Data-driven Performance Metrics are improving the Efficiency of Mining Machines

Prof. Dr.-Ing. Andreas Merchiers, Bochum University of Applied Sciences, Bochum, Germany
Prof. Dr. rer. nat. Henrik Blunck, Bochum University of Applied Sciences, Bochum, Germany
Arne Köller, M. Sc., Institute for Advanced Mining Technologies (AMT), RWTH Aachen University, Aachen, Germany
Dr.-Ing. Christian Gierga, Eickhoff Bergbautechnik GmbH, Bochum, Germany

The ongoing development of sensor and actuator technology opens up huge opportunities for manufacturers and users of underground mining machines. The acquisition and analysis of sensor data contributes to the process optimisation of mining operations and also helps improve machine production methods, which in turn offers a significant potential for cutting costs. This paper explains the possibilities and particular challenges arising in a number of areas, including data mining processes, manual rule-based modelling, data architecture, data visualisation, statistical analysis and machine learning (ML). Case studies of drum shearer loaders and continuous miners produced by the Eickhoff Group are used to illustrate the knowledge gain and the opportunities now arising in this context.

Mining • Digitisation • Data science • Machines • Process optimisation • Cost saving

1 Data Science in an industrial Environment

Companies in almost every branch of industry have for quite a few years now been coming under growing pressure to 'do their homework' in matters relating to digitisation. The growth in automation and the increased use of on-board sensor and actuator technology, combined with ever more extensive system integration, means that manufacturers and users alike are now holding huge quantities of machine and process data.

The machinery manufacturers have seized on this development as an opportunity to strengthen their competitive position by offering a range of additional digital services. The primary objectives here are to optimise process control on the user side and, at the same time, to leverage savings potential through condition-based maintenance strategies by means of condition monitoring. Moreover, an analysis of the field data also holds enormous potential for the manufacturers' own operations. This provides the R&D and design departments with a much better understanding of the actual operating conditions and usage behaviour of their products (Fig. 1) – yielding complete and objectively analysable data time series instead of merely snapshots of data paired with proprietary heuristic knowledge of the personell. On this basis the machine's performance characteristics and functionalities can be adapted and optimised in a much more targeted way and stockholding and maintenance strategies can be designed along customer- and machine-specific lines.

The industry has come to realise that the sustainable consolidation of its own competitive standing – both at the point of production and at the equipment manufacturers – now lies in the application of data analysis and that this technology transcends its own operational boundaries and has interfaces into peripheral systems. Yet the actual spectrum of requirements is still extremely diffuse: For some, the mere visualisation of sensor data with a focus on the reliability and validity of the data is sufficient, while others see data analytics as the key to achieving process stability and reproducibility, regardless of what is actually known about the cause-effect relationship in complex systems. Common to all intentions, however, is the goal of going beyond the heuristic approaches of the lean world and arriving at evidence-based proposals based on statistical models and evaluations.

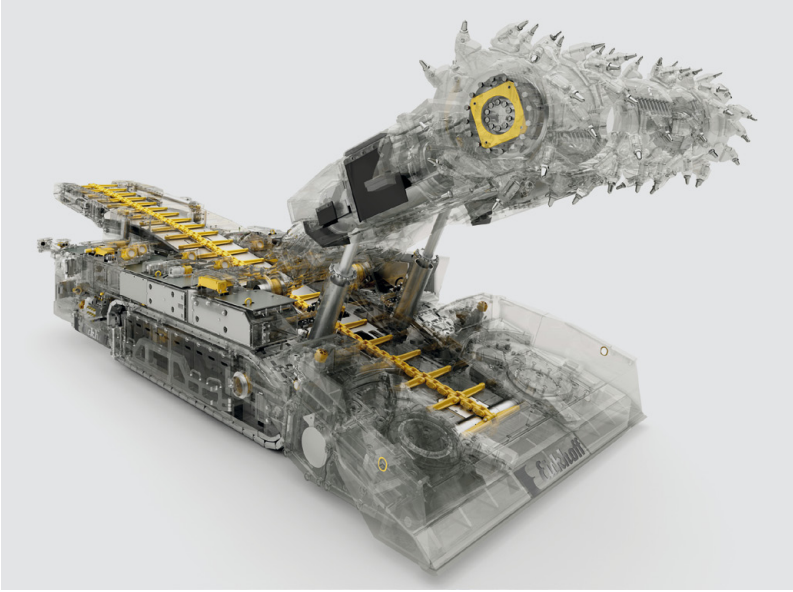

Fig. 1: Continuous Miner machine manufactured by Eickhoff Bergbautechnik
Photo: Godehardt

Success will therefore essentially and always depend on the ability to control and manage the typical stages and aspects of the data pipeline – from sensor data extraction, data buffering, storage and processing to visualisation.

2 Data Science in the Coal Mining Industry

For the underground mining sector focused in this technical paper – using two example scenarios, namely long-

Fig. 2: Data science algorithms and methods for underground mining equipment manufactured by the Eickhoff Group

wall face extraction by shearer loader and room and pillar working with continuous miners – the developments under way are similar to those taking place in other industrial sectors. The potential for expanding the performance spectrum and existing range of business models is enormous. Most of the machines now being deployed below ground are fitted with sensors that are designed to provide the information base for machine guidance and control. The data collected by the sensors can also be stored for further analysis at some point in the future. The acquired sensor data are accessible via the increasingly well developed IIoT infrastructure of the superordinated mine information systems and in this way can be used for machine-to-machine communication [1], SCADA ('supervisory control and data acquisition') visualisation and mine-external storage and processing [2]. Channelling the data from several machines of similar or different type into an information system for machine-overarching storage and processing also enables existing hypotheses and data-analytical models to be cross-checked against entire fleets of machines and quantitative assessments of mining subsystems and overall systems. This latter option can have a significant impact on the operational and strategic optimisation decisions taken by mine operators and machine manufacturers.

Real-case examples that have been applied by the Eickhoff Group are presented below in order to illustrate the opportunities now provided by using data science algorithms and methods on mining machines and to set out the infrastructure requirements needed for their implementation (**Fig. 2**).

The operational deployment of data science algorithms and methods can be divided into the algorithmic treatment of data for the purpose of information and knowledge acquisition, on one hand, and the creation of infrastructure-related conditions for the systematic and continuous integration of these algorithms into a mining machinery manufacturer's products and processes, on the other. In this context Section 3 (below) uses the example of a system for identifying the extraction cycles of continuous miner machines, and the associated process of calculating the excavated volume, in order to show algorithmic data treatment in action, i. e. the Data Mining Process with modelling of a manual rule-based system. Section 4 then presents an infrastructure-based data architecture that provides for the efficient calculation of arbitrary algorithms based on arbitrary data. Following on from this Section 5 then provides further examples of application scenarios for data science algorithms and methods. Finally, Section 6 contains a general summing-up and looks ahead to ongoing developments in this field.

