Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 15 (2014) 153 – 158

www.elsevier.com/locate/procedia

21st CIRP Conference on Life Cycle Engineering

Predictive analytics model for power consumption in manufacturing

Seung-Jun Shin[a,*], Jungyub Woo[a], Sudarsan Rachuri[a]

[a]*National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, 20899, The United States of America*

* Corresponding author. Tel.: +1-301-975-5994; fax: +1-301-258-9749. *E-mail address:* seungjun.shin@nist.gov

Abstract

A Smart Manufacturing (SM) system should be capable of handling high volume data, processing high velocity data and manipulating high variety data. Big data analytics can enable timely and accurate insights using machine learning and predictive analytics to make better decisions. The objective of this paper is to present big data analytics modeling in the metal cutting industry. This paper includes: 1) identification of manufacturing data to be analyzed, 2) design of a functional architecture for deriving analytic models, and 3) design of an analytic model to predict a sustainability performance especially power consumption, using the big data infrastructure. A prototype system has been developed for this proof-of-concept, using open platform solutions including MapReduce, Hadoop Distributed File System (HDFS), and a machine-learning tool. To derive a cause-effect relationship of the analytic model, STEP-NC (a standard that enables the exchange of design-to-manufacturing data, especially machining) plan data and MTConnect machine monitoring data are used for a cause factor and an effect factor, respectively.

Selection and peer-review under responsibility of the International Scientific Committee of the 21st CIRP Conference on Life Cycle Engineering in the person of the Conference Chair Prof. Terje K. Lien

Keywords: Sustainable manufacturing; Sustainability indicator; Big data; STEP-NC; MTConnect, Machine learning

1. Introduction

Intense global competition, uncertainties in energy cost and supply and exponential growth in information technology are shifting industries toward agile, high-performance and sustainable (resource efficient) manufacturing. To cope with broader performance objectives that include agility, asset utilization and sustainability issues, manufacturers have adopted Smart Manufacturing (SM), defined as the intensified application of advanced intelligence systems to enable rapid manufacturing, dynamic response and real-time optimization of production [1].

Data analytics, i.e., the science of examining raw data with the purpose of drawing conclusions about that data, is a key enabler for implementing SM [2]. Exploratory solution searching utilizing analytics can facilitate diagnosis, optimization, and prognostics for known and even unknown problems. In SM, big data infrastructure is needed to conduct the desired analytics due to the huge amount of data created on shop floors (volume), the various types of structured and unstructured data formats (variety), and the need for fast responses to enable on-time decision making (velocity). Data analytics utilizing big data infrastructure, called big data analytics, enables timely and accurate insights to make better decisions.

The objective of this paper is to present big data analytics modeling in the metal cutting industry. This paper includes: 1) identification of manufacturing data to be analyzed, 2) design of a functional architecture for deriving analytic models, and 3) design of an analytic model to predict a sustainability performance especially power consumption, using the big data infrastructure. A prototype system has been developed for this proof-of-concept, using open platform solutions including MapReduce, Hadoop Distributed File System (HDFS), and a machine learning tool (easy Neurons). To derive a cause-effect relationship of the analytic model, STEP-NC plan data [3] and MTConnect machine monitoring data [4] are used for a cause factor and an effect factor, respectively.

The paper is organized as follows: Section 2 gives the problem definition. Section 3 proposes a functional architecture, and Section 4 presents analytic modeling.

2212-8271 Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and peer-review under responsibility of the International Scientific Committee of the 21st CIRP Conference on Life Cycle Engineering in the person of the Conference Chair Prof. Terje K. Lien
doi:10.1016/j.procir.2014.06.036

Section 5 shows a prototype implementation of a case study, and Section 6 concludes the paper.

2. Problem definition

This section describes the problem and the manufacturing data to be analyzed.

