International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

Detection of interferences in an additive manufacturing process: an experimental study integrating methods of feature selection and machine learning

Darko Stanisavljevic, David Cemernek, Heimo Gursch, Günter Urak & Gernot Lechner

To cite this article: Darko Stanisavljevic, David Cemernek, Heimo Gursch, Günter Urak & Gernot Lechner (2019): Detection of interferences in an additive manufacturing process: an experimental study integrating methods of feature selection and machine learning, International Journal of Production Research, DOI: 10.1080/00207543.2019.1694719

To link to this article: https://doi.org/10.1080/00207543.2019.1694719

Published online: 27 Nov 2019.

International Journal of Production Research, 2019
https://doi.org/10.1080/00207543.2019.1694719

Detection of interferences in an additive manufacturing process: an experimental study integrating methods of feature selection and machine learning

Darko Stanisavljevic[a*], David Cemernek[b], Heimo Gursch[b], Günter Urak[b] and Gernot Lechner[a**]

[a]*Virtual Vehicle Research Center, Graz, Austria;* [b]*Know-Center GmbH, Graz, Austria*

(*Received 28 September 2018; accepted 10 November 2019*)

Additive manufacturing becomes a more and more important technology for production, mainly driven by the ability to realise extremely complex structures using multiple materials but without assembly or excessive waste. Nevertheless, like any high-precision technology additive manufacturing responds to interferences during the manufacturing process. These interferences – like vibrations – might lead to deviations in product quality, becoming manifest for instance in a reduced lifetime of a product or application issues. This study targets the issue of detecting such interferences during a manufacturing process in an exemplary experimental setup. Collection of data using current sensor technology directly on a 3D-printer enables a quantitative detection of interferences. The evaluation provides insights into the effectiveness of the realised application-oriented setup, the effort required for equipping a manufacturing system with sensors, and the effort for acquisition and processing the data. These insights are of practical utility for organisations dealing with additive manufacturing: the chosen approach for detecting interferences shows promising results, reaching interference detection rates of up to 100% depending on the applied data processing configuration.

Keywords: additive manufacturing; machine learning; 3D-printer; interference detection; data processing system; feature engineering

1. Introduction

Market requirements and technical innovations motivate transformations of manufacturing processes at a tearing pace. These transformations have an impact on several fields, including new manufacturing technologies, interconnectedness of manufacturing systems, or availability and access to extended manufacturing data. Improved sensor technology enables a massive collection of measurements, catering for a much more detailed and comprehensive monitoring of manufacturing equipment and machinery. Similar to the sensor technology, the available digital networks, computing power, and data storage get ever more powerful over time. Hence, it is becoming more and more common that manufacturing appliances are equipped with sensors and connected to computational entities, thus forming a fully integrated Cyber-Physical System (CPS). CPSs provide an increased availability of data collected on shop floors and offer new possibilities to master challenges in the manufacturing domain by applying methods with origin in Data Science. This is a big step towards the next generation of manufacturing, referred to as 'Industry 4.0' (Lee, Kao, and Yang. 2014; Lasi et al. 2014).

In addition to these digital changes also new manufacturing technologies arise. Besides progress in 'traditional' technologies based on computer numerical control (CNC) – like milling, turning, or grinding (as indicated, e.g. in Suh et al. 2008) – additive manufacturing technologies like 3D-printing or laser sintering made a great leap forward in their development (Calignano et al. 2017). A rising number of researchers give attention to the proclaimed 'additive revolution', advancing the technology from being applied merely in development, prototyping, or rapid tooling to a serious alternative to conventional manufacturing for more and more applications, e.g. for products with complex and personalised geometries, respectively (Battaïa et al. 2018; D'Aveni 2015; Gao et al. 2015; Jayaram and Vickery 2018).

One natural and frequently occurring issue of new manufacturing technologies is the sensitivity to interferences leading to challenges for manufacturing process control. Difficile manufacturing processes like additive manufacturing technologies are prone to interferences induced by dependent (internal) or independent (external) sources, as indicated for instance by Kim et al. (2018), Lu and Wong (2018) or Tofail et al. (2018). Such interferences might have a severe impact on various production-related topics, as for example the state of the unit itself or the resulting product quality (Franco-Gasca

*Corresponding author. Email: darko.stanisavljevic@v2c2.at
**Current address: Department of Production and Operations Management, University of Graz, Universitaetsstrasse 15/E3, 8010 Graz, Austria.

et al. 2006; Oropallo and Piegl 2016; Teti et al. 2010). To explore and overcome such issues and to pave the way for intelligent machining, machines are equipped with sensor-based measuring systems collecting data related to interferences which occur during the manufacturing process execution (Altintas 2012; Anderegg et al. 2019).

The sensors and the data they produce are the basis for an analysis, resulting in a deeper understanding of the additive manufacturing processes and technologies. Huang et al. (2015) describe that the research need in additive manufacturing not only lies in materials and processes, but also in sensing, control, and modelling to derive a better understanding of additive manufacturing processes. The work of Tofail et al. (2018) highlights that measuring and evaluation of product quality are still a challenge in science and industry, especially in-situ measuring and quality evaluations supporting the manufacturing process. In this context, increased knowledge about product quality and manufacturing conditions by improving detection of interferences supports the optimisation of additive manufacturing in order to better utilise feed materials, increase production yield, avoid rejects and rework, and improve energy efficiency.

This article presents an approach to detect such interferences during an additive manufacturing process with a 3D-printer. The 3D-printer is placed in a controlled laboratory setting ensuring reproducible experimental conditions. Vibrations as interferences are applied while the printing process is ongoing. The presented approach imitates a shop floor for additive manufacturing where both the build plate and the print head of a 3D-printer are equipped with three-dimensional acceleration sensors with a high sampling rate and resolution. The collected measurements contain periods with and without vibration interference, thereby creating the data corpus for subsequent data processing and machine learning (ML) analysis. In order to process this data in a structured way, a data processing system (DPS) integrating mechanisms of data pre-processing, feature extraction, feature selection, ML, and result processing is developed and implemented.

The work contributes to and extends the scientific literature by (1) introducing a setup suitable for reproducible interference experiments in additive manufacturing independent from real shop floor data; (2) studying ML-based detection of interferences which might occur during an additive manufacturing process and respective detection performance; (3) providing an overview of gained insights from implementing such a system, with a particular focus on the efforts to equip manufacturing systems with sensors and analysing the data. The experimental setup and the DPS provide the basis for the evaluation of 50,000 different interference detection configurations, confirming that ML can achieve excellent interference detection rates when combined with appropriate pre-processing and data transformation methods.

The next section discusses the existing literature and work related to this paper. Section 3 elaborates on the methodology, the experimental setup and data acquisition. This is followed by a description of the DPS and ML pipeline in Section 4. Section 5 then evaluates the different pipeline configurations and respective results and sheds light on the effectiveness of and the effort spent on the study. The paper closes with a discussion of the findings and an outlook on further research opportunities in Section 6.

2. Related work

The approach in this work is a prototypical application of Industry 4.0 using a 3D-printer, dealing with ‘the establishment of intelligent products and production processes’ (Cemernek, Gursch, and Kern 2017). It is embedded in the context of cyber-physical systems, integrating typical characteristics and technologies like computation, communication, physical production, additive manufacturing, or (big) data (Ghobakhloo 2019; Hahn 2019; Ivanov, Dolgui, and Sokolov 2019; Wang and Wang 2016; Winkelhaus and Grosse 2019).

Due to the focus on an experimental setup using additive manufacturing for data-based detection of interferences, we first discuss scientific literature related to applications and experiments for anomaly detection in manufacturing, mainly based on sensor data. Thereafter, an overview of current data-driven studies involving technologies of additive manufacturing is provided.

2.1. Sensor-based applications and experiments in manufacturing

Teti et al. (2010) provide an overview of monitoring of machining and discuss respective sensors and sensor systems (e.g. measurement of motor power, forces, acoustics, vibration), advanced signal processing, scopes of monitoring and methods for decision making and support. In a more specialised review, Ambhore et al. (2015) summarise the topic of tool condition monitoring, particularly process parameters measuring, data acquisition and processing, and decision-making techniques. The authors conclude that progress regarding new sensors and sensor systems, advanced sensor signal and data processing, and intelligent sensor monitoring is required.

A bulk of literature deals with the detection of machine and tool condition, primarily for Computerized Numerical Control (CNC) machinery. In an early study, Liu, Zhang, and Lin (1998) use typical fault categories of CNC machine tools as reference patterns to identify deviations from the desired output. Using a comparably small set of data, the presented fuzzy pattern recognition technique based on primary component extraction is able to identify major components of machine tool faults and also respective categories and direction. Downey et al. (2015) present a data acquisition approach using multiple sensors – acoustic emission, cutting forces, vibration, images for wear of tools – in a real-time production environment. The continuation of the work includes the demonstration, description and visualisation of the sensor-fused acquired data (Downey et al. 2016). Balsamo et al. (2016) could detect critical tool failures on a turning machine by means of statistical characteristics of cutting forces while the application of acoustic emission failed. Signals of a camera and a three-component dynamometer are fused and processed applying unsupervised batch training in order to monitor the tool condition of a CNC milling machine in Wang et al. (2007). The self-organising map facilitates a model independent from cutting conditions. Grasso, Maria Colosimo, and Pacella (2014) study the application of two Principal Component Analysis (PCA) methods on multi-channel data for profile monitoring, in simulation and in a real waterjet cutting test case. The PCA methods slightly outperform location control chart and artificial neural network in terms of fault detection. The autonomous diagnostics using unlabelled data is targeted by Chinnam and Baruah (2009). A Hidden Markov Model caters for an online estimation of remaining-useful-life of drill-bits by incorporation of thrust-force and torque sensors of a CNC machine. An alternative approach for tool failure monitoring in drilling machines is based on measurement of spindle current (Franco-Gasca et al. 2006). Autocorrelation asymmetry analysis applied on data subject to Discrete Wavelet Transformation indicates that spindle current can be related to wear of tool. Improvement of contouring accuracy of CNC machine tools also in consideration of external disturbances is in the scope of Yeh and Sun (2013). A feedforward motion control design and a digital disturbance observer process measurements of control input and position output and can significantly improve tracking accuracy and contouring accuracy.

Some recent studies focus on experimental setups for additive manufacturing. Goegelein et al. (2018) introduce an online process monitoring system based on optical tomography in order to detect defects, i.e. lack of fusion. Studying the potential of polymer pellets as raw material for additive manufacturing is in the scope of Volpato et al. (2015). The authors designed a novel extrusion head which is able to convert polymer pellets into continuous filament. In an experimental setup, Anderegg et al. (2019) provide insights into the design of a fused filament fabrication nozzle which facilitates in-situ measurements of temperature and flow rate during the part build process. Kim et al. (2018) investigate the relationship between deposition defects in a fused deposition modelling machine and the current supplied to the filament feed motor. The defect of unintentional obstruction could be detected with this setup.

2.2. Data-driven studies with additive manufacturing

The rising importance of additive manufacturing also increased related research work in this field, particularly concerning sensor technology combined with ML. Tapia and Elwany (2014) review the scientific literature on process monitoring and control in metal-based additive manufacturing and identify research opportunities. Similar but with a focus on powder bed fusion, Mani et al. (2017) identify needs for measurement science relevant for real-time additive manufacturing process control. The authors provide insights into control schemes, variety of pre-/in-/post-processing measurements, different modelling and simulation approaches and commercially used real-time control.

Manifold experimental studies on quality issues in additive manufacturing were conducted. Huang et al. (2014) provide a statistical predictive modelling in order to predict and improve the quality of printed parts by adaption of input product geometry based on output product deviation. The impact of parameter choice on quality and reproducibility of the printer builds is studied by Sbriglia et al. (2016); based on the frequency response function induced by a shaker, experiments are conducted and examined judging the impact of print layer height, print speed, and tool path. Further approaches for laser powder bed fusion use images (Scime and Beuth. 2018) or acoustic emissions (Shevchik et al. 2018b). The former focusses on anomaly detection and classification by processing images with unsupervised learning. However, the model is appropriate for post-build analysis but not for online analysis due to the quite low classification rate. Shevchik et al. (2018b) elaborate on how data for quality monitoring is acquired using an acoustic sensor installed inside the process chamber. Applying the method of spectral convolutional neural networks results in a correct classification of 83–89% of three different part qualities. In a sensor fusion approach, Rao et al. (2015) use thermocouples, accelerometers, infrared temperature sensor and video for online real-time quality monitoring. The method mix of Bayesian Dirichlet process and evidence theory facilitates real-time identification and correction of process drifts while at the same time improving results of formerly applied methods.

In summary, the related literature reveals the scarcity of laboratory settings and applications for data-oriented experiments in additive manufacturing. Although numerous studies in the field of CNC machinery show the meaningfulness of data-driven approaches, the number of such studies in additive manufacturing is limited. In contrast to that, the huge variety of potential sources for interferences indicates a relevant research gap to be closed in order to pave the way for the development of additive manufacturing.