3 Handling of Sensor Data using the Example of Extraction Cycles and Mining Volumes for Continuous Miners

This section starts by presenting the algorithmic treatment of sensor data from a continuous miner machine for the purpose of identifying the extraction cycles and calculating the extracted volume of material, this serving as an example of how data science algorithms and methods can be implemented in the field. The cutting and/or extraction cycle, in idealised form, comprises the following individual stages that are abstracted for the purpose of cycle identification (**Fig. 3**):

1 Approach run to the coal face
2 Sumping into the roof of the heading
3 Lowering the cutting boom/downward cut
4 Cutting into the floor and backing out with lowered cutting boom to clear the floor material

The individual stages that make up this extraction cycle may vary according to the colliery and geological conditions [3, pp. 22f, 56f]. However, the idealised sequence of the individual steps of the extraction cycle

cannot generally be identified in the actual positional data generated by the cutter boom, which is why the statistical threshold value and control systems have to be supplemented by methodologies borrowed from the world of data science. As will be seen later, this includes, for example, explorative-visual data analysis techniques. When the extraction cycle is identified and defined this information can then be used to calculate performance metrics for the cycle duration and extracted volume.

The operational-algorithmic procedure for detecting extraction cycles and calculating the volume of mineral extracted typically corresponds to the flow diagram depicted in Fig. 4, where the following stages apply:

- Selection
- Date pre-processing
- Transformation
- Data mining
- Interpretation/evaluation

In the following, the steps depicted in the flow diagram in Fig. 3 are worked through in the real-case example. The data source is a relational database into which the sensor data from the continuous miner is channelled via a realtime protocol as a communications interface. The **selection** of the data required for the particular application can therefore be made using SQL retrieval language. Sensor data relating to the motor currents and the cylinder position of the boom are especially useful for identifying the cycle and calculating the volume of mineral excavated. Moreover, in the **data pre-processing stage** threshold values for the motor current can be established for motor start-up and switch-off in order to be able to identify such data outliers and deal with them appropriately. In this particular case there is no need to carry out a data **transformation** process.

In the present context **data mining** can be defined as a specific operation aimed at acquiring information and knowledge from pre-processed data. The data mining process uses data science algorithms and methods to generate knowledge from the data sets [5, p. 24]. The following specific process options are available to choose from:

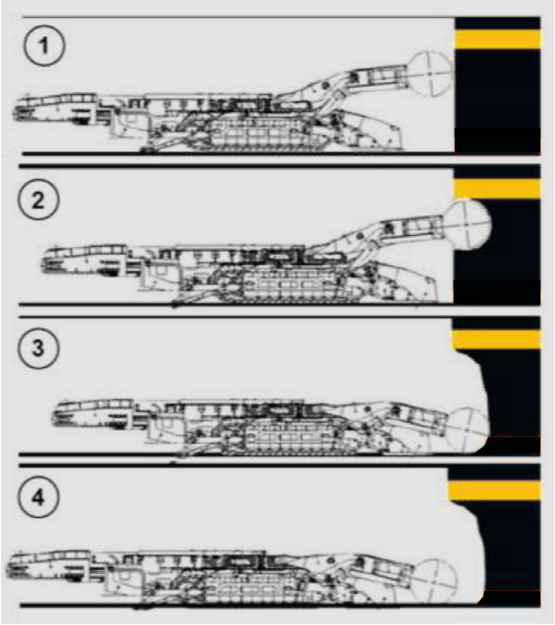

Fig. 3: Simplified continuous miner extraction cycle focused on cycle identification

- **Classification:** a process for predicting a categorical variable based on sensor data
- **Regression:** a process for predicting a numerical variable based on sensor data
- **Clustering:** a process for dividing the consolidated data into groups presenting similar characteristic feature

In the context of learning algorithms, also known as 'machine learning', classification and regression processes are referred to as 'supervised', whereas clustering processes are called 'unsupervised' [5, p. 49 ff]. In the context of non-learning algorithms all three processes listed above can be understood to be manual rule-based methods. Models are formed via the sensor data both in the context of learning and non-learning algorithms. According to the above listing these models perform the role of predicting a variable (classification and regression) or of dividing the data into similar groups (clustering).

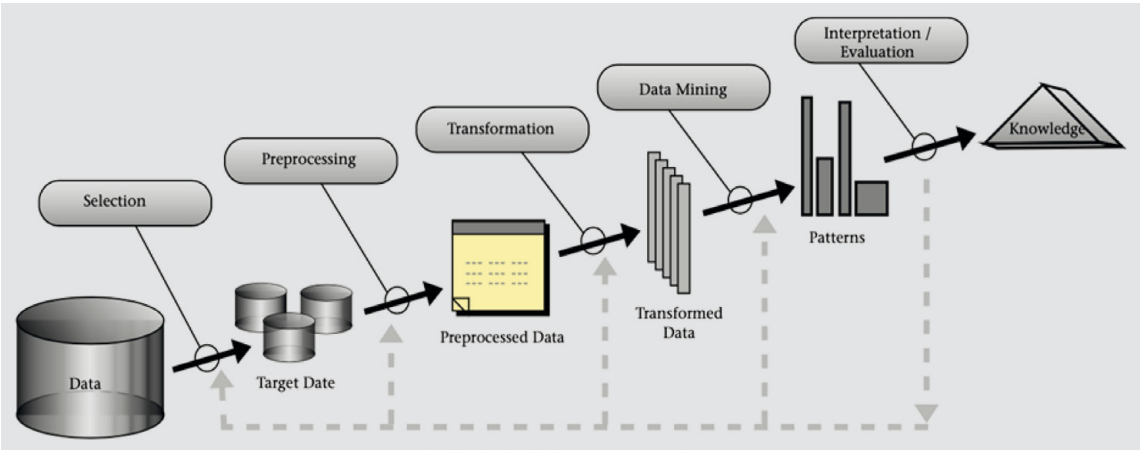

Fig. 4: Flow chart for the algorithmic processing of data for knowledge acquisition [4]

Special Topic

Fig. 5: Continuous miner extraction cycles on two temporal zoom levels
The blue graph line represents the height position of the cutting boom depending on the time. The orange and turquoise lines represent the cutter motor currents and the yellow line the hydraulic motor current. The bar chart (below) represents the cutting phases (green), cutting preparation phases (light green) and other (orange), as floor cleaning phases.

In the example case in point a manually generated, rule-based classification model is selected for detecting the individual steps in the extraction cycle. The reason for this choice is that the individual stages that make up the extraction cycle can be described with sufficient precision using a fairly small set of rules that have been very selectively defined using process know-how. An explorative data visualisation, whose results are explained below, is carried out beforehand in order to determine the set of rules required. **Fig. 5** shows a sequence of several extraction cycles over a longer period of time. The blue graph represents the height position of the hydraulic cylinder that operates the boom. Clearly recognisable are the individual extraction cycles and, in particular, those points in time when the continuous miner lowers its boom for the downwards cut. At these points in time the cutter motor currents (shown in orange) also present higher amplitudes and variances. **Fig. 5** also shows, however, that the actual extraction cycles do not fully correspond to the ideal, resp. simplified, version of an extraction cycle but rather produce a less consistent overall picture. Individual geological circumstances, and even the cutting technique used by the machine operator, can result in variations in both the maximum and the minimum height of the cutter boom and these positions can differ quite significantly over successive winning cycles. It is also evident from the sensor data that the downwards cut can be made at different speeds and with pauses of different length in between. In addition to this, floor cleaning cuts may be carried out before the machine engages in a new roof cut.