It is usually difficult to predict manufacturing performances, because it takes a lot of times to gather manufacturing data and to derive reliable models from the data. This is especially true for machining process. When deriving the models, either theoretical or empirical methods are commonly used. A theoretical method is based on the metal cutting mechanics. Meanwhile, an empirical method can reflect more practical phenomena, because the method is accompanied with an actual experiment and operation. Using either theoretical or empirical models has some problems. For example, the theoretical result sometimes does not reflect the real manufacturing process. In the case of empirical model, even if a model is derived at a certain machining condition, it is only applicable for the same condition. To solve these problems, we adopt a data-driven analytic modeling approach based on feature vectors which are n-dimensional vectors of numerical or nominal features that classify a machining operation. This approach is a useful method to derive reliable prediction models by finding an applicable data pattern. Furthermore, this approach facilitates to classify the analytic models efficiently.

To apply this analytics modeling, manufacturing data to be analyzed should be identified. Rich planning (what-to-make and how-to-make) and monitoring (machining action and performance) data are primary requisites in machining process. They can be cause (planning) and effect (monitoring) factors in analytics modeling, because the planning obviously influences the monitoring. In other words, machine-monitoring data depends on process plan data [5] [6] [7].

In this paper, STEP-NC is chosen to be the planning data, because its formalized data model provides comprehensive contents about the planning. Meanwhile, MTConnect is chosen to be the delivery and monitoring data through a common language and structure for utilizing the data extracted from the machinery. STEP-NC and MTConnect are briefly explained below:

- STEP-NC: STEP-NC formalized as ISO14649 has been developed to represent a common standard specially aimed at NC programming. STEP-NC specifies machining processes in an object-oriented manner via the concept of a workingstep [3]. Each workingstep includes a ‘manufacturing feature’ (what-to-make) and a ‘machining operation’ (how-to-make). The machining operation is associated with ‘machining strategy’, ‘technology’, ‘machine function’ and ‘cutting tool’ [3].
- MTConnect: MTConnect enables interoperability by allowing access to manufacturing data using standardized interfaces for web technology. Its XML structure defines information models as a set of their constituent axes, spindle, program, and controller [4]. A ‘sample’ is the value of a continuous data item at a point in time. An ‘event’ describes an asynchronous change in state. A ‘condition’ communicates the device’s health [4].

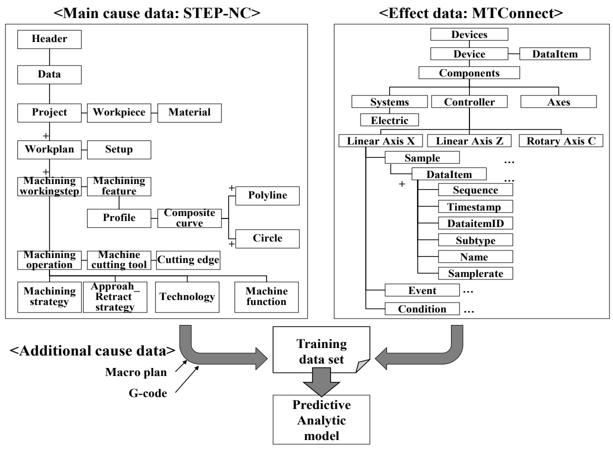

Fig. 1. Structures of cause-effect data and problem definition.

Fig. 1 presents the high-level structures of STEP-NC and MTConnect data, and the abstraction of the problem. The contents of the STEP-NC structure influence each component’s sampling data items of MTConnect, such as direction, position, force, velocity and wattage (power). These heterogeneous data are accumulated into a training data set, together with inputs of the macro plan and G-code data. The prepared training data set is used for analytic modeling. In summary, the purpose of this paper is to describe how to derive and organize predictive analytics models for power consumption from STEP-NC and MTConnect training data set.

3. Functional architecture

To solve the problem stated in Section 2, its operational mechanism should be defined. This section proposes a functional flow in addition to its primary data input and output, as shown in Fig. 2.

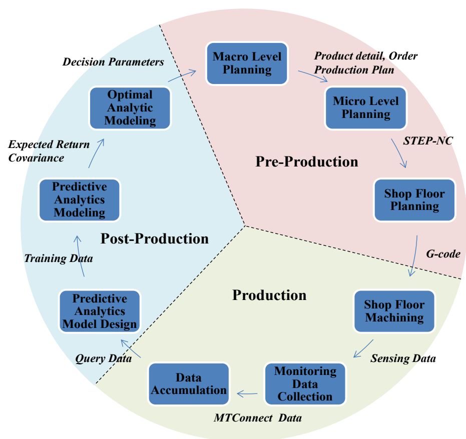

Fig. 2. A functional flow.