3. Experimental setup, hardware and data acquisition

To obtain the envisaged contributions of the paper, experiments using appropriate technology are conducted. Therefore, this section elaborates on the experimental setup and the hardware components required for the experiment in this section. The 3D-printer, applied hardware consisting of sensors and data acquisition device, the data acquisition process and the resulting data are described.

3.1. Experimental setup and applied hardware

One of the main objectives is to gain insights into the detection of interferences during an additive manufacturing process, whereby a 3D-printer was used as an example. Interferences have to be induced in a controllable way, and the impact is measured using sensor equipment.

To generate a meaningful dataset with controlled interferences, three types of external disturbances to the printing process were deliberately introduced by the operator.

- Vibrations caused by a mobile phone placed on the build plate of the 3D-printer causing rhythmic acceleration forces on the build plate.
- Vibrations caused by random finger-knocking of the operator on the build plate, resulting in random patterns concerning timing, position and intensity.
- Combination of the first two vibration types.

The different interferences were chosen as reproducible representatives of interferences encountered in the typical operations of the 3D-printer, like vibrations caused by people passing by or a train passing by the building. Obviously, in real-life manufacturing environments there are manifold types of and sources interferences. For example, the conditions for additive manufacturing located in a stamping plant manufacturing car body parts – where intense vibrations every few seconds might occur – vary tremendously from a typical machining company focusing on turning or milling, where continuously a-rhythmical interferences can impact the process. Also structural conditions like vibration absorbers, aircondition, or ambient temperature might influence the type of interference in a real manufacturing case. Although this study does not provide a complete set of interferences, it provides a first modelling approach in order to demonstrate how such interferences in the form of vibrations can be detected and analysed so that potential insights can be fed back to the manufacturing or other processes. In any case, the chosen sources of interference facilitate controlled experimental conditions in a laboratory environment.

For technical specification of the used 3D-printer we refer to A.1 in Appendix. The measurement equipment needs to satisfy several requirements when applied on a 3D-printer use case. Particularly, the 3D-printer must be delicate enough to be influenced by the experiment operator but protected from other sources of interference. The used sensors must be small and of low weight to be mountable on the 3D-printer without impacting the printing process. Furthermore, a high resolution and a wide range of sensor measures are needed to capture the acceleration forces. The data acquisition connected to the sensors needs a high performance to cope with the amount of sensor data.

For data collection, we equipped the 3D-printer with acceleration sensing devices able to detect the acceleration in three axes. Two accelerometer devices were used, where one was placed on the print head and the other one under the build plate. The used accelerometer model is ***i*MEMS® ADXL330**,[1] its specification is given in A.2 in Appendix.

To obtain an additional control variable, the 3D-printer was also equipped with two temperature sensors for measuring temperatures of the nozzle and the environment. One sensor was placed on the Olsson block close to the print head nozzle, whereas the other one was placed on the outer surface of the printer casing. The sensors used are **Labfacility Pt100 Elements, Thin Film (100 Ohm)**,[2] with further details described in A.3 in Appendix.

Data from the sensors was gathered using the data acquisition device ViFDAQ (Lieschnegg et al. 2011). The dimensions of this autonomous acquisition system are 30 mm length, 20 mm width, and 15 mm height. It features synchronous sensor acquisition and synchronous wireless sensor networks with a battery lasting for multiple hours of measurement. The schematic view of the printer and related components can be seen in Figure 1.

3.2. Data acquisition and resultant data

During the process of printing the experiment, the operator intervened by creating the three aforementioned types of interference. The process of printing lasted 4:21 minutes at a sampling frequency of 2 kHz, resulting in a final dataset consisting of roughly 523,000 rows with 8 columns. In addition to that also the time points of intervention were recorded, resulting in a columnar vector of labels indicating the presence or absence of interference. The sampling frequency of 2 kHz is in

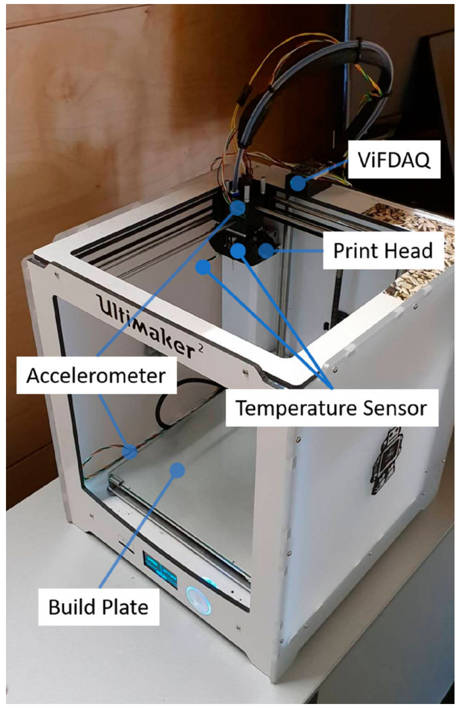

Figure 1. Schematic view of hardware components. Source: Own representation.

line with requirements defined in the current literature on vibrations in the area of additive manufacturing (Duan, Yoon, and Okwudire 2018). This sampling frequency allows – although ending up in a rather large dataset – to acquire data of the induced interferences with a sufficiently high resolution, thereby providing a clear picture of detailed characteristics of induced interferences. Such a high sampling frequency enables to evaluate different data configurations for detection of interferences, and in particular, provides the base for determining minimum resolution requirements for enabling future near real-time detection of interferences. The data gathered from sensors were stored on a microSD card inserted in ViFDAQ in binary form, then transferred to a PC and converted into a comma-separated (CSV) file.

In this CSV file, a row represents a measurement with nine variables as columns:

- `ADC1`, `ADC2`, `ADC3` – x, y, z coordinate values of the printing head accelerometer.
- `ADC4`, `ADC5`, `ADC6` – x, y, z coordinate values of the build plate accelerometer.
- `ADC7`– temperature of print head nozzle measured on the Olsson block.
- `ADC8`– temperature of environment.
- `interference`– class label indicating interference.

`ADC1` to `ADC6` measure the acceleration g, while `ADC7` and `ADC8` hold temperature values in °C. It is important to state that there were no missing values or outliers in the obtained dataset.

For the exploration of data by visual data analytics, values of the print head's and building plate's acceleration forces were plotted. The measured values of the z-axis – which turned out to be the most distinct ones – can be seen in the Figure 2: the block sections (in light red colour) are the time spans during which a known interference occurred. With the naked eye, it is immediately recognisable that there are structural differences between the signals. The magnitude of the head signal appears to have a much broader standard deviation than the magnitude of the acceleration in z-direction recorded on the build plate. Furthermore, the signal recorded on the build plate contains abrupt changes in the behaviour of the signal, which are not present in the signal of the head (or at least cannot be easily spotted).

To compare the two signals more accurately, they where differentiated comparing the rate of change and yielding a more pronounced signal as seen in the top half of Figure 3. The resulting patterns of interference exhibit quite unique features that are characteristic of the two types of interferences. The bottom row in Figure 3 highlights these sections with subplots. The first located on the left shows the interferences caused by knocking on the build plate, the middle subplot visualises the result induced by mobile phone vibrations, and the subplot on the right depicts the combination of both. It features the

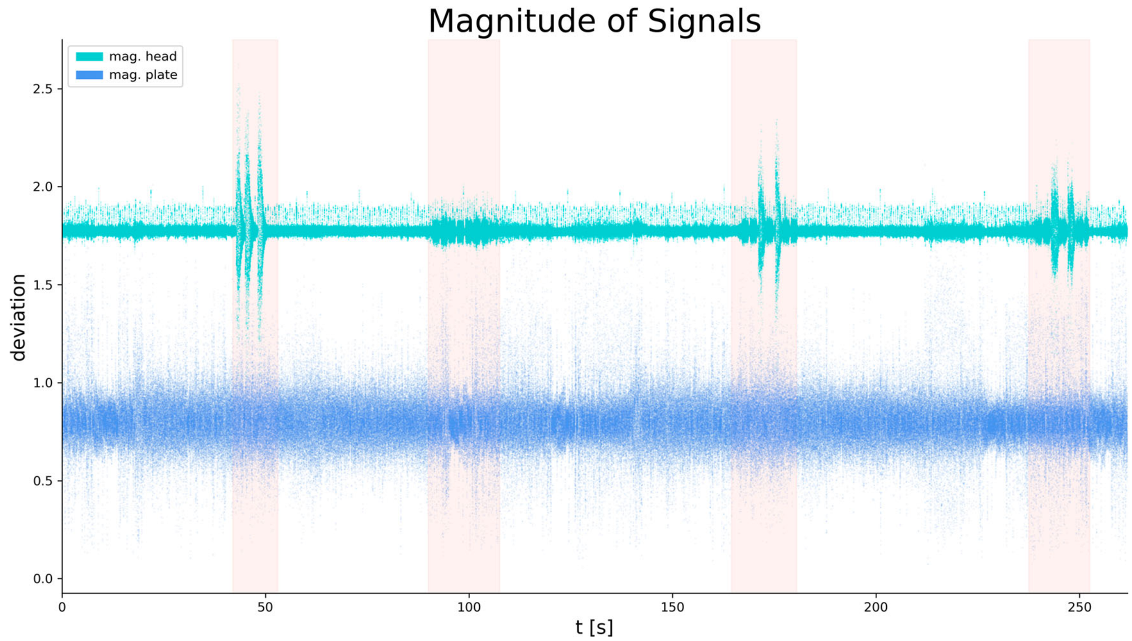

Figure 2. Acceleration in z-direction of print head and the build plate.

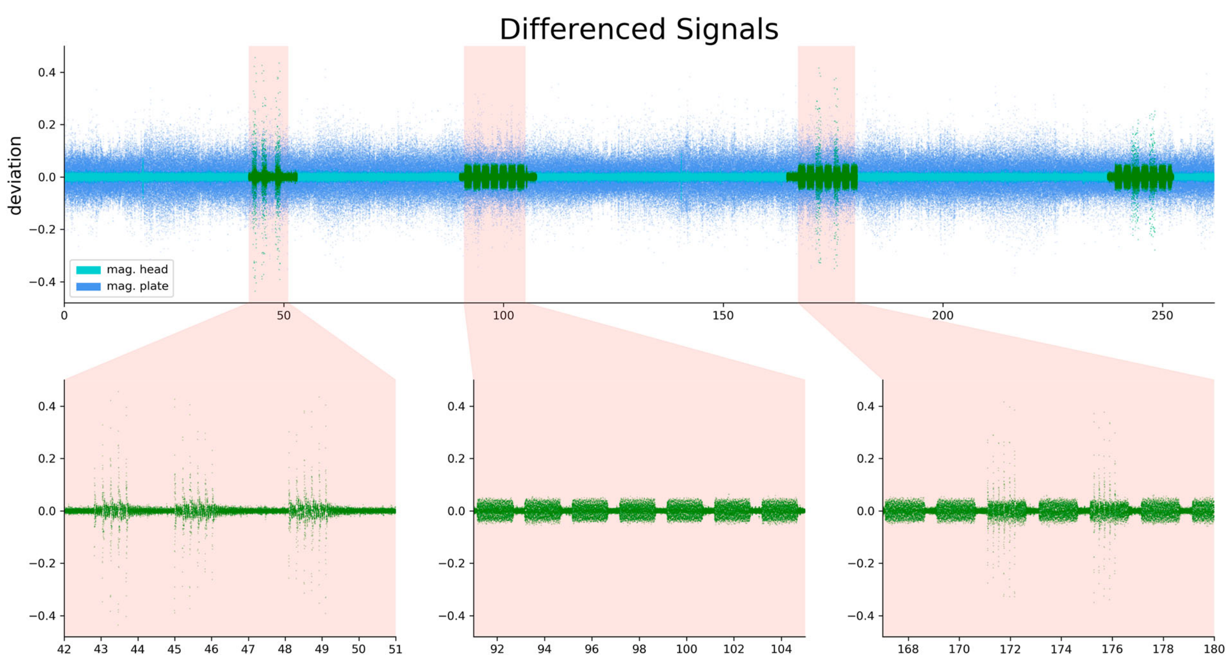

Figure 3. Acceleration in z-direction of print head as well as the build plate: on both signals the difference operation is applied.

large and unevenly scattered points occurring in the first plot, and the dense and more evenly scattered points in the second subplot.

4. Data processing system

As indicated in Lee et al. (2018), a cyber-physical production system that considers existing data sources but also complies with real-time production activities based on big data requires an elaborated architectural framework. This section presents the background on data processing mechanisms followed by the description of the implementation of a DPS in the project.

4.1. Background on mechanisms of data processing

4.1.1. Data pre-processing

Pre-processing is usually the first step in any data processing. Pre-processing aims to transform the raw data into a representation best suited for processing while reducing redundant or irrelevant parts of the data (Hoffmann and Wolff 2015). The aspects of pre-processing and transformations have a strong background in signal processing. In the ML or time series analysis research field, these transformations are often categorised as feature extraction (Zhang et al. 2006).