While these deviations from the idealised extraction cycle do complicate the detection of the individual extraction stages to some degree, they also yield quite valuable information that can be used, for example, for training the machine operators. Here an extraction-cycle detection routine can be used to quantify the operating profile of different machine operators – with the aid of known quantities of coal mined by the individual operators, resp. shifts.

Having access to a graphic image of the cutter-boom height, as presented in **Fig. 6**, serves to illustrate the differences that exist between the shifts, resp. crews, deployed on machine operating duties. The cuts taken by one crew of operators show very consistent maximum boom heights, while the data obtained for a second crew indicate a less regular performance. Having said that, the winning cycles completed by the 'irregular' group are on average shorter than those produced by the more consistent group, so that the mere awareness of these differences does not provide a clear evaluation of the situation and ultimately only confirms the correlation between machine operators and cutting behaviour,

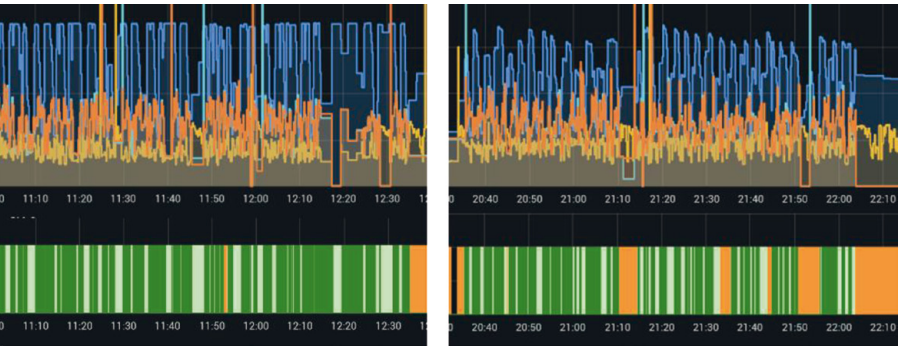

Fig. 6: Comparative visual data analysis of different groups of continuous miner operators – plot lines for the boom heights associated with uniform and non-uniform cutting

as these differences can be verified to be consistent over time as long as external conditions do not change.

The foremost objective of an extraction-cycle detection system, however, is to count the extraction cycles and in this way to conduct a performance evaluation of the continuous miner and quantify the volume of coal extracted, taking account of the known geometry of the cutter drums and the depth of cut. These aims can be achieved, in spite of any irregular cutting behaviour, provided that the commencement and end of the extraction cycle are correctly recognised and floor cleaning stages are not erroneously identified as new cutting cycles, as represented graphically in the stylised depiction in **Fig. 7**. Here the blue plot line depicts the position of the boom while the red line represents the cutter motor current. Two individual extraction cycles can be identified in this case. The movements taking place at the end of the second cycle are not incorrectly identified as the start of a new winning cycle. A rule-based classification model can be derived from the findings of this explorative data visualisation process and this will provide an accurate definition of the extraction cycles of the continuous miner machine. There are several different ways in which this can be used to obtain an approximated calculation of the volume of material extracted. For example, the number of extraction cycles identified over the course of a working shift can be multiplied by an empirically determined, average volume of material per cycle. Alternatively, or in addition, the volume of mineral extracted during each extraction cycle can be calculated using the cutting height, the depth of cut taken by the drum and the drum width. The current recorded at the cutter motor can also be used in order to remove any empty runs or non-productive cuts from the calculation.

As noted above, obtaining a precise, reliable and sound calculation of the key performance indicators, and particularly the number of cutting cycles and the tonnage, will ultimately depend on the approach taken in handling the many irregularities, the different operator styles and the local conditions. Here help can be provided in the form of data science algorithms and methods, which are first trained with real machine operating data in which ideally many of the aforementioned variations can already be observed. Using additional real data the recognition of extraction cycles and mineral volumes can be tested for general accuracy and undesirable local artefacts.

A high degree of accuracy in the identification of the performance indicators is particularly possible when the variations lead to different findings for the recognition of the performance parameters, such as the typical maximum and minimum boom heights, the typical cycle times and the typical horizontal machine travel paths during one cutting cycle. The collection and interpretation of 'big data' obtained under real conditions, i. e. the transmitted sensor data, play a key role here.

4 Data Architecture and IT Infrastructure Requirements below Ground

The operational deployment of data science algorithms and methods for the analytical treatment of data, as described in the example presented in the previous section, first requires the creation of infrastructural IT systems that are capable of storing and processing the sensor data and making it available for visualisation and/or further utilisation. For mining equipment manufacturers this presents the challenge of having to systematically integrate these algorithms and processes into an already existing and only part-integrated data infrastructure, or indeed of having to create a new one completely from scratch. For the purpose of this paper a data infrastructure is defined as the entire body of data handling systems that are used for the transmission, storage, processing and visualisation of data. Persistent data sinks, such as databases, are required for example to store the sensor data and these have to be populated

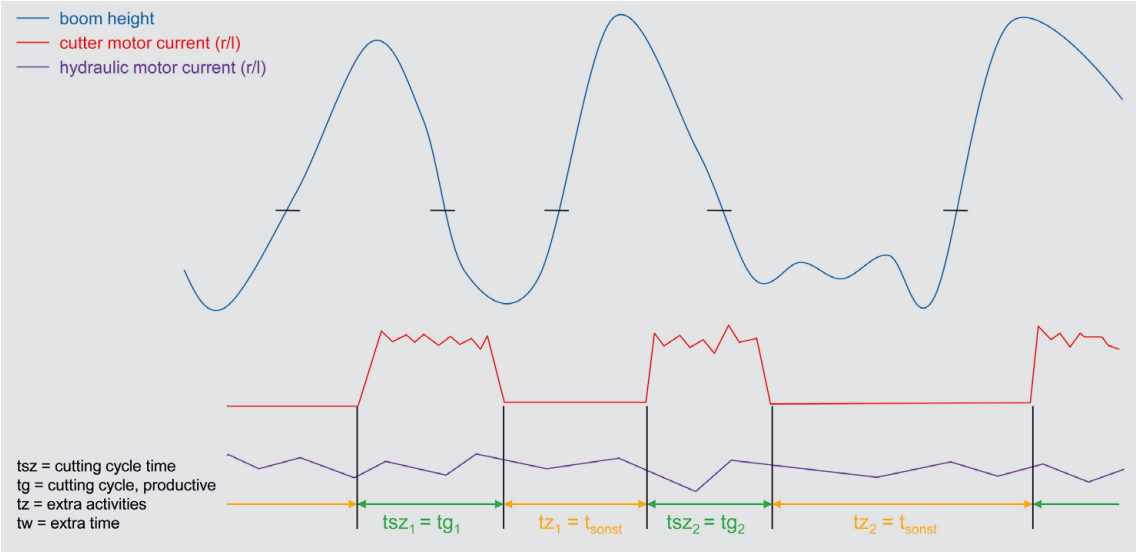

Fig. 7: Identification of winning cycles despite irregularities

into a coherent system landscape. Access to these data sinks should also be presented as unitary, fail-safe and responsive for downstream data processing and visualisation systems [6, p. 7 ff.]. For mining equipment manufacturers in particular it is vital to have intermediate buffering of the data stream as it 'flows' through various subsystems in the data infrastructure. This has to be seen against the background of highly latency-prone communication pathways between the on-board data source and any distant data storage and processing devices. Here the complexity of a data infrastructure system is very much measured and defined by the demands being placed on it. Depending on the circumstances and the requirements they impose a semi-integrative data infrastructure can in fact achieve pragmatic objectives, which may provide grounds for the use of data science processes.