- Macro level planning: makes a periodic macro plan that assigns yield, manufactured products (design and material differences) and their machine tool allocation.
- Micro level planning: establishes a specific micro plan for each product, based on the macro plan. The design and material differences influence operation sequences and process parameter sets. This function outputs separate STEP-NC files in terms of the product design and material.
- Shop floor planning: generates machine-executable format (e.g., G-code) from the planning information.
- Shop floor machining: executes practical machining, following the given parameters. Sensor systems attached in a machine tool measure movement of driver sub-systems, and transfer the sensor signals to the machine controller.
- Monitoring data collection: extracts sensing data gathered in the controller and then transmits to the agent that formalizes the data in a standardized way.
- Data accumulation: stores the standardized monitoring data and all of the planning data from previous stages. This paired data set is used for making cause-effect relationships.
- Predictive analytic model design: retrieves training data set from the accumulated raw data according to the cause-effect relations. The time-series data is expressed using descriptive statistics to remove data redundancy. The mechanism of this function is described in Section 4.1.
- Predictive analytic modeling: carries out analytic modeling, using the training data set. (Detail in Section 4.2)
- Optimal analytic modeling: optimizes a performance (power consumption) at given plan (process parameters) [8]. These optimal parameters and their performances are fed forward to the planning phase. This function is out-of-scope of this paper.

4. Analytic modeling

A training data set should be prepared for machine-learning approached analytic modeling. It is important to note that the training set can greatly affect the reliability of an analytic model [9]. This section gives an overview of how we prepare the training data set and how we derive an analytic model in our prototype.

4.1. Preparation of training data set

To figure out a meaningful cause-effect relationship, cause factors extracted from input data and effect factors from output data should be clarified. Our approach requires a categorical representation of machining operations, since such representations facilitate classification of the machining operations. Therefore, the first step in our analytic modeling is to identify feature vectors (FV).

Table 1 presents a list of FVs used to influence the machine monitoring data. The adoption of a 'machine tool model' comes from its unique relevance to machining performance [5]. The types of cutting tools and their inserts impact the machinability [6]. The combination of cutting tools and workpiece materials necessitates different properties in process planning [7]. Machining operation and the use of a cooling system also impacts machining performance [10]. G-code instructions such as rapid positioning (G00), linear (G01) and circular interpolation (G02 and G03) generate different performance patterns. For example, cutting power is added to steady spindle and axes powers and basic power in actual cutting caused during the linear interpolation.

Table 1. Feature vector list.

Feature vector	Category	Example
Machine tool model	Machine tool	Puma 8HC, Integrex e-420H
Cutting tool type	Cutting tool	Turning, Grooving, End mill
Insert material	Cutting tool	Titanium coated, High speed steel
Workpiece material	Workpiece	Aluminum, Brass, Steel, Titanium
Machining operation	Process plan	Contouring, Facing, Rough/Finish
Cooling type	Process plan	Emulsion, Mineral Oil, Dry
Code instruction	G-code	G00, G01, G02, G03

The next step is to extract FVs and input parameters for machine learning from the manufacturing data. Regarding FVs, Fig. 3 shows how to directly extract FVs from a STEP-NC part program. As STEP-NC is capable of representing workpiece, machining operation, cooling type, cutting tool type and insert material, the five FVs (red-boxed) can be extracted from its part program. However, the current scheme of STEP-NC excludes both machine tool specification and G-code instruction. As an alternative, 'machine tool model' is assumed to obtain from an external macro plan. The 'code instruction' can be acquired from a G-code program generated by post-processing of the STEP-NC program. In this example, a single FV set is *'Machine tool model: PUMA 8HC-3A'*, *'Workpiece material: Steel'*, *'Machining operation: Contouring Rough'*, *'Cooling type: On'*, *'Cutting tool type: General Turning Tool'*, *'Insert material: TIN'*, *'G-code instruction: G01'*.