Data obtained from sensors can be investigated directly in the temporal domain which is most beneficial if the dependency of the current or future data point with past values is of interest. The recorded data can also be transformed into a different domain, most commonly the frequency domain, assuming that the interesting aspects are best modelled by orthogonal base functions (Shumway and Stoffer 2015). While the Fourier transformation is widely used, other integral transformations suit better in some cases, e.g. the Wavelet transformation and its localisation characteristics (Vetterli, Kovačević, and Goyal 2014). If the temporal and the frequency domain are of interest, the recorded data can be transformed using a Time-Frequency Distribution (Boashash 2003).

While most of the transformations were initially formulated for the continuous and analogue case, they can be reformulated to discrete and quantified cases. The underlying processes and signals are still analogue and contentious, but the measurements are represented in a discrete and quantified way to analyse the digital representation of the observed processes and signals. Block transformations split up the input data into smaller units, usually of constant size, and transform the block into a target domain. A common example is the Short-Time Fourier Transformation (STFT) (Mertins 2013).

An important parameter in the continuous and the discrete case is the size of the block. If the block is small, the resolution in the time domain is high but low in the frequency domain and vice versa. This is referred to as the uncertainty relation (Boashash 2003; Mertins 2013). To reduce the effects of circular convolution and to improve the spectral resolution, the blocks are not transformed directly. Zero padding and a window function can be applied. Common window functions include Cos^2, Hann, Hamming, or Gaussian. Nevertheless, the block size sets a hard limit for the spectral resolution which can be approached but never overreached (Madisetti and Williams 1999; Mertins 2013; Allen and Mills 2004).

4.1.2. Feature engineering

Feature engineering follows the pre-processing or feature extraction. It is sometimes also called feature construction and composes higher-level features by combining individually measured properties available as raw data. The higher-level features are highly domain and problem dependent, i.e. they are designed with domain knowledge in mind (Guyon et al. 2006; Tan, Steinbach, and Kumar 2014).

4.1.3. Feature selection

The next step is the feature selection, targeted to pick the features holding the most information and which are easy to use for ML algorithms. Please note that we use terms feature selection and attribute selection synonymously. Features holding little to no relevant information are removed, what can be achieved in different ways (Dietterich 2002; Tan, Steinbach, and Kumar 2014; Zhang et al. 2006):

Wrapper: Different feature subsets are created and the effect on the ML algorithm is measured. The subsets can be generated by starting either with a small feature set and then adding additional features (forward selection) or with a large feature set and continuously removing features (backward elimination). The subset delivering the best results is used for the subsequent learning phase.

Penalty: All features are considered, and a penalty term is assigned to them in the learning model. Unimportant features get low weights, potentially even zero, effectively eliminating them from the feature set. The model can then be used including the low weighted features or re-trained without them.

Filter: The importance of each feature is evaluated by some relevance scores or domain knowledge, without any involvement of the actual learning algorithm. The features with the highest relevance score are then selected for the leaning phase.

Embedded: All features are fed directly into the ML algorithm. The model derived by the ML algorithm is analysed and the relevance of each feature is determined. The model is then re-trained with a subset of the features.

It is important to consider that searching through the total feature space with potential 2^N subsets requires comprehensive search strategies. However, as this procedure is an NP-hard problem, rendering the evaluation of all subsets inefficient or almost impossible (Guyon et al. 2006). This complexity can be reduced by manual selection of features by domain experts, what is the chosen approach in this work.

4.1.4. Machine learning

Many different ML algorithms exist, with the common objective of deriving a model from data so that the model can be used to draw conclusions from new, previously unseen data. The creation of the model is called the training or learning phase and requires training data. How the training phase is organised and what kind of training data is needed, distinguishes different ML algorithms.

Supervised learning requires labelled training data consisting of input data already mapped to the desired target data, like classification on discrete or regression on continuous target data (Tan, Steinbach, and Kumar 2014; Alpaydin 2010). The training and application phases of different classification algorithms vary hugely in its computational complexity. K-nearest neighbour employs so-called lazy learning, i.e. it uses all training examples directly as a model. Decisions trees and its ensemble counterpart random forests model the data by simple rules. Naive Bayes classifiers generate probabilistic models, and Support Vector Machine (SVM) classifiers divide the feature space piecewise linearly (Tan, Steinbach, and Kumar 2014; Sammut and Webb 2011).

Unsupervised learning does not require any labels in the training data, instead it tries to extract structures directly from the unlabelled training data. Clustering is an example, where the algorithms target at finding groups splitting the training data into different partitions. Independent Component Analysis (ICA) and Principal Component Analysis (PCA) also belong to the group of unsupervised learning algorithms. Clustering algorithms can be divided by the different properties they use to describe a model, like centroid-based (e.g. *k*-means clustering), connectivity-based (e.g. hierarchical clustering), distribution-based (e.g. expectation maximisation), or density-based like DBSCAN (Tan, Steinbach, and Kumar 2014; Sammut and Webb 2011).

The third class of ML algorithms is called reinforcement learning, where the algorithm interacts with an agent giving feedback about the decisions the algorithm made. The algorithm adapts its decisions based on the feedback from the agent, therefore the algorithm learns from the agent. Since the feedback can be given constantly, there is no need to separate phases into training and application (Alpaydin 2010; Sammut and Webb 2011).

This work focuses on the application of supervised and unsupervised learning algorithms, employing the widely used F1-measure as an evaluation criterion. The F1-measure is defined as the harmonic mean of the precision and recall metrics. Precision quantifies the fraction of correctly classified positives over all positives of a class. Recall quantifies the correctly classified positives over all true samples of a class (Tan, Steinbach, and Kumar 2014; Sammut and Webb 2011). In this work, the main quality criterion is a technical one: to optimise the overall classification performance. Hence, we do not favour false alarms (for example, classifying a typical operating state as interference) over missed interferences. Since the F1-measure weights both precision and recall equally and hence both precision and recall are of equal importance, it is a suitable metric for the evaluation of our approach. Notwithstanding that, equal weighting is not always desirable, as in some applications differently weighted precision and recall measures are more accurate, especially when it comes to an economic evaluation of the misclassifications' impact. In particular, this concerns cases where costs related to false alarms deviate to a great extent from costs of missed interferences: for example, pausing the production of a simple part for a few seconds due to a single false alarm might not have the same impact on cost as a defective part which is delivered to the customer and mounted in a safety-critical equipment. Such economical impact is highly dependent on the context of manufacturing, and case-specific impact assessment might lead to different choice or weighting, repespectively, of evaluation criteria. For details and a critical view of the F1-measure, the interested reader is referred to Hand and Christen (2018).

4.2. Implementation of the data processing system (DPS)

The purpose of the DPS is to support the evaluation of different ML methods to detect interferences in the workflow of a 3D-printer. The DPS (henceforth referred to as pipeline) is organised as a pipeline and consists of different steps: it offers functionalities such as loading, storing, transforming, and evaluation of results in a coordinated manner. In addition, the pipeline employs transformations of the raw data and selects the most significant features. The pipeline is built around the ML core of the Waikato Environment for Knowledge Analysis, short Weka, (Witten et al. 2017)[3] using the Java programming language.

4.2.1. Structure and components of the machine learning pipeline

The pipeline is structured analogously to the individual components presented in the preceding section providing background related to the DPS. It follows a modular principle and is split into steps which facilitate specific operations, thereby integrating elements of pre-processing, feature extraction, feature selection, and ML. Each step is independent from all others, and the steps of the same type are completely interchangeable. From one pipeline step to the next, data is passed as a so-called pipeline context storing all pipeline-relevant information. A full overview of the pipeline is depicted in Figure 4.

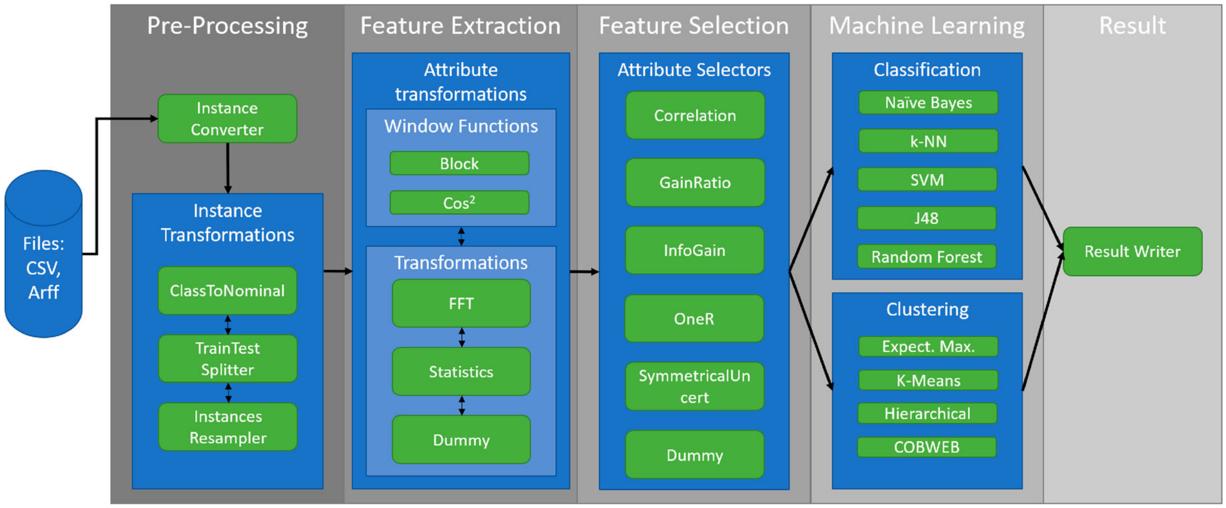

Figure 4. The full pipeline will all available steps grouped by their type. Steps belonging to the same type are always interchangeable.

At the beginning of each pipeline execution, the actual data needs to be read into memory. CSV or Arff (Weka specific file format[4]) are allowed as input files. The `InstanceConverter` transforms the measurements into so-called instances, organising data in a way that Weka algorithms can process it. An instance represents one point in time holding nine attributes (Weka primarily uses attribute instead of features), namely the eight sensor signals `ADC1` to `ADC8` and the class attribute, i.e. the indicator if interference is present required for the evaluation. The `ClassToNominal` instance transformation converts the attributes from numeric to the nominal data type. Separate training and test sets are required to evaluate an ML algorithm, as it is provided by the `TrainTestSplitter` transformer. The `InstanceResampler` can create random subsets of the dataset, whereby the size of the resulting new dataset can be specified as a percentage of the input data.

The pipeline steps belonging to the attribute transformation can be further divided into window functions and transformations. Window functions combine the number of instances configured by the `blockSize` to a single block, integrating all attributes of the original instances. The block and the Cos^2 window functions are independent from each other and are the two window functions implemented in the pipeline. The window functions prepare the data for subsequent transformations but can also be used without further transformations. The `blockSize` determines the temporal resolution, since the subsequent learning is conducted on a per-block basis. A small `blockSize` corresponds to a fine-grained monitoring and a large `blockSize` to a coarse-grained monitoring. The sampling frequency and the `blockSize` determine how fast the monitoring can detect interferences. The sampling rate of 2 kHZ and a `blockSize` between 64 and 1024 allows for monitoring the printer in 32–512 ms periods. Although these are small periods for a simple interference detection, such small periods become a requirement if the interference detection is used as a feedback control to the printer itself. While this is not the scope of this study, it could be an extention following the work of Duan, Yoon, and Okwudire (2018).

The `windowOverlapPercentage` specifies if consecutive blocks should overlap, meaning that not all but only a fraction of the combined instances change from block to block. This can be illustrated with an example. As stated before, the raw dataset has eight attributes and a class attribute. If the block has a length of 8, then 8 instances with 8 attributes each become a new block instance with 64 attributes and 1 class attribute. The class attribute is determined by majority vote and in case of a tie, it is assumed to be a normal instance without an interference.

Regarding the interrelation of window functions and transformations, each pipeline configuration can only include a single-window function but multiple transformations. Window functions can be used without transformations, but vice versa this is not true: transformations must be combined with a window function and optionally with further transformation(s). Fast Fourier transformation (FFT) and statistical transformation are available as pipeline steps. The FFT implements a transformation from the time into the frequency domain with the requirement that the block size is a power of two. To keep up the pipeline links, the block size was chosen as the power of two for any configuration. The number of input and output attributes are equal for the FFT. The statistics transform has a constant number of output attributes regardless of the number of input attributes. It always calculates the Kurtosis, maximum, minimum, mean, skewness, standard deviation, sum, sum squared, variance, root mean square value and the number of peaks per block. The first nine values are calculated by the Apache Commons Math package,[5] while the calculation of root mean square value and the number of peaks were implemented directly. The dummy transform is an identity transformation leaving the values unchanged.

Following the window functions and transformations, the attribute selectors are used to determine the most relevant attributes. Weka provides subset search, subset evaluation and individual search methods for attribute selection. Since the prediction power of single attributes is of interest and not the subsets of attributes, all but one individual search methods were employed[6]:

`Correlation` evaluates the attributes based on their correlation with the class attribute.
`GainRatio` allows to rank attributes by measuring the gain ratio between attributes and the class attribute.
`InfoGain` evaluates attributes based on the information gain of attributes towards the class attribute.
`OneR` assesses attributes using the OneR classifier.[7]
`SymmetricalUncert` measures the symmetrical uncertainty between the attributes and the class.
`Dummy` does not apply any evaluation and all provided attributes are used. This is merely a baseline, ranking all attributes equally.
`ReliefF` samples an instance and considers the distance of the attribute for the same and the different class. Due to performance issues, this search method was omitted.