This initial description of the requirements that have to be met by a data infrastructure always reveals the software-architectonic complexity that equipment manufacturers have to face in the context of a systematic integration of data science algorithms and methods, especially against the backdrop of 'big data'. The aim of this section is therefore to present an exemplary description of such a data infrastructure in consideration of the particular conditions that apply in the mining industry. With this in mind it is first necessary to explain the aforementioned term 'big data'.

4.1 What is 'Big Data'?

The term 'big data' refers to the huge rise in the volume of data in terms of the required storage resources, the generation, transmission and processing speeds involved and the inhomogeneity of the data formats [7, 8]. The increasing integration of sensors in machines means a continuous growth in the quantities of data to be processed [9, p. 1 ff]. The challenges involved are intensified by the costly realtime processing operations that have to be carried out on these substantial data streams, e. g. when using prognostic machine controls in support of operational processes. This situation is aggravated by the fact that these data processing routines usually cannot be undertaken by the hardware that is embedded in the machine itself. What is more, different sensor datasets can exhibit structural disparities ('inhomogeneity') when it comes to their (format) semantics and logical data type. A machine-overarching data infrastructure must have the characteristics needed to meet all the different requirements imposed by large, high-frequency, inhomogeneous data collections.

4.2 Characteristic Features of a big-data Data Infrastructure

The purpose of a data infrastructure is to perform arbitrary calculations on arbitrary datasets and to do this efficiently [6, pp. 27, 83]. The characteristics required of a big-data data infrastructure can be systematically derived and specified in the following context:

- Large volumes of data and hence considerable storage requirements
- High data generation, transmission and processing speeds
- Inhomogeneous data formats.

Big-data data infrastructures first try to process large volumes of data in an efficient manner. Here the speed at which the data can be read by an permanent memory plays a central role: a continuous miner fitted with five sensors, for example, generates a total of 1.73 million data points a day at a rate of 250 ms per data point and this has to be stored after it has been transferred into the data infrastructure. The terabyte-sized permanent memories in typical use today can now offer sufficiently large and convenient storage space for data volumes of this kind. However, the real problem for such large quantities of data has proved to be the read rate of the permanent memory. If this is about 600 MB/s, as is the case when rapid SSDs (solid state disks) are being used, it still takes about 28 minutes to read the entire contents of the permanent memory. This is not sufficient to meet the data infrastructure's expressed target of being able to perform arbitrary calculations on arbitrary volumes of data in an efficient manner. A possible solution here is to use a distributed system for the parallel processing of the data sets [9, p. 3 ff]. Here the entire body of data is distributed over several permanent memories that can be read from simultaneously. If a set of data amounting to 1 terabyte, in other words 1,000 billion bytes, is divided separately on to 25 SSDs each with a read rate of 600 MB/s, and if the amount of data on all the SSDs is read off simultaneously, it will only take about 1 minute and 6 seconds to read in the 1 terabyte of data.

However, a distributed system does not only consist of computers and data stores each with 25 SSDs but also requires an entity that holds and records which sets of data are held on which permanent memory. By the same token the body of data is not necessarily distributed over the permanent memories in a completely unconnected way but can, for reasons of failure safety, also be held on a redundant basis. If an permanent memory should fail the data sets it contains can still be found on other permanent storage points.

These connections and interdependencies between the individual system parts – the data storage computers on one hand and the computer nodes coordinating and logging the data distribution process on the other – represent the constitutive elements of a distributed data infrastructure. If more storage space is needed, or if the resilience to hardware and software crashes is to be enhanced, this can be achieved by using distributed data infrastructures: either by means of horizontal scaling-out, i. e. adding new computers to the network, or vertical scaling-up, i. e. upgrading the existing network, this usually focussing on the computer hardware system.

Big data also presents a data infrastructure with the challenge of having to achieve rapid handling speeds for data processing operations. One example of such a data processing operation is the use of the classification model (see Section 3 above) for handling distributed sensor data from one or several continuous miners. In practice, a MapReduce program or I/O-efficient data processing systems are mainly used for this purpose [9, p. 18f, p. 98; 10, p. 14f]. In the first case, namely the MapReduce approach, partial results are first filtered into subsets of the overall body of data and then subsequently merged together again to produce an overall outcome. The MapReduce option can also be described as parallel data processing. In the second case, namely the I/O-efficient approach adopted by Apache Spark, the distributed system attempts to carry out as many calculations as possible in a memory-uniform way using high-performance storage units such as processor registers, processor caches and random-access memory devices, with the aim of avoiding time-consuming permanent memory operations. Sufficiently rapid connection speeds can be achieved by employing the latest telecommunications technology and by the additional use of edge computing techniques, if required.

And finally, big data also compels the data infrastructure to support as many domain-relevant data formats and types as possible. The latter can broadly be divided into structured data formats (e. g. relational data sources), semi-structured data formats (e. g. XML or JSON) and unstructured data formats (e. g. video and audio files) [9, p. 5 f.]. In the given example of the continuous miner in particular it is useful to provide for storage over and above the time-series type of sensor data, for example by storing microphone, video and image data or even RADAR and LiDAR datasets.

The characteristic features of a big-data data infrastructure can be summed up as follows [6, p. 7 ff]:

- Robust against software and hardware corruption and fault tolerant of human error
- Low read latency of the data memory and ad hoc data requests
- Horizontal scalability of the data storage and processing hardware
- Generalisability of the data formats and types and easy expandability of the data processing environment
- Low maintenance effort required for the data storage systems through the use of inherently distributed system architectures
- Simplified troubleshooting and debugging through the separated storage of processed data and raw data.

The following sections will seek to explain how a big-data data infrastructure – such as that used for the classification model described in Section 2 – can be fitted out in this way and can be prepared for practical deployment at product and company level.