Based on this FV combination, input parameters should be determined for machine learning. The input parameters are used for making actual relationship with a learning output parameter. Three main process parameters – *cutting depth*, *feedrate*, *spindle speed* – and *cutting diameter* (blue-boxed) are chosen as the input parameters. Because *cutting diameter* (the diameter machined at an actual cutting movement) influences material removal rate and cutting power, *cutting power* should also be considered as a learning input. The three main parameters can be extracted from the STEP-NC part program, as shown in Fig. 3. Meanwhile, *cutting diameter* should be acquired from MTConnect, because it is a changeable parameter during a machining operation. Fig. 4 illustrates an example of MTConnect sample streaming data at an identical timestamp. *Cutting diameter* can be inferred from axis position, considering axis moving direction.

The final step is to extract a learning output parameter. In Fig. 4, each streaming data point accompanies 'name', 'data item', 'sequence', 'timestamp' and so on. 'Name' is a tag of 'data item' this sample is associated with. 'Data item' is the corresponding data retrieved in the probe request, 'sequence' is the sequence number of this event and 'timestamp' is the time the sample value is reported. 'Position' and 'wattage' are the representation of an axis position and a power use. The real numbers, called CDATA, indicate measured value at the timestamp [4]. Consequently, a total power relevant to the learning output parameter can be extracted by aggregating component's 'wattage' items (gray-boxed).

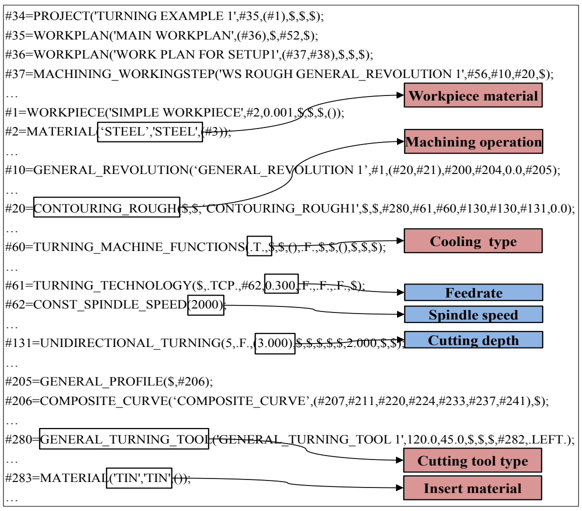

Fig. 3. An example or feature vector extraction.

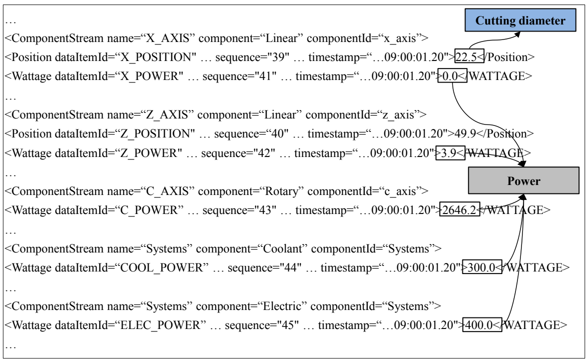

Fig. 4. An example of learn output (power) extraction.

Table 2 shows an example of a prepared training data set as an input of analytic modeling. For example, Case 1 is the training data obtained from Fig. 3 and Fig. 4. 'G01' only considers an actual cutting state which separates air cutting caused by overcut length. This separation is possible, when considering collision of axis position and workpiece geometry.

MTConnect XML files can have a huge amount of streaming data contents. In our simulation, a file size of an MTConnect file, which only includes 'X position', 'Z position' and 'wattage' sampled with 0.1 sec interval for one part is around 800 KB. According to the increase in the number of machine tools and yields, a shop floor obviously generates a huge amount data to be shown in Section 5. This large data volume exceeds performances of conventional database systems, when concurrently considering variety and faster derivation of the models. This is the reason the concept of a distributed database such as HDFS should be applied to overcome the limitation of the conventional systems.

Table 2. An example of training data set.