The search methods provide a ranking from best to worst for all attributes, laying the basis for selecting a subset. The parameter `attributeSelectionRatio` specifies how many attributes should be selected.

The penultimate step in the pipeline is the training of the classification or clustering model. Here, Weka implementations were used with default parameter setting, i.e. without any model parameter adoptions or optimisations. Five different supervised ML classification algorithms from Weka were used:

`IBk`: K-nearest neighbours classifier.
`J48`: C4.5 decision tree classifier.
`LibLINEAR`: Linear Support Vector Machine classifier.[8]
`NaiveBayes`: Naive Bayes classifier.
`RandomForest`: Random forest classifier.

Clustering was used as a starting point for exploration regarding the capability of ML-approaches to distinguish between interference and no interference. In a second step, since labelled data was available, we performed classification in order to compare the results of the basically different approaches of supervised and unsupervised learning (using class WEKA's `ClassificationViaClustering` in the package `weka.classifiers.meta`). The clusters were evaluated by comparing their overlap with the labelled training data, what leads to an F1-measure quantifying how precisely the clusters match with the labelled classes. This implies that the labels are required for evaluation purposes, although Clustering does not require any labelled training data.

The choice of Clustering algorithms was limited to these where the number of clusters can be specified, thereby fixing two clusters. Hence, the applied Clustering approaches split the provided data into two clusters, one including interferences and one without interferences. Please note that this approach is only valid under the conditions that labelled data exists, and data consists of exactly two different classes. Four clustering algorithms were selected from Weka, and for all of them the number of clusters was fixed at a value of two in order to allow an easy comparison with the results of the classification. Due to the runtime complexity only two of the four algorithms implemented in the pipeline were used in the evaluation.

`SimpleKMeans`: k-means clustering.
`EM`: Simple Expectation Maximisation clustering.
`HierarchicalClusterer`: Hierarchical clustering (included in the pipeline but not used due to its runtime complexity).
`Cobweb`: Incremental Hierarchical Conceptual clustering (included in the pipeline but not used due to its runtime complexity).

The final step of the pipeline is the `ResultWriter` logging the results of the attribute selection and the evaluation to a file. Comparing the five classification algorithms with clustering algorithms allows a meta-comparison of supervised classification and unsupervised clustering.

4.2.2. Pipeline combinations

For an extensive test all possible combinations of window functions, transformations, attribute selections and attribute selection ratios have to be tested with all implemented classification and clustering algorithms. All parameter settings amount to 7,200 different parameter settings:

`blockSize`: five different values with 64, 128, 256, 512, 1024.

`windowOverlap`: three different values with 25, 50, 75.
`datasetSize`: three different values with 33, 66, 100.
transformation combinations: eight different values with None, FFT, statistics and all combinations thereof.
attribute selection: five different values with `Correlation`, `GainRatio`, `InfoGain`, `OneR`, `Dummy`.
`attributeSelectionRatio`: four different values with 0.05, 0.1, 0.2, 1.0.

Each of these parameter settings has to be tried against each of the five classifiers and the two clustering algorithms. Hence, there are 36,000 runs of the pipeline for the classification and 14,400 runs for the clustering, making a total of 50,400 pipeline runs. Proper processing of this great number of different combinations motivated the development of the modular pipeline with configurable parameter settings. The execution time highly depends on the parameter configuration, the dataset size and available computing hardware. Hence, no details on runtime are stated.

5. Evaluation

The in-depth evaluation provides insights into the attribute selection, classification and clustering results, showing if and to what extent interferences could be detected. The remainder of the section is dedicated to an analysis of effectiveness of the experimental setup and the efforts required to conduct the study. The analysis of efforts mainly concerns data processing and setting up the equipment/hardware to share experiences about the realisation and effectiveness of such a setup.

5.1. Evaluation of quantitative interference detection

In order to assess the results of the variable models with respect to their classification performance, baseline results for the individual algorithms were created by manually loading and modelling the raw data without any transformations. The algorithms introduced in Section 4.2.1 were applied on this data, whereby blocks of $32, 64, \ldots, 2048, 4096$ sequential data points were used. The best average results across these blocks were achieved by the random forest with a median classification performance measured by F1-measure of 0.9225 (= 92.25%, mean: 92.74%, standard deviation SD: 0.0188), while the algorithm delivering the poorest result was SVM with a median F1-measure of 0.6725 (mean: 66.6%, SD: 0.0121). The best three individual results were 0.941 using Naive Bayes with a block size of 1024, and random forest with F1-measures of 0.955 (block size: 1024) and 0.96 (block size: 4096), respectively. All tests are based on a single run using a random choice of 80% of existing data for training and the remaining data as test data. Although not delivering a perfect classification performance, the baseline results are promising to be improved by the established pipeline.

5.1.1. Feature selection

The relevance of the attributes (features) was evaluated by stopping the execution of the pipeline combinations after the attribute selectors step. Although the different attribute selectors employ differently scaled scores, the results of multiple attribute selectors can be compared by the ranking of attributes. From a qualitative perspective the parameters `blockSize`, `windowOverlap` and `datasetSize` minorly influenced the ranking of different attributes. Concerning the attribute selectors only the `OneR` algorithm provided minor changes in ranking for both window functions (Block and Cos^2) combined with FFT or no transformation. The statistical transformation had the greatest impact on the attribute ranking of all transformations, regardless of the application as single transformation or as an additional transformation after the FFT. Table 1 provides an overview of the average ranks for each attribute and the transformations applied to it. For both the block window transformation in the upper part and the statistical transformation in the lower part of the table, `ADC6` has an average rank of 1, and is therefore the most relevant in any case. It is followed by `ADC1` ranked as the second most important channel of all sensors on the 3D-printer.

In order to further investigate the importance of the `ADC6` attribute, the standard deviations of varying block lengths of `ADC6` were compared. Figure 5 shows a large difference in the standard deviation for `ADC6`, depending on the presence of interferences but far less influenced by the window size. This is contrary to the exemplarily chosen `ADC4`, where no obvious difference in the standard deviation appears, regardless of interference present or not. The evident characteristic explains the importance of `ADC6` over other attributes.

5.1.2. Evaluation of classification methods

The classification performance was evaluated by running the complete pipeline with each of the five classification algorithms in the ML step. The F1-measure was used to evaluate the classifier's performance dependence on different feature

Table 1. The average attribute ranks for various block and statistical transformations show that ADC6 is always the most important attribute.

Block transformations								
Combination	ADC1	ADC2	ADC3	ADC4	ADC5	ADC6	ADC7	ADC8
Block, Stat	2	5	4	6	3	1	8	7
Block, FFT, Stat	3	7	5	6	4	1	8	2
Cos^2, Stat	2	5	4	6	3	1	8	7
Cos^2, FFT, Stat	3	7	5	6	4	1	8	2
Average Rank	2.5	6	4.5	6	3.5	1	8	4.5
Statistical transformations								
Combination	ADC1	ADC2	ADC3	ADC4	ADC5	ADC6	ADC7	ADC8
Block, Stat	2	6	5	7	4	1	8	3
Block, FFT, Stat	2	6	5	7	4	1	8	3
Cos^2, Stat	2	5	4	8	6	1	7	3
Cos^2, FFT, Stat	2	6	4	7	5	1	8	3
Average Rank	2	5.75	4.5	7.25	4.75	1	7.75	3

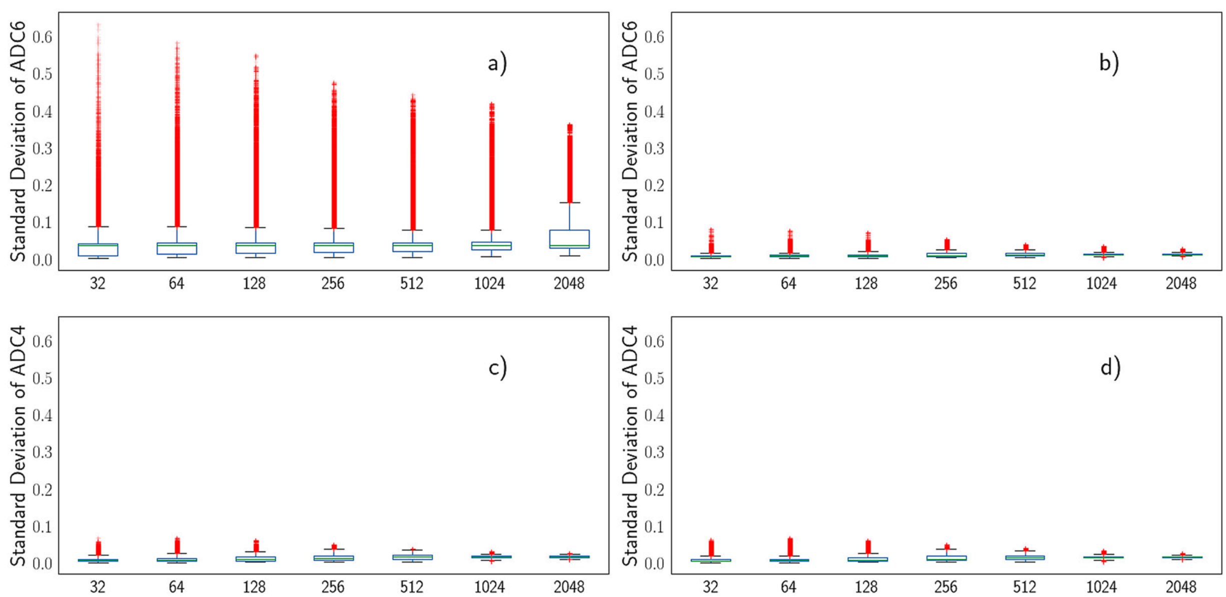

Figure 5. ADC6 shows a large difference in the standard deviation whether there is (a) or is no (b) interference present for different block lengths. This is not the case with ADC4, when comparing its standard deviation in with (c) and without (d) interference, explaining the importance of ADC6 over other attributes.

extraction and attribute selection combinations. For those pipeline runs achieving a classification result of F1-measure equal to one, the underlying attribute sets were counted in a histogram. This established a ranking of attribute sets serving as a proxy of importance, depicted in Figure 6. The results found in the attribute selection were confirmed, since ADC6 and its derived attributes were always part of the most relevant attribute sets, followed by ADC1 and ADC8. In addition, Figure 6 shows that statistical attributes (like kurtosis or skewness) are of prime importance for achieving a perfect F1-measure. The statistical transformation induces an information reduction, since the input block consisting of 32–1024 data points is always transformed into eleven statistical values. Despite the compression, the statistical transformation preserves sufficient information for a perfect classification.

In order to further investigate the relationships between the different sets of parameters and transformations, pivot tables highlighting the influence of different parameter settings were created. Figure 7 shows a pivot table displaying the interdependencies of the F1-measure as a function of a particular classifier, `windowOverlap`, chosen transformation

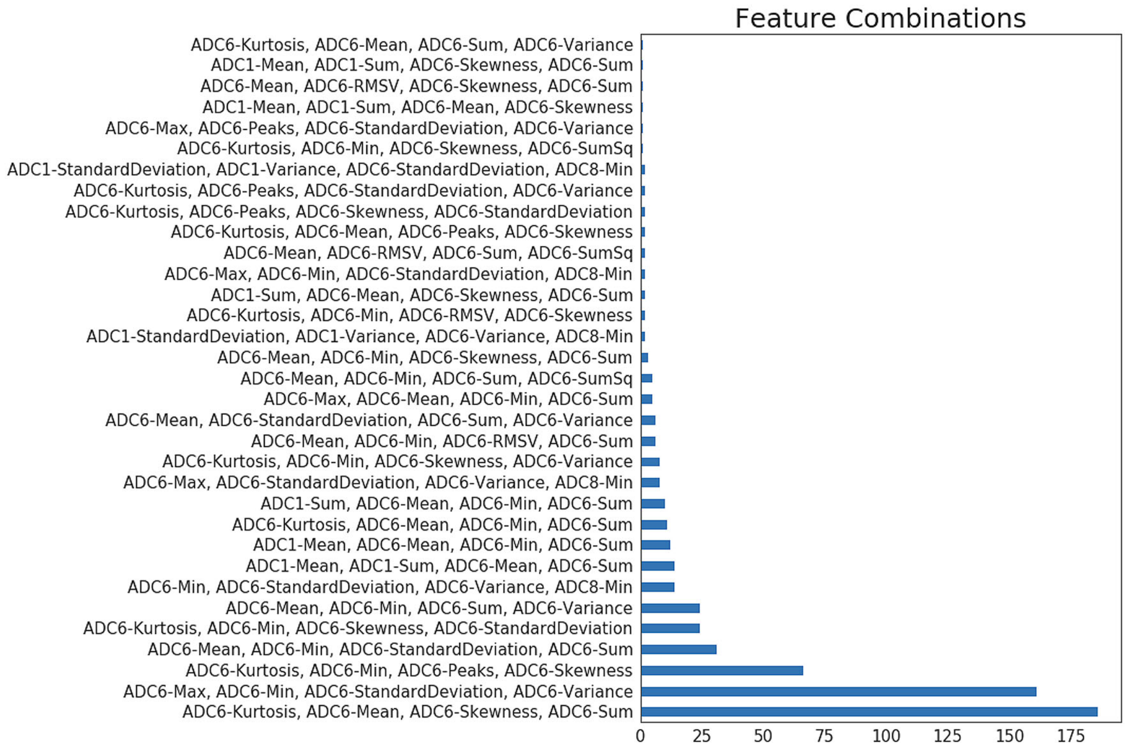

Figure 6. Frequency count of attributes sets achieving a F1-measure equal to one for the classification.

method and `blockSize`. The median was selected as the aggregation function due to its statistical robustness, important since the F1-measure is a mean value.