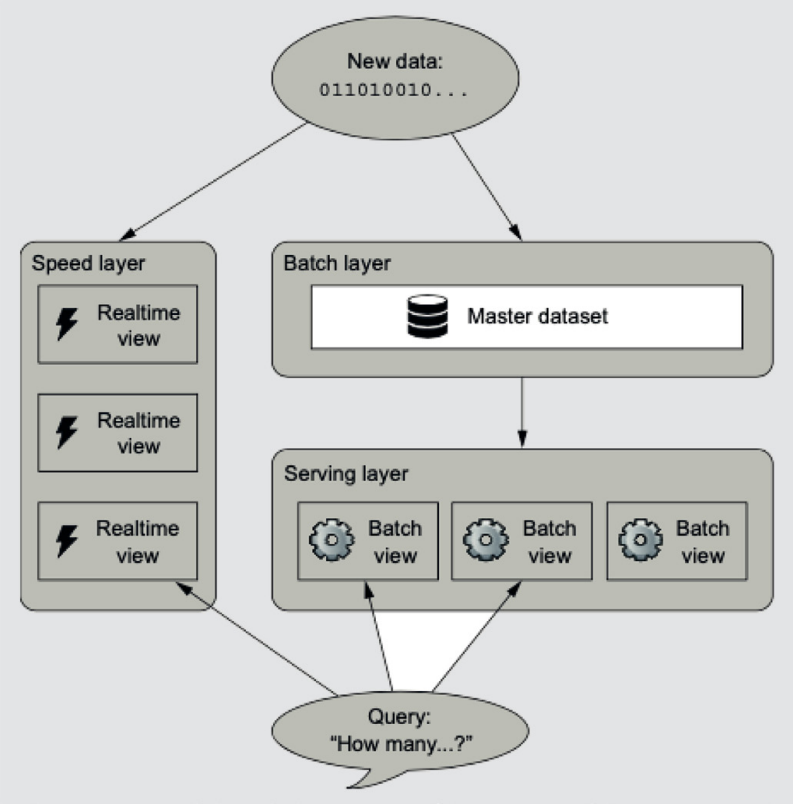

Fig. 8: Lambda architecture as layer model [6, p. 19]

4.3 Example of a big-data Data Infrastructure Concept with Lambda Architecture

A lambda architecture, which can be represented in software-topological terms as a layer model, can be used to integrate the various systems and tools needed for developing a data infrastructure into a complete system [6, p. 14]. Here a distinction is made between batch layer, serving layer and speed layer as the key system components (**Fig. 8**).

4.3.1 The Batch Layer

The batch layer of a lambda architecture stores and processes the data in fixed cycles. In the given example of the continuous miner the sensor data, after going through a buffered and mine-external transmission process, are first transferred to the batch layer of the lambda architecture and held in storage there. As outlined above, the sensor data are stored in a distributed system using several computers with permanent memory, possibly with redundancy grouping. This creates robustness against the software and hardware corruption of a data infrastructure and also provides for horizontal system scalability. One example of such a distributed system used as a batch layer for a big-data data infrastructure is the Apache Hadoop framework whose functionalities and features are called on below for the application of the lambda architecture.

As the continuous miner then continuously generates sensor data and sends this information to the

big-data data infrastructure increasing quantities of data will be compiled over time in the batch layer. If an arbitrary function, such as the extraction cycle detection process described in Section 3 or a machine learning model, is then executed on this body of data, the runtime of the calculation can be substantially reduced by deploying the MapReduce-based parallel processing program or an I/O-efficient processing routine operating in combination with suitable systems. However, this time reduction is often insufficient to produce a result in real time.

In order to be able to supply results to ad hoc data queries immediately the subcalculation or precalculation of the function is available in 'batch view', this combining with the storage of the calculation results that takes place in the serving layer [6, p. 15]. A batch view will, for example, contain the requested results from the extraction cycle detection regime, in other words the particular point in time when the continuous miner was at a certain stage in the winning cycle. Another batch view will contain the results for the number of extraction cycles per day or for the volume of mineral extracted by the continuous miner per minute. The actual sensor data held in the batch layer, which is also referred-to as the master dataset, remains unaffected when calculating a batch view. This represents a paradigm shift in relation to incremental architectures, as the batch views are recalculated cyclically, e. g. every three hours [6, p. 9 f, p. 88 ff]. One key aspect here is that the calculation logic, which prescribes the function for the batch view generation, is completely removed from the master dataset. If for example the function for extraction cycle detection has a software bug then it is only the batch view that will be defective, not the master dataset. The batch view can then be overwritten with the corrected program code in the next batch cycle. Here the interpretation of the format of the sensor data ideally takes place during the runtime of the data processing operation. This means for example that Codec-formatted sensor data, such as audio, graphic and video data, can be saved as binary files. By providing additional support for the storage of sensor data in pure text formats it is possible for a data infrastructure based on Apache Hadoop to support many different data formats at the same time.

4.3.2 The Serving Layer

The purpose of the serving layer is to supply the end user, e. g. a data visualisation system or an API, with the batch views calculated by the batch layer [6, p. 179]. The data in the batch views must be provided very quickly, that is to say without any high latency. The serving layer therefore uses index data structures to help speed up the process of retrieving certain data within a data memory. However, data storage in the master dataset of the batch layer does not necessarily use indices, as these can slow down the insert operations to the data memory. Running an ad hoc data query in the lambda architecture is therefore made possible by the generation of batch views in the batch layer and by the provision of batch views in the serving layer.

4.3.3 The Speed Layer

As the batch layer only carries out calculations on the master dataset cyclically, e. g. every three hours, it would not be possible, without an additional system component designed to supplement the batch and serving layers, to ensure that calculation results for ad hoc data queries will always be available for the most recent data. If the calculation cycle of the batch layer takes three hours, for example, then downstream systems such as data visualisation will in the worst-case scenario have to wait three hours for newly calculated data. For condition monitoring systems, or even real-time recommendation systems, this time span is simply too long in any case. That is why batch and serving layers are supplemented by a speed layer that performs incremental calculations on the data accruing during this intervening period.

4.4 Technologies and Frameworks for Building a Data Infrastructure

The lambda architecture as described above is just one of the potential architecture models that can be used for a big-data data infrastructure. If the results of the sensor data-based calculations are not time critical, in that they are not to be visualised in real time, the speed layer can be omitted entirely. On the other hand, if the results of the calculations are exclusively time critical in nature, e. g. because realtime condition monitoring is to be applied, then the batch and serving layers can be left out.

The lambda architecture therefore describes how these two architecture models are connected together. Apache Hadoop can specifically be used for the batch layer while, to complement this, Apache Spark can be employed for an I/O-efficient data processing routine. ElephantDB and Apache HBase are technology options for the serving layer, while Apache Storm and Apache Kafka, along with their relevant microservices as processing elements, can potentially be used for the speed layer.

The following now presents further examples of how different algorithms can be used for underground mining machinery, these operations being executed using the aforementioned lambda architecture of a big-data data infrastructure.

5 Example Applications from the Eickhoff Group

The data science set-up as described above, that is to say the entire body of data science algorithms and methods available here, can be built up to different configuration levels to meet the specific needs of the end users.