Type	Attribute Name	Case 1	Case 2	Case 3
FV	Machine tool model	Puma 8HC-3A	Puma 8HC-3A	Puma 8HC-3A
	Workpiece material	Steel	Aluminum	Titanium
	Cutting tool type	Turning	Turning	Turning
	Insert material	TIN	TIN	TIN
	Machining operation	Contouring	Contouring	Contouring
	Coolant	On	On	On
	Code instruction	G01	G01	G01
Learn input	Cutting depth(mm)	3.0	3.8	1.8
	Feedrate(mm/rev)	0.3	0.35	0.30
	Spindle speed(RPM)	2000	1500	750
	Cutting diameter(mm)	42.0	46.2	48.2
Learn output	Wattage(W)	3350	3495	3770

4.2. Model structuring

In analytics, model structuring is a significant aspect, because the structure determines how efficiently a group of analytic models is segmented and classified. Traditional analytic models are applicable only for targeted shop floor environments, because the models are not clustered in terms of FVs. In other words, the traditional models encounter model isolation, similar to the physical isolation of the shop floor. To resolve the isolation problem, the model structure should comply with the concept of unit analytic models (common building blocks classified by the FV set) and a composite analytic model (the aggregation of the building blocks), as shown in Fig. 5. For example, the three cases of Table 2 are used for three different unit analytic models because their workpiece material FV has different types. The model structure clustered by n-dimensional FVs ensures 'commonality', 'reusability' and 'composability'.

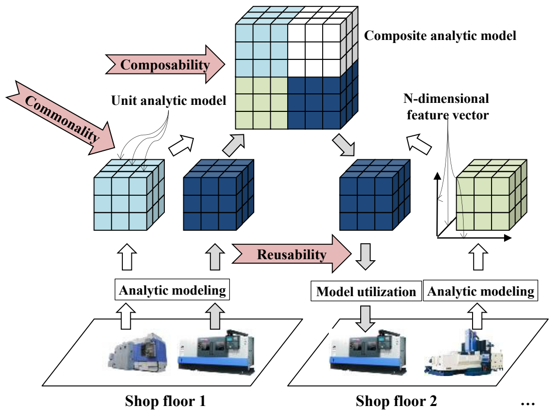

Fig. 5. A concept of unit and composite analytic modeling.

Through standardization, 'commonality' enables to register any unit models regardless of shop floor environments in which the models are derived. 'Reusability' implies the registered unit model in a shop floor can be reused in another shop floor in cases where the latter shop floor uses the identical or similar FV set. According as the unit models increasingly include more FV combinations, 'composability' allows the expansion of the composite analytic model. Finally, the composite model possibly covers all analytic models scoped in the FVs.

4.3. Predictive analytic modeling

The training data is an aggregation of a pair constituting a learning input set and a desired output value as mentioned in Section 4.1. Thus, this machine-learned modeling should be approached by supervised learning, which analyzes the training pair sets and infers a function. To develop the unit analytic model, we select a back propagation neural networks algorithm for the learning, because the algorithm has strength for figuring out the complex relationship due to many learning input sets. Fig. 6 shows a structure of a neural network in an FV set. w_{oj} and w_{ji} denote the synaptic weights from neuron j in the hidden layer to the single output neuron. f_o and f_j stand for activation functions of the neurons from the output layer and hidden layer, respectively.

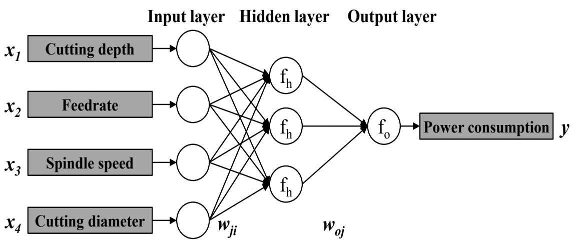

Fig. 6. A structure of neural network.

5. Implementation and case study

To prove our concept, we are implementing a prototype system using open platform solutions including MapReduce, HDFS, and a supervised machine learning tool (easy Neurons).