The usage of different transformations has minor effects on the performance of the `NaiveBayes` and `RandomForest` classifiers, as the median F1-measure is about 1.00 for all of the various cases. In contrast, `J48`, `k-NN`, and in particular, `LinSVM` classifiers show a substantial improvement in performance when using FFT or statistical transformation instead of the identity transformation, i.e. no transformation at all. While an explicit improvement for all block sizes and overlaps can be observed for `J48` (increase of range of F1-measures from 0.84–0.96 in the identity case to 0.99–1.00 with FFT or statistical transformation) and `k-NN` (analogously from 0.65–0.69 to 0.99–1.00), the increase of `LinSVM`-related values is depending on the block size and the overlap. The `LinSVM` classifier in combination with the FFT reacts critically to changes of `blockSize`, since the F1-measure is mainly decreasing with increasing `blockSize`. Similarly, the `windowOverlap` has some minor influence on the `LinSVM` classifier for the FFT and the statistical transformation, as the median F1-measure improves with rising overlap.

In summary, the FFT transformation with a `blockSize` between 64 and 256 proves to be the most robust pipeline combination when comparing all five classifiers, since there is no median F1-measure below 0.91. In the case that data reduction is a primary objective, the statistical transformation with a `blockSize` of 128 also performs well and reduces the number of attributes in a ratio of 128:11. `RandomForest` and `NaiveBayes` are very robust against variation of pipeline parameters and can also cope with raw data without any transformation. `J48` and `kNN` classifiers perform well when an FFT or statistical transformation is applied. `LinSVM` is consistently ranked last, not only due to its sensitivity to the employed transformations but also because of its computational complexity compared to the others. Please note that compared with the baseline introduced in Section 5.1, all classification algorithms improved. For further results of the evaluation of classification methods, we refer to Section A.4 in Appendix.

5.1.3. Evaluation of clustering methods

The frequency count of attributes achieving perfect F1-measures for clustering is shown in Figure 8, depicting a very similar ranking for clustering compared with the classification results. Attribute `ADC6` and its derivatives were almost always part of the most relevant attribute sets, followed by attribute `ADC1` and `ADC8` and their derivatives. The main difference for clustering compared to classification is that the best attribute set achieved a larger number of perfect scores.

Transformer		Id.					FFT					Stats.				
BlockSize		64	128	256	512	1024	64	128	256	512	1024	64	128	256	512	1024
Clf	Overlap															
J48	25	0.95	0.94	0.92	0.89	0.84	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00	1.00	1.00
	50	0.95	0.95	0.94	0.92	0.88	1.00	1.00	1.00	0.99	1.00	0.99	1.00	1.00	1.00	1.00
	75	0.96	0.96	0.94	0.93	0.92	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00
LinSVM	25	0.67	0.67	0.68	0.67	0.65	0.99	0.99	0.91	0.71	0.66	0.72	0.94	0.75	0.68	0.66
	50	0.67	0.67	0.68	0.67	0.68	0.99	1.00	0.99	0.76	0.68	0.80	0.89	0.71	0.67	0.69
	75	0.67	0.67	0.67	0.67	0.67	1.00	1.00	0.99	0.84	0.68	0.86	0.95	0.79	0.76	0.75
NaiveBayes	25	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.99	1.00	1.00	1.00
	50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.99	1.00	1.00	1.00
	75	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.99	1.00	1.00	1.00
RandomForest	25	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00
	50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00
	75	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00
k-NN	25	0.69	0.68	0.68	0.67	0.65	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00	1.00	1.00
	50	0.69	0.68	0.68	0.67	0.68	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00
	75	0.68	0.68	0.67	0.67	0.67	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00

Figure 7. Pivot tables displaying the impact of different classifiers with different `blockSize`, `WindowOverlap` and no (Id., identity), FFT and statistics transformer (stat.) combinations on the median of the F1-measure. The red dashed square highlights the different results of the `LinSVM` for different FFT block sizes.

A further analysis of the different parameters combinations in Figure 9 indicates that `kmeans` cannot cope with identity transformation: the median F1-measure is below 0.7 for all instances. In contrast, in the case of identity transformation `windowOverlap` the results for the `EM` algorithm deviate from 1.0 only when `blockSize` is 1024 and overlap of 25 (F1-measure: 0.82). Almost all other results with FFT or statistical transformation are within the range of 0.97–1.0. As an exception, `kmeans` algorithm with a `blockSize` of 64 and combined with a statistical transformation results in a decreased detection rate of 0.89 for all types of overlap.

Figure 10 investigates the effects of the attribute selection and the `attributeSelectionRatio` without any additional transformation of instances ('dummy'). For both clustering algorithms, the worst choice is not to use any attribute selection since this results in lower F1-measures. For the clustering algorithms, the `attributeSelectionRatio` does not show a great impact on the F1-measure.

The performance of the different clustering algorithms is closely related to the used transformations. Focusing on the different window functions without any transformations, the `EM` algorithm obtained an acceptable F1-measure, but as demonstrated in Figure 10 the `SimpleKMeans` algorithm failed.

Further analysis showed that with regard to FFT, neither the window function nor any downstream statistical transformation had any effect on the perfect F1-measure. In a similar way, the statistical transformation combined with `SimpleKMeans` yields good results. The clustering quality leads to a clear recommendation for the `EM` algorithm, since it is far less influenced by any transformation parameters. The `kMeans` algorithm requires an FFT or statistical transformation to achieve similar performance values. Since the `kMeans` is computationally less expensive, this might be a worthwhile trade-off for many applications. For the most robust clustering, the `EM` algorithm in combination with a `blockSize` between 128 and 512 and a `windowOverlap` of 50% can be recommended.

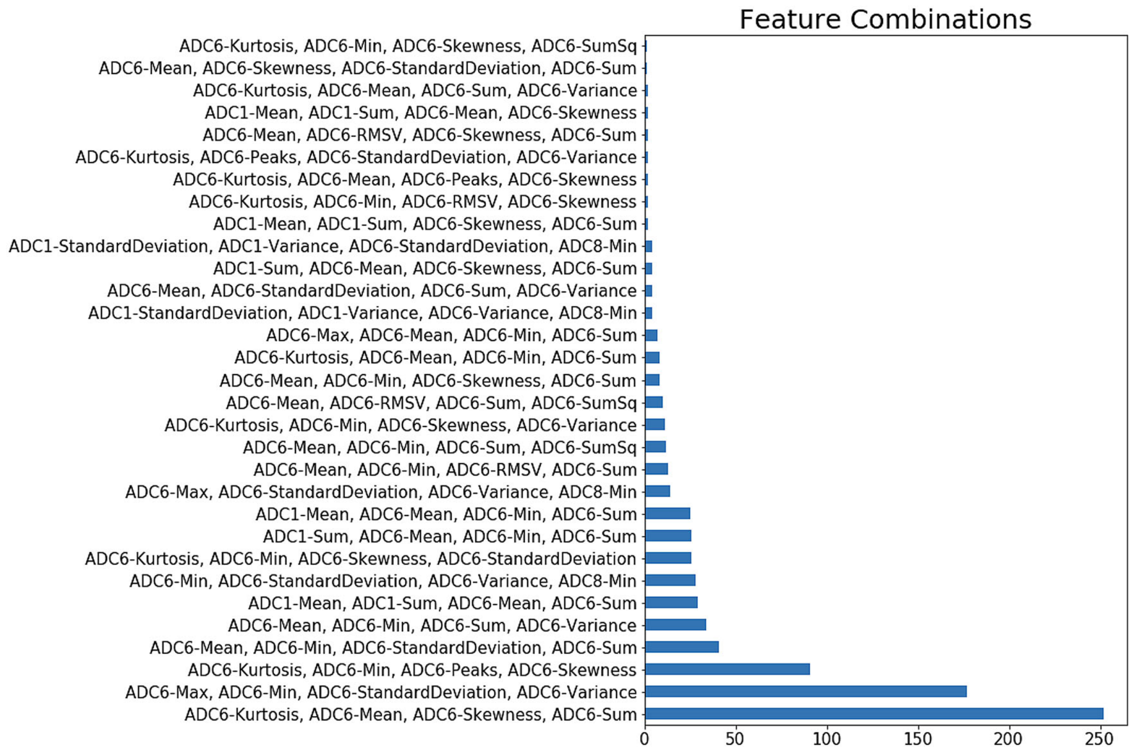

Figure 8. Frequency count of attribute sets achieving a perfect F1-measure for clustering. Like for classification, `ADC6` is the most relevant attribute.

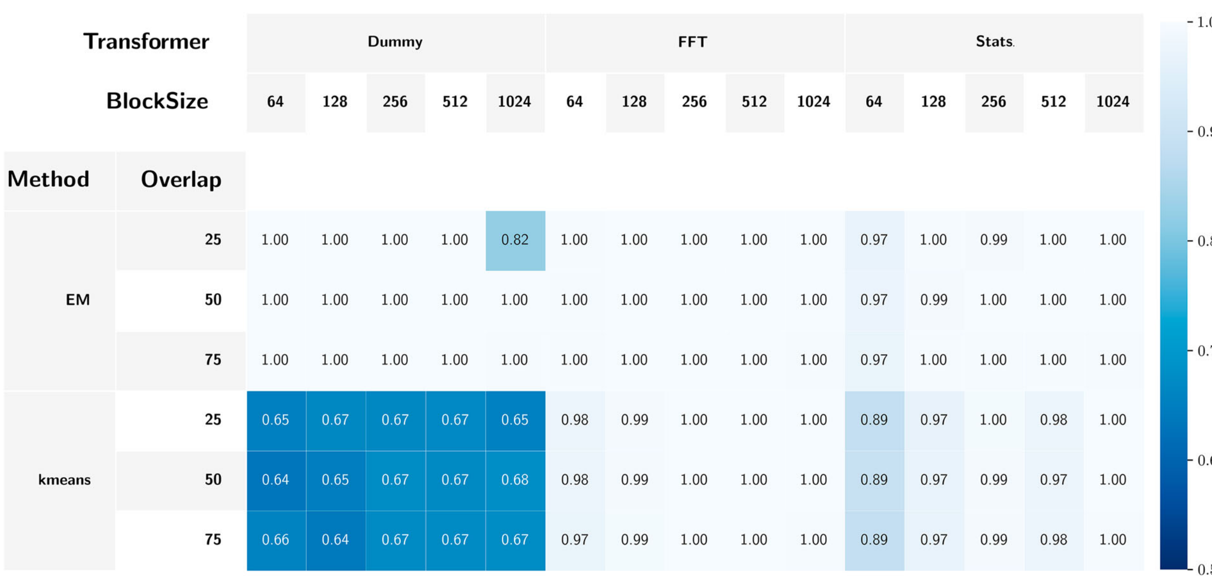

Figure 9. Pivot tables displaying the impact of different clustering algorithms with different `blockSize`, `WindowOverlap` and no (Id., identity), FFT and statistics transformer (stat.) combinations on the median of the F1-measure.