5.1 Configuration Stages for Data Science Infrastructure in the Mining Sector

The **first stage** often involves the visualisation of raw data using machine data dashboards. Depending on the application, commercially available solutions can be used as a visualisation front end for fixed or web-based activities. In the specific case of the Eickhoff Group, Microsoft PowerBI (**Fig. 9**) and Grafana (**Figs. 5 and 6**) are employed as web-based user environments for data visualisation. The purpose of raw-data visualisation is to determine key machine and process parameters and to present them in raw form and in chronological sequence so that discontinuities and deviations from norm can be quickly identified and analysed in detail. Of special relevance here is the dashboard, where fixed target values and averages from the past can be called upon as a benchmark – rather like the rev counter in a motor car. As far as underground mining machines such as shearer loaders and continuous miners are concerned this is especially useful for displaying motor current data, and hence for determining the load status and the productive usage of the machine, and for visualising error frequency rates and other general (production) parameters such as travel distance, runtime, etc. In addition to the basic set-up parameters the solutions available also allow for customised adaptations and evaluations.

As outlined in Section 4, a serving layer is a fundamental requisite for visualising raw data, this providing for direct, rather than just sequential access to the data in as responsive a manner as possible. Here it is particularly important to be mindful of the amount of data being depicted. If this volume exceeds many tens of thousands of individual data points, as is often the case when viewing sensor data acquired over long periods of time, the body of data involved will have an impact on the responsiveness available to the end user within the data visualisation system especially when zooming or panning. Here again the batch layer is able to pre-aggregate and present the body of raw data in various batch views with different depths of detail. For example a batch view can provide sensor-data averages calculated as data points over the course of one day, while another batch view can produce data points with per-second resolution. The latter is used when zooming into a small time range, while the former is used when zooming out in order to get an overview of a whole year's output without having to explicitly load and display millions of data points. In this way the raw data can be made available in different levels of detail and in a manner that is adapted to the detail setting selected for the visualisation. If realtime data are also to be visualised it is advisable to use data-driven updating of the visualisation elements. The raw data are actively transferred to the visualisation program as soon as they have been processed by the speed layer, rather than being supplied as batch views from which they can be loaded by the visualisation system.

The **next configuration stage** involves the visualisation of calculated values using classical statistical and

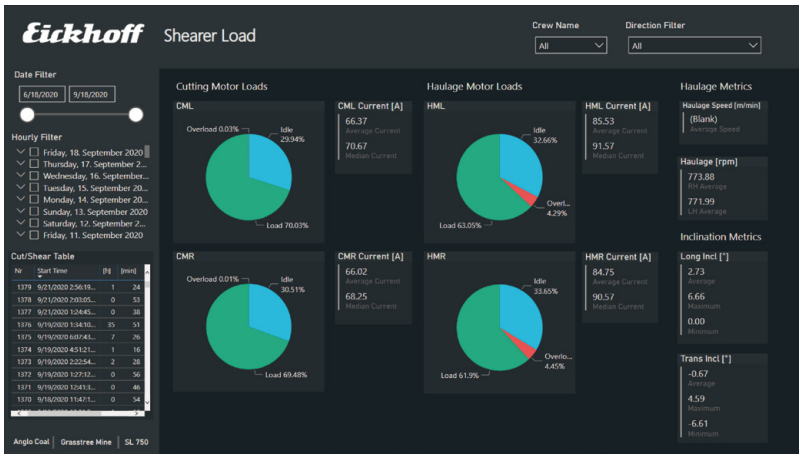

Fig. 9: Raw-data visualisation in a machine data dashboard for a shearer loader

manual rule-based models. The data infrastructure that processes the required calculated values and delivers them to the visualisation front end must, as outlined in Section 4, be capable of performing any calculation on the data. Raw-data time series from different sensors can be correlated, also visually, this point in order to reveal patterns and correlations and to help test pre-established assumptions. In this case the values calculated using the data infrastructure correspond to the findings of a correlation analysis. Root cause analyses, which seek to establish the relationship between certain conditions and the cause(s) of the error or defect, are also focused in this configuration stage. Likewise, the visualisations of analyses serve to select the right machine maintenance strategy on the basis of observed machine behaviour compared to reference benchmarks of the machine in the sense of a 'machine fingerprint'. In this context **Fig. 10** presents by way of example the working load at the cutter motors of coal shearer machines of identical type as recorded during operations at different collieries. While, as **Fig. 10** shows, the data for machine tracks (i. e. mines) 1 and 2 clearly indicate that the shearer in question is on a unidirectional cutting plan (because one of the two cutter motors is much more heavily loaded than the other), the data for mine 3 point to a bidirectional mode of operation. It is none the less clear to see that the loads acting on the left-hand motors (CML) of the shearers operating at mine 1 and 2, both generally as well as in terms of peak loading, are frequently greater than those for the corresponding right-hand motors (CMR) operating at the same mine. Furthermor, these figures also lie above the more constant load values for the two motors fitted to the shearers operating at mine 3. This is useful information for the servicing department as far as the supply of spare parts is concerned and will also be of interest to the operator of colliery number 3 (**Fig. 10**) in the light of the unused machine potential.

In keeping with the visualisation of the raw sensor data, the batch, serving and speed layers of a data infrastructure are also required for the visual display of calculated values. Batch and speed layers are responsible for

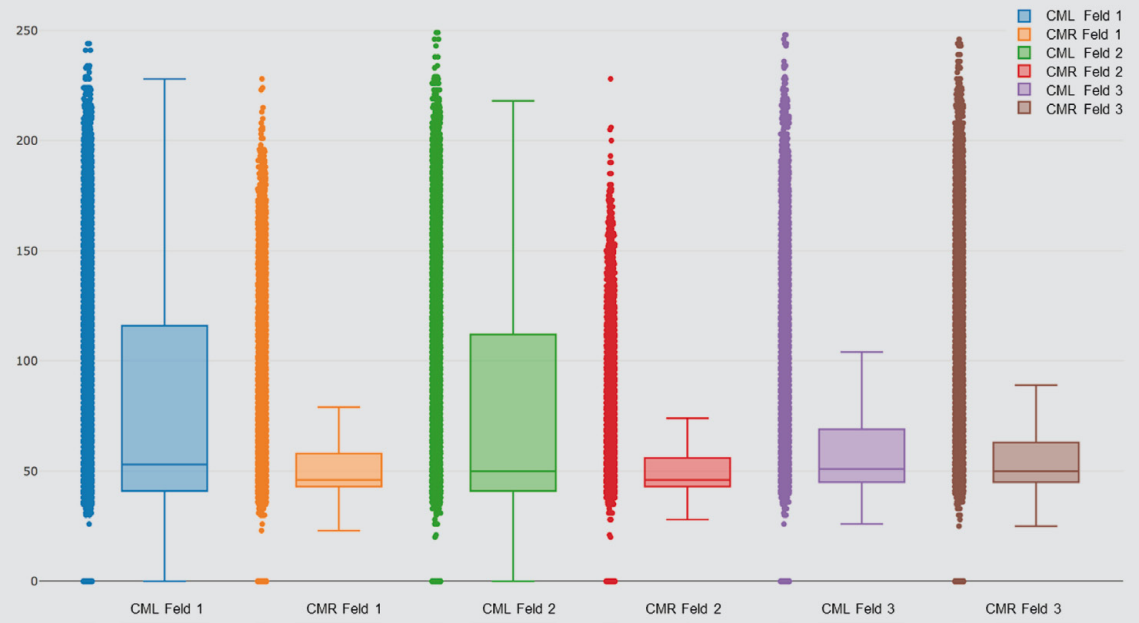

Fig. 10: Machine fingerprinting of shearer loader operations
Each boxplot represents the distribution of the respective motor current in the left (CML) and right (CMR) cutter motor in a specific extraction field of a mine.

performing the calculations on the raw data. The results of these calculations, such as the findings of a correlation analysis or a statistical hypothesis test, are always stored in batch views in the serving layer, so that adjacent systems (e. g. visualisation) can have responsive access to the resulting output.