5.1. Big data infrastructure

The system consists of three sub systems: a data collecting system, HDFS, and a machine learning expert system. Table 3 and 4 show the implementation environment and specification. The data collection system is tightly coupled with the shop floor system to continuously gather all kinds of data during production time. The data includes the manufacturing data coded by the external macro plan mentioned in Section 4.1, STEP-NC part program and MTConnect runtime data. The MTConnect data is artificially generated by our in-housed simulator that makes tool movements and power by a theoretical mechanism [11]. The data collection system is based on Java Server Faces (JSF) web application for a user to control and monitor the data collection.

HDFS contains and manages big data from the data collection system. HDFS stores a file system metadata on a dedicated server, called the NameNode. HDFS also stores application data on other servers, called DataNodes. All servers are fully connected and communicate with each other using TCP-based protocols. For our case study, we have one NameNode and three DataNodes. Table 5 shows the performance of HDFS, when data sizes are assumedly accumulated on DataNodes. 'Time' stands for the purely reading time of the given data size.

Table 3. Implementation environment.

CPU	OS	JDK	Storage	RAM	Network
4 quad core Xeon @ 2.5ghz	Ubuntu Linux Server Release7.0	Sun Java JDK 1.6.0	4 directly attached SATA	16G RAM	1 gigabit Ethernet

Table 4. Prototype system specification.

System	Toolbox	Specification
Data Collecting system	Primeface, JTL, JSF	Web application, runtime module
HDFS	Hadoop, Mapreduce, Hive	1 NameNode and 3 DataNode, Apache generic setting
Machine Learning system	Easy Neurons	4 input neurons, 3 hidden neurons, 1 output neuron

Table 5. Performance of HDFS.

Bytes (TB)	Nodes (#)	Maps (#)	Reduces (#)	Time (sec)	Agg. Nodes IO speed (GB/s)	Per Node IO speed (GB/s)
1	1452	8121	2700	62	31	22.4
2	1560	10120	3600	112	32	18.2

5.2. Case study

The case study aims at making predictive models by the neural network illustrated in Fig. 6 for turning machining. Three unit analytic models are derived in terms of a workpiece material (Steel, Aluminum and Titanium). The models assumedly have the same FVs except the material (see Table 2). Table 6 shows the range of three learning input parameters. The cutting diameter is given the same value. 10000 learning input samples for each material are randomly generated and their relevant outputs are gathered. The set of the samples is used as a training data set. The machine learning system generates three predictive models for each material type from the training set, using easy Neurons. The model parameters are learned in batch mode.

Table 6. Ranges of learning input parameters.

Material	Feedrate	Spindle speed	Cutting depth	Cutting diameter
	mm/rev	RPM	mm	mm
Steel	0.2~0.6	900~1200	2~6	44~48
Aluminum	0.1~0.5	900~2000	1~4	46~49
Titanium	0.2~0.4	500~1000	1~3	47~49

To measure the model performance, test samples are used to get prediction results from the machine-learned model. 1000 samples for each material are randomly generated within the cutting condition (Table 6). Fig. 7 presents a scatter plot of theoretical powers versus machine-learned powers. Our observation and result are described below:

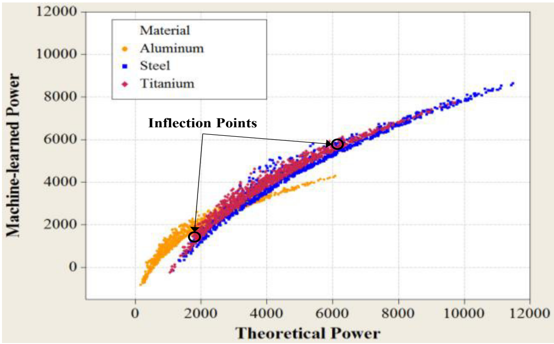

Fig. 7. A scatter plot of theoretical powers vs. machine-learned powers.