5.1.4. Evaluation of feature importance

The evaluation of attribute selection, classification and clustering methods emphasised that the *x*-axis acceleration sensor mounted on the print head (ADC1), the *z*-axis acceleration sensor mounted on the build plate (ADC6) and the environment temperature (ADC8) are the most important attributes for interference detection. In the case of an induced interference, it is obvious that the mounting of the sensors – especially the acceleration sensors – determines on which axes data can be measured. Beyond that, also not being empirically proven, the basic working principles of the 3D-printer and the setup of

AdditionalTransformer		InstancesTransformationDummy																	
AttributeSelector		Correlation			DummyAS			GainRatio			InfoGain			OneR			SymmetricalUncert		
AttributeSelectorRatio		0.05	0.1	0.2	0.05	0.1	0.2	0.05	0.1	0.2	0.05	0.1	0.2	0.05	0.1	0.2	0.05	0.1	0.2
Method	**WindowFunction**																		
EM	InstancesWindowFunctionBlock	1.00	1.00	1.00	0.63	0.63	0.57	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	InstancesWindowFunctionCosSquare	1.00	1.00	1.00	0.63	0.65	0.61	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
kmeans	InstancesWindowFunctionBlock	0.99	1.00	1.00	0.63	0.64	0.63	0.99	1.00	1.00	0.99	1.00	1.00	0.99	1.00	1.00	0.99	1.00	1.00
	InstancesWindowFunctionCosSquare	0.97	0.97	0.97	0.64	0.66	0.63	0.97	0.96	0.97	0.97	0.96	0.97	0.97	0.96	0.97	0.97	0.96	0.97

Figure 10. Pivot tables displaying the relationship between clustering algorithms, window function, transformation, attribute selector and attribute selector ratio by the median of the F1-measure. Similar to classification, all attribute selection algorithms outperform the dummy attribute selection.

how to induce interferences allow to explain the outcome. First, the build plate can only move in z-direction (up and down). Thus, only the z-component of induced interferences is reflected in measured sensor data what results in the importance of ADC6. Second, the print head has two degrees of freedom, being able to move in x- and y-directions. Although we induced vibrations on build plate, both print head and build plate are placed in the same printer case made of metal. Any vibration induced on the build plate can thus also be measured in a damped manner on the print head. The respective interference signal was easy to spot even with bare eye in data of ADC1. Less founded is the impact of ADC8, as induced interferences in this experiment do not change the ambient temperature. However, the proximity of the experimenters to the 3D-printer case during inducing interferences might change the ambient temperature. In particular, for this case, but also for other attributes a detailed root cause analysis is required to substantiate the explanatory model. Definitely the observations emphasise the need to integrate sensor measurements for obtaining further potential factors which might also have an influence (e.g. humidity).

5.2. *Implementation and data processing: effort, effectiveness, and managerial insights*

The installation of the measurement system (ViFDAQ) together with all the required sensors was done in less than one day. One of the two accelerometers was fixed under the build plate. The cable had to be placed in such a manner that it does not interfere with the vertical movement of the build plate. The second accelerometer was installed at the side of the print head, thereby paying attention that no interference with the horizontal movement of the print head along the x- and y-axes occurs. The installation procedure for the two sensors was rather straightforward and not time-consuming. Contrary to that, the installation of the temperature sensor on the print head (nozzle) required more effort. The requirement that nozzles should be interchangeable after mounting the temperature sensor led to the idea to place it on Olsson block, requiring to disassemble the whole print head. Since the print head nozzle is heated by the Olsson block, one can assume that they have the same temperature. The second temperature sensor was placed on the inner side of the printer casing as a reference point. Cables were used to connect all the mentioned sensors together with the measurement system, whereas the system itself was placed on the outer side of the printer casing.

The second aspect to be considered is the development and implementation of the DPS. The effort spent on the data acquisition was moderate, as the recording of a few minutes of measurement data was straightforward due to the easy-to-use graphical user interface of the measurement system. Nevertheless, the acquired data had to be transferred from ViFDAQ to the PC, taking up to several hours depending on the file size. Subsequently, the binary data had to be parsed to obtain a processable CSV format, resulting in several hours of effort. In addition to the creation of readable data, data were manually labelled. The process of manual labelling the training data has its downsides, mainly due to is time consumption and imperfections in the alignment of the boundary between two classes, i.e. the denoted boundary between existing interference or normal operating state. These imperfections are shown in Figure 11, where the class label changes before the statistics of the measurements reflect an interference. In general, the class label was assigned conservatively, meaning that it was preferred to extend the interference indicator over the true length of the interference so that no interference was missed. These imperfections are further attenuated by the block-based processing, since the class label does not change within a block.

The development of the software pipeline started right after labelling data, aiming for the evaluation of different combinations of feature extraction techniques and classification/clustering algorithms. The objective was to identify the

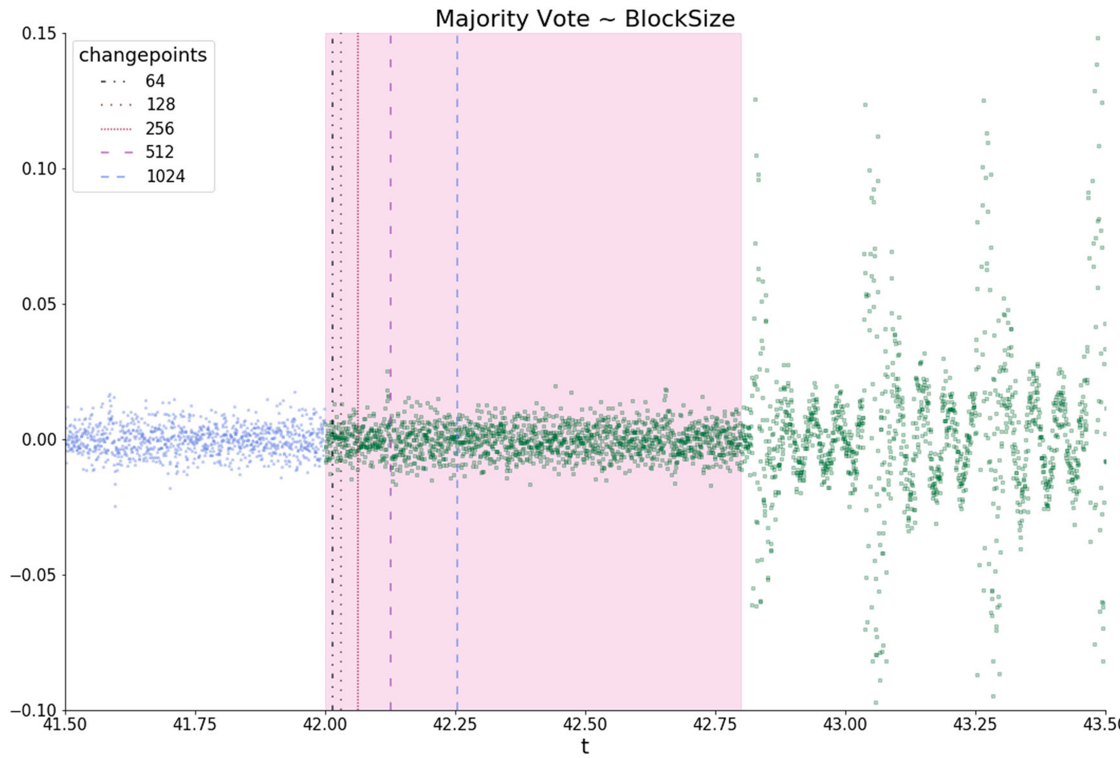

Figure 11. (Conservative) Assignment of interference label: measurements without interference present are labelled as interference so that no interferences are missed. This can be seen as the measurement colour changes from blue (no interference) to green (interference) before the interference can be seen in the measurements. The actual decision boundary also depends on the block length, as denoted by the different vertical lines representing the change points for each block length.

combination yielding the best performance in terms of detecting the induced interferences. A significant amount of time was spent for software development, and it was considerably more effort than the preceding steps. Nevertheless, the processing pipeline is a requirement to evaluate the 50,400 different combinations of the setup. Manual scheduling of all evaluations would reduce the development time, but the effort would increase disproportionately and render the undertaking infeasible. This is particularly true since different pipeline combinations can differ in execution times by orders of magnitude. With the pipeline setup it was possible to distribute the execution of different combinations across multiple computers, thereby speeding up the evaluation of complex pipeline configurations.

6. Conclusion and outlook

The study in this article dealt with setting up an experimental environment for detection of interferences in an additive manufacturing process. The experimental setup and the applied hardware proved to be effective for conducting experiments. Mounting sensors with high resolution was done with little effort. With a matching pre-processing and transformation, the DPS could accurately detect interference from the data obtained from the sensors. Although the creation of a complete DPS is a complex and laborious task, it reduced overall time and effort, since manual execution of all possible combinations of window functions, transformations and respective classification/clustering algorithms is nearly impossible. Related to this, the quality of the experiment rises as a huge amount of parameter combinations can be studied and the best-fitting chosen. As the developed pipeline is modularly designed, further data transformation options with subsequent revaluation can be added easily.

Pre-processing data is certainly a key factor for obtaining a good performance, as almost all classifiers and clustering algorithms show very good detection rates. Hence, the decision which algorithm to choose can rely on different factors like the required time to build and evaluate a model. One of the main findings of the study is that a further reduction of observed data to statistical aggregates is possible with unchanged detection precision. This has large implications on real-world use cases at industrial machinery. For instance, data could be reduced by simple statistical aggregates, thus reducing the amount of data to be examined. Consequently, less storage space and less resources for data processing would be required. Additionally, the feature selection is a valuable step in the DPS since is was shown almost all transformations combined with a non-dummy attribute selector lead to a steep increase of detection performance. The experiments also showed that some of the used algorithms are not performant when applied on large datasets due to the required computational resources. The software was running on a cluster for several days in order to assess the performance of all combinations of data transformations and classification/clustering algorithms, hence considerable computing resources were used to produce the relevant results.

Being a starting point for further research, the incorporated interferences do not represent the entire set of interferences occurring in industry. Despite that, to some extent, the chosen approach allows a general interference detection. The detection model was trained and tested with two reproducible representatives (and their combination) of common interferences, and the most important features for the detection of interferences turned out to be standard deviation and amplitude of the signal. Since the interferences were not specifically adapted, the pipeline and the ML algorithms used are robust and generic enough to also work with other interferences but same statistical characteristics. Furthermore, with labelled training data the ML algorithms can also be re-trained to work with other types of interference as well. Beyond that, attributes composed of several sensor values – for example, magnitude of acceleration values – might lead to an even more generalised interference detection by uncoupling attributes from sensors' mounting.

In order to further optimise the DPS and to increase the number of potential (real-world) applications, the exploration of further algorithms as well as additional types of interferences (complementing vibrations) and sensors to measure potential sources of interference (e.g. humidity), respectively, are needed. In addition, an automated data transfer, processing and automated analysis of results including visualisations is required for industrial applications. To simplify the interaction, the pipeline could be integrated in a GUI like the WekaGUI where pipeline parameters can be configured easily. Finally, the realisation of a similar setup in an industrial context would allow to further evaluate the insights gained from this study, in particular when linked with product data from the field to explore the impact of interferences over the product lifecycle.

Notes

1. https://www.sparkfun.com/datasheets/Components/ADXL330_0.pdf.
2. https://www.labfacility.com/media/attachment/file/pdfs/data-sheet-pt100-pt1000-thin-film-detectors.pdf.
3. https://www.cs.waikato.ac.nz/ml/weka/.
4. https://www.cs.waikato.ac.nz/ml/weka/arff.html.
5. https://commons.apache.org/proper/commons-math/userguide/stat.html.
6. http://weka.sourceforge.net/doc.dev/weka/attributeSelection/package-summary.html.
7. http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/OneR.html.
8. The library needs to be downloaded separately: https://mvnrepository.com/artifact/nz.ac.waikato.cms.weka/LibLINEAR.

Acknowledgments

The authors thank Roman Kern (Know-Center) and Alexander Stocker (VIRTUAL VEHICLE Research Center) for their valuable comments and feedback.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The work was supported by the project Power Semiconductor and Electronics Manufacturing 4.0 (Semi40), under grant agreement number 692466. The project is co-funded by grants from Austria, Germany, Italy, France, Portugal and – Electronic Component System for European Leadership Joint Undertaking (ECSEL JU). VIRTUAL VEHICLE Research Center is funded within the COMET – Competence Centers for Excellent Technologies – program by the Austrian Federal Ministry for Transport, Innovation and Technology (bmvit), the Austrian Federal Ministry of Digital and Economic Affairs (bmdw), the Austrian Research Promotion Agency (FFG), the province of Styria and the Styrian Business Promotion Agency (SFG). The COMET program is administrated by FFG. The Know-Center is funded within the Austrian COMET Program – Competence Centers for Excellent Technologies – under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Digital and Economic Affairs and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

Allen, Ronald L., and Duncan W. Mills. 2004. *Signal Analysis*. New York: IEEE Press.
Alpaydin, Ethem. 2010. *Introduction to Machine Learning*. 2nd ed. Cambridge: MIT Press.

Altintas, Yusuf. 2012. *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*. 2nd ed. Cambridge: Cambridge University Press.

Ambhore, Nitin, Dinesh Kamble, Satish Chinchanikar, and Vishal Wayal. 2015. "Tool Condition Monitoring System: A Review." *Materials Today: Proceedings of 4th International Conference on Materials Processing and Characterzation* 2 (4): 3419–3428. http://www.sciencedirect.com/science/article/pii/S22147853150 05623.

Anderegg, David A., Hunter A. Bryant, Devante C. Ruffin, Stephen M. Skrip, Jacob J. Fallon, Eric L. Gilmer, and Michael J. Bortner. 2019. "In-situ Monitoring of Polymer Flow Temperature and Pressure in Extrusion Based Additive Manufacturing." *Additive Manufacturing* 26: 76–83. http://www.sciencedirect.com/science/article/pii/S22148604183 07097.