A further configuration stage that can be applied to the data science set-up as described here involves the inclusion of exogenous, i. e. in this case machine-external, data. This will be described using the example of a continuous miner operating on a two-shift basis. Here the exogenous data consists of information relating to the shift allocation of the mine operator.

Selecting a larger time frame makes it possible to identify various performance metrics that can then be used to evaluate and consequently interpret persisting operation characteristics as well as trends. In Fig. 11 it is clear to see that even though the two teams of continuous miner operators are each executing a similar number of cuts per shift over the two-week period shown, team 2 (right) is generally raising the cutter boom higher and the standard variance of the cutting height as a measure of the irregularity of the cutting action is much lower than for team 1 (Fig. 6). As far as the extraction volume is concerned team 2 were seen to have performed better by nearly 10 %, this also being reflected in the OEE diagram (middle) for this timeframe. The total accumulated productive times (light and dark green), and especially the cutting time (dark green), recorded by team 2 are significantly higher than those registered by team 1.

Fig. 11: Analysis of the quantities of coal produced by different groups of continuous miner operators

The identification, and more particularly the evidential proof of such disparities can act as valuable starting points for mine operators in their efforts to apply successful operating strategies across the board and exploit the productivity potential to the full. However, these findings are also extremely relevant for the equipment manufacturer: they enable him to optimise the machine's functions by applying operating parameters and constraints based on actual usage conditions and to organise appropriate operator training for this and for the relevant automation routines.

The visualisation of calculated values based on classical statistical and manual rule-based models also provides the basis on which machine learning models (ML models) can be built and deployed for particular applications. One of the benefits of the ML approach is that the complex know-how built up by the machine operator can be mapped algorithmically and this can contribute towards developing intelligent automation, for example by using regression and classification models. Another is that clustering models can be used, for instance, to identify complex cause-effect relationships with multiple variables without the need for continu-

ous and precise knowledge of these influencing factors for all possible scenarios. The two examples presented below relate to the first of these applications.

5.2 ML Example – Control and Optimisation of the Machine Winning and Travel Paths

ML models can be used for example to control and optimise the travel paths of a shearer loader within the longwall system and can make a contribution to better process automation. In this example the research focuses on predicting the longitudinal gradient of the face system in order to support the automation of the ranging drum arm – and in particular to help the machine adapt to changes in face gradient. In real terms this can lead to the development of proposals on machine steering and to the implementation of machine control systems.

Based on the first 40 cuts taken by the shearer loader the ML model is able to learn the gradient position of the face – this is illustrated in the heat maps shown in **Fig. 12** that represent the positive and negative longitudinal tilt of the machine over the length of the face. In visual terms the model learns that a ripple in the coal seam will be cut earlier or later in each following pass. In accordance with the learned pattern it will then be possible to predict the slope angles of the next three to five cuts with a high degree of precision.

As soon as sensor data are available for the longitudinal tilt of the shearer loader learning ML models can be developed on the basis of this information. The recorded data for longitudinal tilt and other variables (shearer position on the face, motor currents, etc.) are modelled on the target variables in the course of the learning step, which in this case means the longitudinal tilt values for the next five cuts. The role of the neural network is to approximate the mapping instruction so that after the learning stage the neural network is in a position to calculate the target variable by itself. In this specific case various architectures of neural networks were tested and validated, including pure feedforward networks and others with time-series optimisation. The ultimate result is a pure feedforward network (**Fig. 13**) that can predict the longitudinal tilt values of up to five future cuts better than the reference metrics of constant tilt settings.

As is also the case with the calculation of classical statistical and manual rule-based models the ML model as described here can be computed in the batch and speed layers of a big-data data infrastructure as outlined in Section 4.

5.3 ML Example – Compensation for Sensor Failure during Cutting Cycle Identification

The second concrete application is aimed at the reliable identification of the cutting cycle, and hence at es-

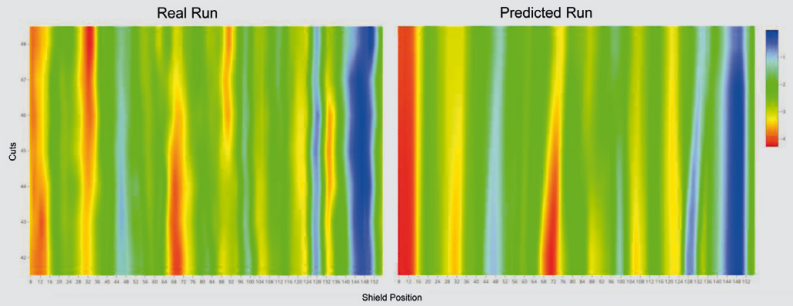

Fig. 12: Actual and predicted longitudinal tilt of a shearer loader over the length of the face

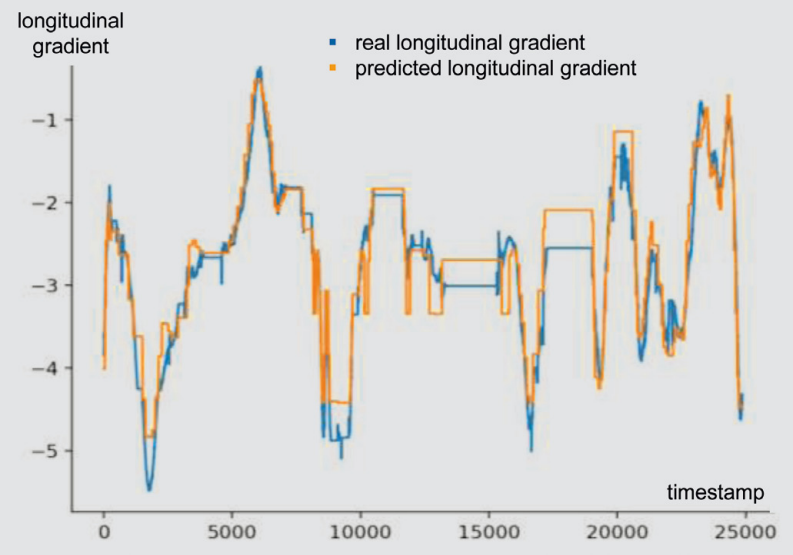

Fig. 13: Deep-learning prediction with MLP-FF regressor grid of 8 layers and 86,785 parameters

tablishing the amount of mineral extracted by the continuous miner, despite a failure of the sensor system. The volume of material extracted correlates strongly with the number of cutting cycles completed, except for variances in the height of the cut. As described earlier in Section 3, the determining factor for extraction cycle identification is the change in the height of the cutter boom. However, the failure of the relevant sensor, or rather the non-transmission of the recorded data, often has to be compensated for – not least because of the extremely inhospitable conditions that can occasionally prevail in room and pillar mining and the intense strain that this can place on the machinery. Such malfunctions can be neutralised by using an alternative method for gauging the height of the boom: this can be done directly using the amount of travel at the cylinder or indirectly based on the failsafe measurements of the hydraulic flow at the actuating valves (**Fig. 14**).