- Observation: the three scatter trends have a similar pattern, because they use the same random generation and theoretical power formula (but different coefficients). The overlap between 'steel' and 'titanium' comes from titanium's low cutting condition although 'titanium' demands more power for the same material removal rate. Normalized Root Mean Square Error (NRMSE) of 'steel' marks 17%, 'aluminum' 21% and 'titanium' 11%, respectively. Interestingly, all the three trends have two inflection points which starts to increase NRMSE. The learned powers coincide with the theoretical ones quite well inside the points. However, they have differences outside the points (accuracy drop-off). In case of 'titanium', the inflection points are located in around 1940 and 6050 theoretical powers. The samples (N=60) less than the first point scores NRMSE 39%, the samples (N=118) greater than the second point scores 13% and the samples (N=822) within the two points scores 7%.
- Result: this case study show the feasibility of data-driven analytic modeling based on feature vectors. Although our training set uses theoretical powers, we can apply the same method to actual machining data. Although the only difference of the FV set is 'workpiece material', our model structuring can make unit analytic models and classify a group of the models efficiently in many FV combinations. Although the learning input parameters use the process parameters correlated with cutting power, our method can find an applicable data pattern in unknown data correlation. This is main difference with a statistical approach (e.g., a regression model) that allows finding a correlated data pattern in a known problem. However, we find a practical issue when applying machine learning. The accuracy drop-off implies the machine learning performance is not good to predict a power at the sides because out-of-range data is not used as the training data. That is, the range of the learning input data is too narrow to properly represent the data pattern existed in the range. Therefore, the range of training data set should be well arranged, considering a threshold of model accuracy and a learning bias.

6. Conclusion

This paper presented the design of a big data analytics model, accompanied with the identification of its functional architecture. The work presented contributes toward: (1) the big data analytic model for machining process as a starting point for manufacturing process analytics, (2) the expansion of the composite analytic model to enable data-driven planning and control with faster decision making, and (3) the utilization of open platform tools makes SM possible for small and medium-size manufacturers.

We showed four leading-edge technologies – data analytics, Big Data infrastructure, MTConnect and STEP (-NC) – could be key techniques for realizing the SM paradigm. However, there are some limitations on the data acquisition through the use of a simulator and not through actual machining, the exclusion of optimization modeling in the circled functional architecture, and the partial integration of the implementation components. Future works include: (1) real data acquisition by construction of big data infrastructure in a shop floor, (2) the inclusion of the optimization modeling, which makes a virtuous circulation for continuous improvement, (3) the full integration of big data infrastructure and analytic modeling, and (4) the extension to other considerable sustainability performances besides power consumption.

Disclaimer

Any mention of commercial products is for information only; it does not imply NIST recommendation or endorsement, nor does it imply that the products mentioned are necessarily the best available for the purpose.

References

[1] SMLC (Smart Manufacturing Leadership Coalition). Implementing 21st century smart manufacturing – workshop summary report. 2011.

[2] McKinsey Global Institute. Manufacturing the future: the next era of global growth and innovation. Atlanta; 2012.

[3] ISO (International Standards Organization). ISO14649-1: Overview and fundamental principles. 2003.

[4] MTConnect Institute. MTConnect standard Part 3: streams, events, samples and condition. 2012.

[5] Posselt G, Kellens K, Thiede S, Renaldi, Herrmann C, Dewulf W, Duflou JR. Combining Machine Tool Builder and Operator Perspective towards Energy and Resource Efficiency in Manufacturing. 20th CIRP Int Conf Life Cycl Eng 2013.

[6] Diaz N, Redelsheimer E, Dornfeld D. Energy Consumption Characterization and Reduction Strategies for Milling Machine Tool Use. 18th CIRP Int Conf Life Cycl Eng 2013.

[7] Kara S, Li W. Unit process energy consumption models for material removal processes. CIRP Annals – Manuf Tech 2011;60:37~40.

[8] Winter M, Li W, Kara S, Herrmann C. Determining optimal process parameters to increase the eco-efficiency of grinding processes. J Clean Prod 2014;66:644-654.

[9] Bruce R. Statistical and machine-learning data mining: techniques for better predictive modeling and analysis of big data. 2nd ed. Boca Raton: Taylor & Francis Group; 2012.

[10] Winter M, Bock R, Herrmann C. Investigation of a new ecologically benign metalworking fluid in abrasive machining processes to substitute mineral oil based fluids. CIRP Conf High Perform Cut 2012:393~398.

[11] Amstead B, Ostwald P, Begeman M. Manufacturing process. 8th ed. New York: John Wiley & Sons; 1987.