. *Int. J. Business and Systems Research, Vol. 5, No. 6.*

Ganji, Z. A. & Gildeh, B. S., 2016. A class of process capability indices for asymmetric tolerances. *Quality Engineering.*

Gauri, S. K. & Chakraborty, S., 2007. A study on various features for effective control chart pattern recognition. *International Journal of Advanced Manufacturing Technology.*

Graves, A., Mohamed, A.-r. & Hinton, G., 2013. *Speech recognition with deep recurrent neural networks.* Vancouver, Canada, International Conference on Acoustics, Speech and Signal Processing.

Guh, R. S. & Shiue, Y. R., 2005. On-line identification of control chart patterns using self-organizing approaches. *International Journal of Production Research.*

Hinton, G. et al., 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine.*

Ho, E. S. & Chang, S. I., 1999. Integrated neural network approach for simultaneous monitoring of process mean and variance shifts - a comprehensive study. *International Journal of Production Research.*

Johansson, K., 2017. *Production Quality Issues* [Interview] (06 07 2017).

Ko, B. J., Ortiz, J., Salonidis, T. & Wood, D., 2016. *Acoustic Signal Processing for Anomaly Detection in Machine Room Environments.* s.l., ACM International Conference.

Kotsiantis, S. B., 2007. Supervised Machine Learning: A Review of Classification Techniques. *Informatica,* pp. 249-268.

Li, M. & Zhou, Z.-H., 2007. Improve Computer-Aided Diagnosis With Machine Learning Techniques Using Undiagnosed Samples. *Transactions on Systems, Man, and Cybernetics,* 37(6).

Low, C., Hus, C. M. & Yu, F. J., 2003. Analysis of variations in multivariate process using neural networks. *Journal of Advanced Manufacturing Technology.*

Lu, L. et al., 2016. *Segmental Recurrent Neural Networks for End-to-End Speech Recognition.* s.l., Interspeech 2016.

Maas, L. A., Hannun, Y. A. & Ng, Y. A., 2013. *Rectifier nonlinearities improve neural network acoustic models.* s.l., ICML.

Montgomery, D. C., 2013. *Introduction to Statistical Quality Control.* s.l.:Wiley.

Müller, M., 2015. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications.* s.l.:Springer International Publishing.

Ng, A., 2013. *Anomaly Detection vs. Supervised Learning,* Stanford: Stanford University.

Oakes, J., 2016. *Search Engine Land.* [Online]
Available at: http://searchengineland.com/experiment-trying-predict-google-rankings-253621
[Accessed 15 04 2017].

ODSC Team, 2016. *ODSC.* [Online]
Available at: https://opendatascience.com/blog/how-to-perform-multiple-regression-using-neural-networks-in-keras/
[Accessed 15 06 2017].

Olah, C., 2015. *Github - Understanding LSTM Networks.* [Online]
Available at: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
[Accessed 13 08 2017].

Pacella, M. & Semeraro, Q., 2007. Using recurrent neural networks to detect changes in autocorrelated processes for quality monitoring. *Computers and Industrial Engineering.*

Pandey, R., Naik, S. & Marfatia, R., 2013. Image Processing and Machine Learning for Automated Fruit Grading System: A Technical Review. *International Journal of Computer Applications.*

Piczak, K. J., 2015. *Environmental Sound Classification with Convolutional Neural Networks.* Boston, USA, International Workshop on Machine Learning for Signal Processing.

Raschka, S., 2015. *Python Machine Learning.* Birmingham, UK: Packt Publishing Ltd..

Raschka, S., 2017. *sebastianraschka.* [Online]
Available at: https://sebastianraschka.com/faq/docs/diff-perceptron-adaline-neuralnet.html
[Accessed 25 04 2017].

Ribeiro, B., 2005. Support vector machines for quality monitoring in a plastic injection molding process. *IEEE Transactions on Systems, Man, and Cybernetics, Part C.*

Ross, D. J., 2006. *Bird Call Recognition wih artificial neural networks, support vector machines, and kernel density estimation,* Winnipeg, Canada: University of Manitoba.

Sa, I. et al., 2016. DeepFruits: A Fruit Detection System Using Deep Neural Networks. *MDPI.*

Shao, Y. E. & Chiu, C., 1999. Developing identification techniques with the integrated use of SPC/EPC and neural networks. *Quality and Reliability Engineering International.*

Sigtia, S. & Dixon, S., 2014. *Improved music feature learning with deep neural networks.* Florence, Italy, International Conference on Acoustics, Speech and Signal Processing.

Sim, K. L. & Rogers, J. W., 2008. Implementing lean production systems: barriers to change. *Management Research News,* Volume 38.

Smith, A. E., 1994. X-bar and R control chart interpretention using neural computing. *International Journal of Production Research.*

Snellman, I., 2017. *Field Quality* [Interview] (26 07 2017).

Srivastava, N. et al., 2014. Dropout: A Simple Way to Prevent Neural Networks from. *Journal of Machine Learning Research.*

Stanford Vision Lab, 2017. *ImageNet Large Scale Visual Recognition Challenge.* [Online]
Available at: http://image-net.org/challenges/LSVRC/
[Accessed 2017].

Strong Analytics, 2016. *Strong Analytics.* [Online]
Available at: https://www.strong.io/blog/deep-neural-networks-go-to-the-movies
[Accessed 15 07 2017].

Strömbäck, H., 2017. *Machine Learning Consulting Interview* [Interview] (10 08 2017).

Sutskever, I., Martens, J., Dahl, G. & Hinton, G., 2013. *On the importance of initialization and momentum in deep learning.* s.l., International conference on machine learning.

Tellaeche, A. & Arana, R., 2013. *Machine learning algorithms for quality control in plastic molding industry.* Cagliary, Italy, Emerging Technologies & Factory Automation Conference.

Teoh, W. L. et al., 2016. 2016. *Quality and Reliability Engineering International.*

Tóth, L., 2013. *Phone recognition with deep sparse rectifier neural networks.* s.l., ICASSP: Acoustics, Speech and Signal Processing.

Wang, C. H., Guo, R. S., Chiang, M. H. & Wong, J. Y., 2008. Decision tree based control chart pattern recognition. *International Journal of Production Research.*

Velik, R., 2008. *Discrete Fourier Transform Computation Using Neural Networks.* Suzhou, China, International Confernce on Computational Interlligence and Security.

Weusten, J. J. A. M. & Tummers, P. H. G., 2016. 2016. *Quality Engineering.*

Xie, Y., Fu, Q. & Liu, Y., 2015. Imbalanced Data Sets Classification Based on SVM for Sand-Dust Storm Warning. *Discrete Dynamics in Nature and Society.*

Yeong, W. C., Lim, S. L., Khoo, M. B. C. & Castagliola, P., 2017. Monitoring the coefficient of variation using a variable parameters chart. *Quality Engineering.*

Zaharia, M., 2016. *Hosted Data Platforms*, Cambridge, USA: MIT Professional.