While these physical correlations are obvious, they are however very difficult to map into formula form. There are too many influencing factors to take into account – from frictional resistance and superimposed dynamic loading through to temperature levels. Further-

Fig. 14: Real and predicted height data for a continuous miner machine

more, these influencing variables are not constant for all application scenarios but can vary enormously depending on the extraction panel. The solution is to generate an ML model that is learned from a recorded time span, e.g a month, in which the height sensor functioned. This means that when no height data are available the data from the ML model can be used in order to replace the gaps in the data transmission. In **Fig. 13** the blue-coloured 'real data' can be seen in place up until about 10:10 hrs, backed by the lilac-coloured prediction data that then continue to estimate the height of the cutter boom and in so doing ensure an interruption-free detection of the cutting cycle. As in the preceding application, the ML model was trained with different machine parameters. The purpose of the model is to estimate the height of the cutter boom based on these machine parameters. As **Fig. 14** shows this ML tool has proved its effectiveness in successfully determining the height of the machine boom.

6 Conclusions and Outlook

While the industry has not yet reached the end of the road as far as digitisation is concerned, the applications presented here are striking examples of the potential that data science technology now has to offer the mining sector – and this will be of benefit to both users and manufacturers alike. Companies that are successfully taking the heuristic approach in using lean methodology, for example, and so are nearing the limits of optimisation, will in particular have access to a whole range of options for identifying and quantifying process deviations and abnormalities and this will facilitate changes and improvements using established practices from the world of lean technology. Hard evidence can be provided for these best practices and the potential of behavioural changes and planned measures can be determined in advance and then monitored during implementation. However, while the use of explorative visualisation and manual rule-based modelling is improving the conditions are also becoming better for the application of machine learning systems in real-life situations. There is valuable support to be had in a number of ways. Deviations from normal can for example be detected using unsupervised learning models without the prior learning of 'bad cases' and the failure of sensor data can also be compensated for by having an alternative body of correlating data. New business models based on availability and output can therefore be developed in a much more robust manner than before. While there is a huge potential on offer here, this is not being properly exploited at the present time. However, more and more operators and manufacturers are beginning to think in these terms and are starting to avail themselves of the opportunities that this technology can provide. For individual companies this also means building up competencies in the application of data science algorithms and methods and at the same time further developing and expanding their data infrastructure systems. The ultimate objective must be to embed data science firmly into the company's structure and thinking – along the lines of the set methods that have been adopted from the world of lean production, such as CIP (continuous improvement process) and FMEA (failure mode and effects analysis). Moreover, the synergy that comes from growing competence and acceptance has a self-amplifying effect – not least through the adaptation and enhancement of existing business models.

7 References

[1] Bartnitzki, T. (2016): OPC UA - Ein Kommunikationsstandard für Bergbau 4.0. In: Ta-gungsband zum AKIDA 2016 / Nienhaus, Karl (editor) (pp. 37-48). Institut für Maschinentechnik der Rohstoffindustrie.

[2] Suciu, M.; Kowitz, S. (2019): Digitizing Raw Material Mining – End-to-End Integration into an IIoT Platform for the Analysis of Machine Data. In: Mining Report Glückauf 155 (2019) No. 4.

[3] Drüppel, E. (2010): Entwicklung eines Konzeptes für die schneidende Gewinnung im Steinsalz. Hochschulbibliothek der Rheinisch-Westfälischen Technischen Hochschule Aachen.

[4] Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P. (1996): From Data Mining to Knowledge Discovery in Databases. In: AI Magazine, 17(3), 37.

[5] Provost, F.; Fawcett, T. (2017): Data Science für Unternehmen. mitp Verlag, Ferchen. ISBN: 978-3-95845-546-7.

[6] Marz, N.; Warren, J. (2015): Big data: principles and best practices of scalable real-time data systems. Manning Verlag, Shelter Island, New York, USA.

[7] Diebold, F. X. (2020): On the Origin(s) of the Term "Big Data". In: arXiv e-prints, arXiv:2008.05835.

[8] Laney, D. (2001). 3D Data Management: Controlling Data Volume, Velocity, and Variety. Technical Report, META Group.

[9] White, T. (2015). Hadoop: The Definitive Guide. 2. Auflage. O'Reilly Media, Sebasto-pol, Kalifornien, USA. ISBN: 978-1-49190-163-2.

[10] Chambers, B.; Zaharia, M. (2018): Spark: The Definitive Guide: Big Data Processing Made Simple. 1. Auflage. O'Reilly Media, Sebastopol, Kalifornien, USA. ISBN: 9781491912218.

Prof. Dr.-Ing. Dipl.-Wirt. Ing. Andreas Merchiers

graduated in mechanical engineering and in business economics. In his professional career he has consistently focused on the challenges facing manufacturing companies, these including the use of lean concepts along the value chain, production and logistics structures and Industry 4.0 applications in the engineering and mining sectors. Prof. Merchiers completed his academic education at RWTH Aachen with a doctorate in engineering and is currently professor at the Bochum University of Applied Sciences. He is engaged in teaching and researching production management/Industry 4.0 and technical investment planning and also acts as an industrial consultant.

Contact: andreas.merchiers@hs-bochum.de

Prof. Dr. rer. nat. Henrik Blunck

is Professor of Applied Computer Sciences at Bochum University of Applied Sciences. After graduating in mathematics he obtained a doctorate in computer science at the University of Münster. His area of research includes data science and big data analytics, with a focus on domains including mobility data, context-aware consumer software and Industry 4.0 applications. These fields of study cover topics ranging from data acquisition and algorithm development through to real-world and market-viable products and innovations in emerging societal and business sectors. He also acts as an industrial consultant.

Contact: henrik.blunck@hs-bochum.de

Arne Köller, M.Sc.,

studied computer science, with a focus on intelligent algorithms, at Bochum University of Applied Sciences. His scientific and practical studies themed on machine autonomy and sensor processes in the raw materials industry. After graduating he was recruited by Eickhoff Bergbautechnik and worked in the Data Engineering section as well as in other departments. He is currently on the scientific staff working on Machine Autonomy and Communications Technology in the Raw Materials Industry at the RWTH Aachen University Institute for Advanced Mining Technologies.

Contact: akoeller@amt.rwth-aachen.de

Dr.-Ing. Christian Gierga

studied mining engineering at RWTH Aachen University and then obtained a doctorate at the Institute for Mineral Processing. His professional career then took him to HAZEMAG & EPR (Dülmen) in 2000, where he worked in the project development team before transferring to the sales department, subsequently becoming head of sales and marketing in 2004. In 2007 he joined Eickhoff Bergbautechnik, where he is now head of Service and Production.

Contact: kontakt@eickhoff-bochum.de