Balsamo, Vittorio, Alessandra Caggiano, Krzysztof Jemielniak, Joanna Kossakowska, Miroslaw Nejman, and Roberto Teti. 2016. "Multi Sensor Signal Processing for Catastrophic Tool Failure Detection in Turning." *Procedia CIRP* 41: 939–944. Research and Innovation in Manufacturing: Key Enabling Technologies for the Factories of the Future – Proceedings of the 48th CIRP Conference on Manufacturing Systems. http://www.sciencedirect.com/science/article/pii/S22128271160 00214.

Battaïa, Olga, Alexandre Dolgui, Sunderesh S. Heragu, Semyon M. Meerkov, and Manoj Kumar Tiwari. 2018. "Design for Manufacturing and Assembly/disassembly: Joint Design of Products and Production Systems." *International Journal of Production Research* 56 (24): 7181–7189. https://doi.org/10.1080/00207543.2018.1549795.

Boashash, Boualem, ed.2003. *Time-Frequency Signal Analysis and Processing*. 1st ed. Oxford: Elsevier..

Calignano, Flaviana, Diego Manfredi, Elisa Paola Ambrosio, Sara Biamino, Mariangela Lombardi, Eleonora Atzeni, Alessandro Salmi, Paolo Minetola, Luca Iuliano, and Paolo Fino. 2017. "Overview on Additive Manufacturing Technologies." *Proceedings of the IEEE* 105 (4): 593–612.

Cemernek, David, Heimo Gursch, and Roman Kern. 2017. "Big Data as a Promoter of Industry 4.0: Lessons of the Semiconductor Industry." In *15th International Conference on Industrial Informatics (INDIN 2017)*, New York, July, 239–244. IEEE. http://ieeexplore.ieee.org/document/8104778/.

Chinnam, Ratna Babu, and Pundarikaksha Baruah. 2009. "Autonomous Diagnostics and Prognostics in Machining Processes Through Competitive Learning-Driven HMM-Based Clustering." *International Journal of Production Research* 47 (23): 6739–6758. https://doi.org/10.1080/00207540802232930.

D'Aveni, Richard. 2015. "The 3-D Printing Revolution." *Harvard Business Review* 93 (5): 40–48. https://hbr.org/2015/05/the-3-d-printing-revolution.

Dietterich, Thomas G. 2002. "Machine Learning for Sequential Data: A Review." In *Structural, Syntactic, and Statistical Pattern Recognition*, edited by Terry Caelli, Adnan Amin, Robert P. W. Duin, Dick de Ridder, and Mohamed Kamel, Vol. 2396 of *Lecture Notes in Computer Science (LNCS)*, Berlin, Germany, August 15–30. IAPR, Springer. https://link.springer.com/chapter/10.1007/3-540-70659-3_2.

Downey, Jonathan, Sebastian Bombiński, Mirosław Nejman, and Krzysztof Jemielniak. 2015. "Automatic Multiple Sensor Data Acquisition System in a Real-time Production Environment." *Procedia CIRP* 33: 215–220. 9th CIRP Conference on Intelligent Computation in Manufacturing Engineering – CIRP ICME '14. http://www.sciencedirect.com/science/article/pii/S22128271150 06824.

Downey, Jonathan, Denis O'Sullivan, Miroslaw Nejmen, Sebastian Bombinski, Paul O'Leary, Ramesh Raghavendra, and Krzysztof Jemielniak. 2016. "Real Time Monitoring of the CNC Process in a Production Environment- the Data Collection & Analysis Phase." *Procedia CIRP* 41: 920–926. Research and Innovation in Manufacturing: Key Enabling Technologies for the Factories of the Future – Proceedings of the 48th CIRP Conference on Manufacturing Systems. http://www.sciencedirect.com/science/article/pii/S2212827115010872.

Duan, Molong, Deokkyun Yoon, and Chinedum E. Okwudire. 2018. "A Limited-Preview Filtered B-Spline Approach to Tracking Control – With Application to Vibration-Induced Error Compensation of a 3D Printer." *Mechatronics* 56: 287–296. http://www.sciencedirect.com/science/article/pii/S09574158173 01277.

Franco-Gasca, Luis Alfonso, Gilberto Herrera-Ruiz, Rocío Peniche-Vera, René de Jesús Romero-Troncoso, and Wbaldo Leal-Tafolla. 2006. "Sensorless Tool Failure Monitoring System for Drilling Machines." *International Journal of Machine Tools and Manufacture* 46 (3): 381–386. http://www.sciencedirect.com/science/article/pii/S08906955050 01288.

Gao, Wei, Yunbo Zhang, Devarajan Ramanujan, Karthik Ramani, Yong Chen, Christopher B. Williams, Charlie C. L. Wang, Yung C. Shin, Song Zhang, and Pablo D. Zavattieri. 2015. "The Status, Challenges, and Future of Additive Manufacturing in Engineering." *Computer-Aided Design* 69: 65–89. http://www.sciencedirect.com/science/article/pii/S00104485150 00469.

Ghobakhloo, Morteza. 2019. "Determinants of Information and Digital Technology Implementation for Smart Manufacturing." *International Journal of Production Research* 188 (1): 1–22. https://doi.org/10.1080/00207543.2019.1630775.

Goegelein, Anian, Alexander Ladewig, Guenter Zenzinger, and Joachim Bamberg. 2018. "Process monitoring of additive manufacturing by using optical tomography." In *Proceeding of 14th Quantitative InfraRed Thermography Conference*, 266–272

Grasso, Marco, Bianca Maria Colosimo, and M. Pacella. 2014. "Profile Monitoring Via Sensor Fusion: The Use of PCA Methods for Multi-Channel Data." *International Journal of Production Research* 52 (20): 6110–6135. https://doi.org/10.1080/00207543.2014.916431.

Guyon, Isabelle, Steve Gunn, Masoud Nikravesh, and Lotfi A. Zadeh, eds. 2006. *Feature Extraction*. Vol. 207 *Studies in Fuzziness and Soft Computing*. Berlin: Springer.

Hahn, Gerd J. 2019. "Industry 4.0: a Supply Chain Innovation Perspective." *International Journal of Production Research* 54 (3): 1–17. https://doi.org/10.1080/00207543.2019.1641642.

Hand, David, and Peter Christen. 2018. “A Note on Using the F-measure for Evaluating Record Linkage Algorithms.” *Statistics and Computing* 28 (3): 539–547. https://app.dimensions.ai/details/publication/pub.1084928040 and http://spiral.imperial.ac.uk/bitstream/10044/1/46235/2/stco-d-16-00349-final.pdf.

Hoffmann, Rüdiger, and Matthias Wolff. 2015. *Intelligente Signalverarbeitung 2*. 2nd ed. Berlin: Springer. https://link.springer.com/book/10.1007%2F978-3-662-46726-8.

Huang, Yong, Ming C. Leu, Jyoti Mazumder, and Alkan Donmez. 2015. “Additive Manufacturing: Current State, Future Potential, Gaps and Needs, and Recommendations.” *Journal of Manufacturing Science and Engineering* 137 (1): 1–10. http://manufacturingscience.asmedigitalcollection.asme.org/ar ticle.aspx?articleid = 1913683.

Huang, Qiang, Hadis Nouri, Kai Xu, Yong Chen, Sobambo Sosina, and Tirthankar Dasgupta. 2014. “Statistical Predictive Modeling and Compensation of Geometric Deviations of Three-Dimensional Printed Products.” *Journal of Manufacturing Science and Engineering* 136 (6): 061008 06101 061008–1 –061008–10 http://dx.doi.org/10.1115/1.4028510

Ivanov, Dmitry, Alexandre Dolgui, and Boris Sokolov. 2019. “The Impact of Digital Technology and Industry 4.0 on the Ripple Effect and Supply Chain Risk Analytics.” *International Journal of Production Research* 57 (3): 829–846. https://doi.org/10.1080/00207543.2018.1488086.

Jayaram, Jayanth, and Shawnee Vickery. 2018. “The Role of Modularity in the Supply Chain Context: Current Trends and Future Research Directions.” *International Journal of Production Research* 56 (20): 6568–6574. https://doi.org/10.1080/00207543.2018.1484574.

Kim, C., D. Espalin, A. Cuaron, M. A. Perez, E. MacDonald, and R. B. Wicker. 2018. “UnobtrusiveIn SituDiagnostics of Filament-Fed Material Extrusion Additive Manufacturing.” *IEEE Transactions on Components, Packaging and Manufacturing Technology* 8 (8): 1469–1476.

Lasi, Heiner, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. 2014. “Industrie 4.0.” *WIRTSCHAFTSINFORMATIK* 56 (4): 261–264. https://link.springer.com/article/10.1007/s11576-014-0424-4.

Lee, Jay, Hung-An Kao, and Shanhu Yang. 2014. “Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment.” *Procedia CIRP* 16: 3–8. Product Services Systems and Value Creation. Proceedings of the 6th CIRP Conference on Industrial Product-Service Systems. http://www.sciencedirect.com/science/article/pii/S22128271140 00857.

Lee, Junehyuck, Sang Do Noh, Hyun-Jung Kim, and Yong-Shin Kang. 2018. “Implementation of Cyber-Physical Production Systems for Quality Prediction and Operation Control in Metal Casting.” *Sensors* 18 (5): 1–17. http://www.mdpi.com/1424-8220/18/5/1428.

Lieschnegg, Michael, Bernhard Lechner, Anton Fuchs, and Olivia Mariani. 2011. “Versatile Sensor Platform for Autonomous Sensing in Automotive Applications.” *International Journal on Smart Sensing and Intelligent Systems* 4 (3): 496–507. https://www.exeley.com/in_jour_smart_sensing_and_intelligent_ systems/doi/10.21307/ijssis-2017-453.

Liu, Q., Chuck Zhang, and A. C. Lin. 1998. “Pattern Recognition of Machine Tool Faults With a Fuzzy Mathematics Algorithm.” *International Journal of Production Research* 36 (8): 2301–2314. https://doi.org/10.1080/002075498192913.

Lu, Q. Y., and C. H. Wong. 2018. “Additive Manufacturing Process Monitoring and Control by Non-destructive Testing Techniques: Challenges and in-process Monitoring.” *Virtual and Physical Prototyping* 13 (2): 39–48. https://doi.org/10.1080/17452759.2017.1351201.

Madisetti, Vijay, and Douglas B. Williams, eds. 1999. *Digital Signal Processing Handbook*. 1st ed. Boca Raton, FL: CRC Press LLC.

Mani, Mahesh, Brandon M. Lane, M. Alkan Donmez, Shaw C. Feng, and Shawn P. Moylan. 2017. “A Review on Measurement Science Needs for Real-Time Control of Additive Manufacturing Metal Powder Bed Fusion Processes.” *International Journal of Production Research* 55 (5): 1400–1418. https://doi.org/10.1080/00207543.2016.1223378.

Mertins, Alfred. 2013. *Signaltheorie*. 3rd ed. Wiesbaden: Springer.

Oropallo, William, and Les A. Piegl. 2016. “Ten Challenges in 3D Printing.” *Engineering with Computers*32 (1): 135–148. http://dx.doi.org/10.1007/s00366-015-0407-0.

Rao, Prahalad K., Jia Peter Liu, David Roberson, Zhenyu James Kong, and Christopher Williams. 2015. “Online Real-Time Quality Monitoring in Additive Manufacturing Processes Using Heterogeneous Sensors.” *Journal of Manufacturing Science and Engineering* 137 (6): 06100706101061007–1 –061007–12. http://dx.doi.org/10.1115/1.4029823.

Sammut, Claude, and Geoffrey I. Webb, eds. 2011. *Encyclopedia of Machine Learning*. 1st ed. New York: Springer. https://link.springer.com/referencework/10.1007/978-0-387-301 64-8.

Sbriglia, Lexey R., Andrew M. Baker, James M. Thompson, Robert V. Morgan, Adam J. Wachtor, and John D. Bernardin. 2016. “Embedding Sensors in FDM Plastic Parts During Additive Manufacturing.” In *Topics in Modal Analysis & Testing, Volume 10*, edited by Michael Mains, Conference Proceedings of the Society for Experimental Mechanics (CPSEMS), 205–214. Cham: Springer.

Scime, Luke, and Jack Beuth. 2018. “Anomaly Detection and Classification in a Laser Powder Bed Additive Manufacturing Process Using a Trained Computer Vision Algorithm.” *Additive Manufacturing*19: 114–126. http://www.sciencedirect.com/science/article/pii/S221486041730180X.

Shevchik, Sergey A., Christoph Kenel, Christian Leinenbach, and Kilian Wasmer. 2018b. “Acoustic Emission For In Situ Quality Monitoring in Additive Manufacturing Using Spectral Convolutional Neural Networks.” *Additive Manufacturing* 21: 598–604. http://www.sciencedirect.com/science/article/pii/S22148604173 0132X.

Shumway, Robert H., and David S. Stoffer. 2015. *Time Series Analysis and Its Applications*. 3rd ed. New York: Springer.

Suh, Suk-Hwan, Seong-Kyoon Kang, Dae-Hyuk Chung, and Ian Stroud. 2008. *Theory and Design of CNC Systems*. 1st ed., Advanced Manufacturing. London: Springer.

Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. 2014. *Introduction to Data Mining*. 1st ed. Harlow: Pearson.

Tapia, Gustavo, and Alaa Elwany. 2014. "A Review on Process Monitoring and Control in Metal-Based Additive Manufacturing." *Journal of Manufacturing Science and Engineering* 136 (6): 06080106081060801–1 –060801–10. http://dx.doi.org/10.1115/1.4028540.
Teti, Roberto, Krzysztof Jemielniak, Garret O'Donnell, and David Dornfeld. 2010. "Advanced Monitoring of Machining Operations." *CIRP Annals* 59 (2): 717–739. http://www.sciencedirect.com/science/article/pii/S00078506100 01976.
Tofail, Syed A. M., Elias P. Koumoulos, Amit Bandyopadhyay, Susmita Bose, Lisa O'Donoghue, and Costas Charitidis. 2018. "Additive Manufacturing: Scientific and Technological Challenges, Market Uptake and Opportunities." *Materials Today* 21 (1): 22–37. http://www.sciencedirect.com/science/article/pii/S13697021173 01773.
Vetterli, Martin, Jelena Kovačević, and Vivek K. Goyal. 2014. *Foundations of Signal Processing*. Cambridge: Cambridge University Press.
Volpato, N., D. Kretschek, J. A. Foggiatto, and C. M. Gomez da Silva Cruz. 2015. "Experimental Analysis of An Extrusion System for Additive Manufacturing Based on Polymer Pellets." *The International Journal of Advanced Manufacturing Technology* 81 (9): 1519–1531. https://doi.org/10.1007/s00170-015-7300-2.
Wang, W. H., Geok-Soon Hong, Y. S. Wong, and K. P. Zhu. 2007. "Sensor Fusion for Online Tool Condition Monitoring in Milling." *International Journal of Production Research* 45 (21): 5095–5116. https://doi.org/10.1080/00207540500536913.
Wang, Lidong, and Guanghui Wang. 2016. "Big Data in Cyber-Physical Systems, Digital Manufacturing and Industry 4.0." *International Journal of Engineering and Manufacturing* 6 (4): 1–8.
Winkelhaus, Sven, and Eric H. Grosse. 2019. "Logistics 4.0: a Systematic Review Towards a New Logistics System." *International Journal of Production Research* 94 (4): 1–26. https://doi.org/10.1080/00207543.2019.1612964.
Witten, Ian H., Eibe Frank, Mark A. Hall, and Christopher J. Pal, eds. 2017. *Data Mining*. 4th ed. Cambridge: Morgan Kaufmann. http://www.sciencedirect.com/science/article/pii/B97801280429 15000015.
Yeh, Syh-Shiuh, and Jin-Tsu Sun. 2013. "Feedforward Motion Control Design for Improving Contouring Accuracy of CNC Machine Tools." In *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2013)*, Vol. 1, Hong Kong, March, 111–116. International Association of Engineers, Newswood Limited.
Zhang, Hui, Tu Bao Ho, Yang Zhang, and Mao-Song Lin. 2006. "Unsupervised Feature Extraction for Time Series Clustering Using Orthogonal Wavelet Transform." *Informatica* 30 (3): 305–319. http://www.informatica.si/index.php/informatica/article/view/ 98.

Appendix

A.1. Used 3D-printer

The 3D-printer used in this experiment is an Ultimaker 2, widely endorsed as a reliable, (cost-)efficient and user friendly 3D-printer. The schematic view of the printer can be seen in Figure 1. Ultimaker 2 features a build area of 223 ×223 × 205 mm and has an open filament construction. It is important to state that this type of printer's casing construction implies higher sensitivity towards the changes of ambient temperature compared to 3D-printers with closed casing/cabinet which allow easier regulation of the ambient temperature. This aspect is important due to the fact that instabilities in ambient temperature could negatively influence the printing quality. Furthermore, it implies that Ultimaker 2 can print only the shrink-insensitive filaments like PLA filament used in the experiment. Shrink-sensitive materials have to be processed in closed-cabinet printers with controlled cabinet temperature. Although closed-cabinet printers usually perform better in terms of performance and resulting product quality, the lab conditions allowed the usage of the cost-efficient Ultimaker 2.

Ultimaker 2 is equipped with a heated build plate made of glass which is able to heat up to temperatures from 50°C to 100°C in under 4 minutes. Heating aims at preventing the warping of the first few layers of the printed object on the build plate due to by variation of temperature. The printer is also equipped with a Bowden-style extruder. At the hot end of the extruder a 0.4 mm nozzle is mounted, replaceable by 0.25 mm, 0.6 mm and 0.8 mm nozzles to vary diameters of filaments by making use of an Olsson block. The nozzle itself heats up a temperature of 180°C to 260°C and reaches this temperature in around a minute.

A.2. Acceleration sensor specification

The ADXL330 is a small, thin, low power, complete three-axis accelerometer with signal conditioned voltage outputs. It is built on a single monolithic integrated chip. The product measures acceleration on range of ± 3 g. It is able to measure the static acceleration of gravity, as well as induced acceleration resulting from motion, shock, or vibration. Measurable bandwidths are on range of 0.5 –1600 Hz for X- and Y-axes, and a range of 0.5–550 Hz for the Z-axis. The mentioned range of measurable frequencies on x-, y- and z-axes describe the minimal and maximal analogue signal frequency but not maximal sampling, and signal can be measured in this range without loss of precision. The ADXL330 is available in a small, low profile, 4 × 4 × 1.45 mm, 16-lead, plastic lead frame chip scale package. This sensor is intended to be used in cost-sensitive, low-power, motion- and tilt-sensing applications. Examples are mobile devices, gaming systems, disk drive protection, image stabilisation, sport and health devices.

A.3. Temperature sensor specification

The thin film sensors are built to these specifications:

- Pt100 elements belong to Class A, B and 1/3 DIN
- For use from -50°C to $+500$°C
- Thin film construction
- Suitable for surface and immersion applications because of the provided protection

- Vibration resistant
- With a thermal response of 0.1s
- Stability $\pm 0.05\%$

A.4. Further results of evaluation

In Figure A1 it is apparent that the `datasetSize` has no influence on the F1-measure.

Figure A2 investigates the influence of the attribute selection ratio and the classification performance. All classifiers in combination with any transformations had a better F1-measure when used with an attribute selector compared to the dummy attribute selector. The dummy attribute selector does no actual attribute selection but just selects attributes based on the natural ordering. Also a tendency of higher F1-measures for lower attribute selection ratios can be seen, indicating that most of the information is embedded in few attributes. Especially, the `LinSVM` classifier's F1-measure decreases with an increasing *attributeSelectionRatio*, i.e. the performance decreases with increasing number of attributes. This can be interpreted that `LinSVM` struggles to separate the many attributes with low information content hence performing better when the number of relevant features is low. Similar to the classification, the parameter `datasetSize` has no effect on the F1-measure.

Concerning the window functions, no main performance differences could be found for any of the classifiers, merely for `LinSVM` and `k-NN` the F1-measure drops if no further transformation is used (see Figure A3).

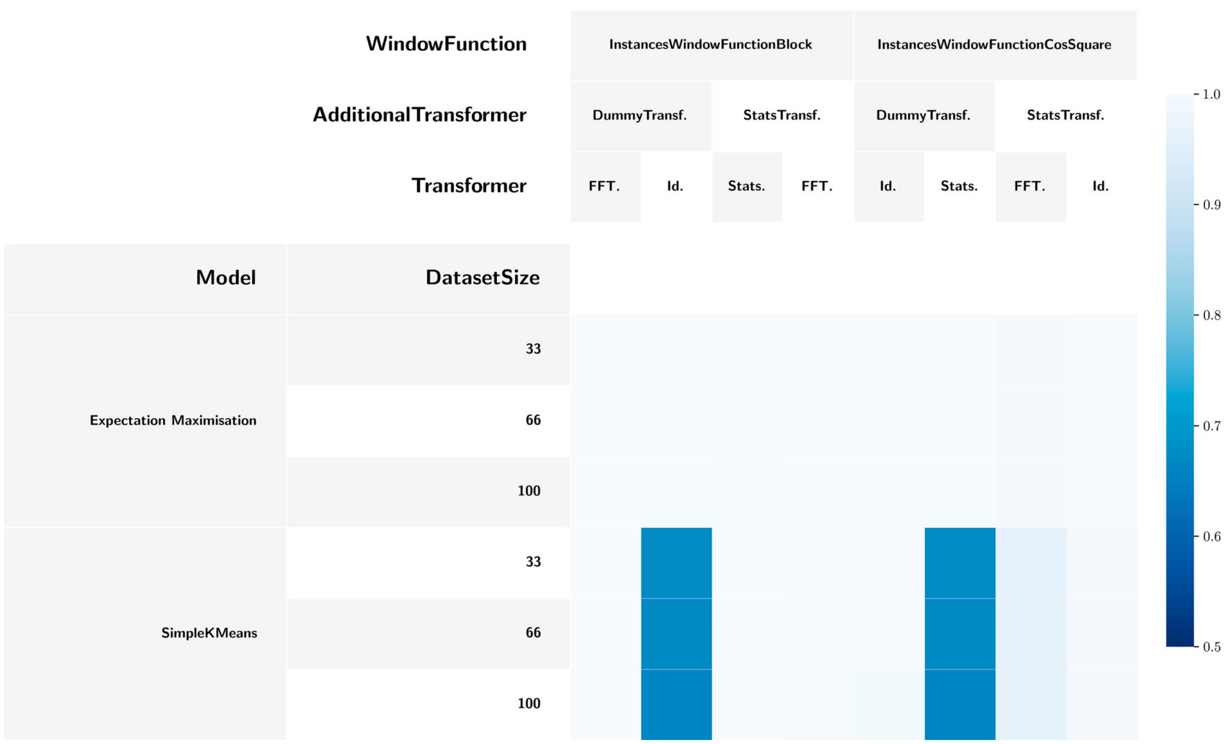

Figure A1. Schematic F1-measures for clustering algorithms and influence of parameter `datasetSize`.

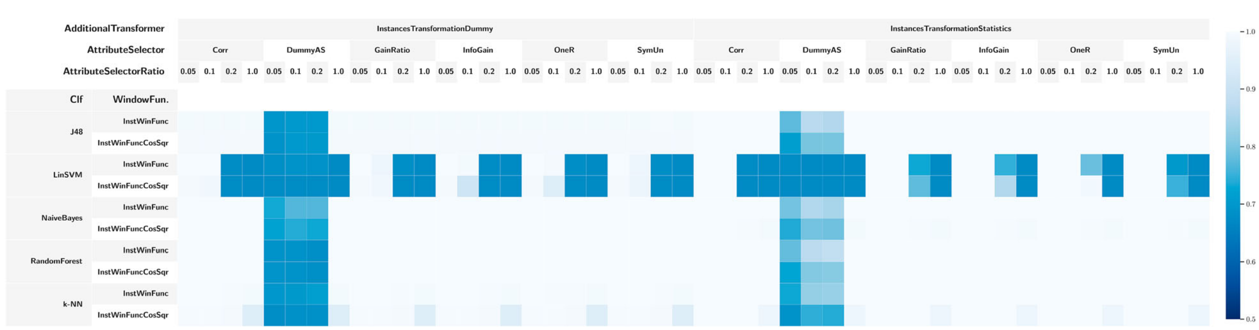

Figure A2. Pivot tables displaying the relationship between Classifiers, window function, transformation, attribute selector and attribute selector ratio by the median of the F1-measure. All attribute selection algorithms outperform the dummy attribute selection which simply selects attributes according to their natural order.

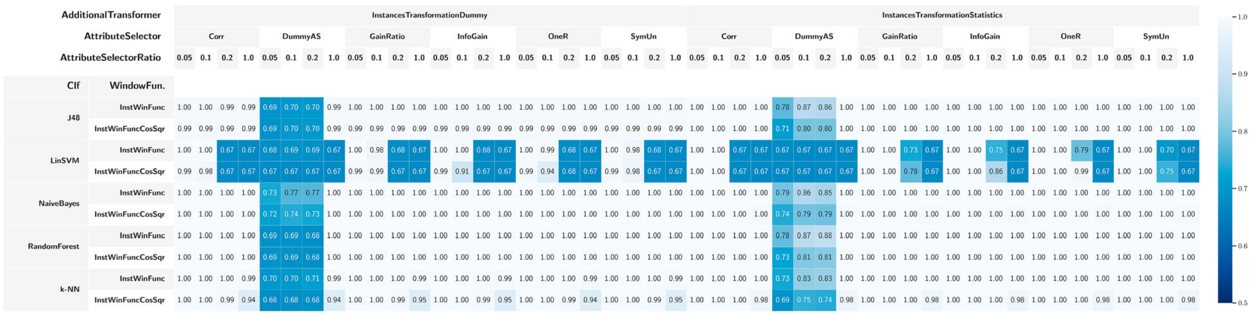

Figure A3. Schematic F1-measures for classification algorithms with different window function combinations.