Issues in Synthetic Data Generation for Advanced Manufacturing

Don Libes
National Institute of Standards and Technology
Gaithersburg, MD, USA
don.libes@nist.gov

David Lechevalier
Le2i, Université de Bourgogne,
Dijon, France
david_lechevalier@etu.u-bourgogne.fr

Sanjay Jain
The George Washington University
Washington, DC, USA
jain@email.gwu.edu

***Abstract*— To have any chance of application in real world, advanced manufacturing research in data analytics needs to explore and prove itself with real-world manufacturing data. Limited access to real-world data largely contrasts with the need for data of varied types and larger quantity for research. Use of virtual data is a promising approach to make up for the lack of access. This paper explores the issues, identifies challenges, and suggests requirements and desirable features in the generation of virtual data. These issues, requirements, and features can be used by researchers to build virtual data generators and gain experience that will provide data to data scientists while avoiding known or potential problems. This, in turn, will lead to better requirements and features in future virtual data generators.**

Keywords- virtual data; synthetic data; data generation; smart manufacturing

I. INTRODUCTION

Much of the research in advanced manufacturing involves the creation of models and simulation-based experimentation. Simulation leads to significantly faster results than physical experiments that require use of real machine tools and materials in a physical shop floor [1]. Such simulations and models require data for carrying out those experiments. These data can be representative of a large number of sources such as machine tools, robots, suppliers, etc. The data types are also quite varied such as material density and strength, machine tool wear and energy usage.

The concept of simulated data has been referred to in multiple ways in the literature. Many people use terms such as artificial data, virtual data, generated data, fake data, and synthetic data to all mean the same thing. In contrast, people denote data generated by physical machines as real data, physically-generated data, or live data. We (the paper authors) use both sets of terms interchangeably in speech but prefer synthetic data and real data in the written word. The term live data is occasionally used to emphasize physically-generated data that is being consumed within a small window of its generation. However, this can be confusing so we generally avoid that term or make clear in context what we mean.

A. Background

Synthetic data generation has received significant interest in recent years across a number of fields. A recent review [2] of the literature related to synthetic data identifies use in multiple fields including economics, urban planning, transportation planning, cyber security, weather forecasting, and bioinformatics. Interestingly this review didn't report any activity in manufacturing. The paper focused on synthetic data generation for learning analytics in the education environment and identifies some of the same issues that are discussed in relation to advanced manufacturing in different situations such as machine-learning training, or extension of real data set. Similarly, [3] focused on synthetic data generation for Internet of Things (IoT) environment to address some of the same challenges including limited access to real world data discussed here.

There have been papers that tangentially touch on synthetic data generation but with a primary focus on simulation. For example, [4] describes the European Virtual Factory Framework (VFF), an interoperability framework for factory modeling. The VFF includes a Virtual Factory Data Model (VFDM) for common representation of factory objects to evaluate performance of production systems and a Virtual Factory Manager to manage a shared data repository. Support for the external simulation tool, Arena, demonstrates the ability of data generation and potential for improved data interoperability [5].

The National Institute of Standards and Technology (NIST) has two efforts specifically in the area of data generation for manufacturing. The first is the STEP2M Simulator which simulates machine monitoring data, given process plans [6]. The process plans are Step-compliant data interface for Numeric Controls (STEP-NC) conforming [7] and the resulting generated data are presented using MTConnect [8]. A second project defined a virtual machining model that simulates a 2-axis turning machine [9]. This model uses STEP-NC [10] commands and material property as inputs, and uses equations to compute time and power consumption depending on the tool path. The simulation generates machine-monitoring data in MTConnect format usable for other simulations. In [11], the authors demonstrated the usability of these data with a 3-axis milling operation. They

integrated the earlier work into an agent-based model to provide capabilities to create a shop floor model [12]. More complex shop floor models could be built combining different agent-based models representing different operations. Such a combination would lead to the generation of data at the machine level but also at the shop floor level where data could be aggregated.

II. WHY IT IS SO HARD TO OBTAIN REAL DATA

There are many reasons why it can be difficult or impossible to get sufficient real data both by type and quantity. Libes et al describe how significant amounts of data exist but are nonetheless unusable [13]. For example:

- Data may be proprietary preventing access.
- Data may have timestamps but are not synchronized preventing data joining.
- Data frequency may be insufficient.
- Data may be formatted inadequately leaving ambiguity.
- Data may have inexplicable gaps.
- Data may have been generated with different underlying goals.
- Metadata may be inadequate or non-standard resulting in semantic confusion.
- Accuracy may be undocumented.
- Data provenance may be suspect (modified without documentation) or unknown.
- Data may be too costly to obtain.

We lack the space to explore all of these in detail but we will consider one as an example: data formatting.

Enterprises sit atop a vast collection of disparate data, likely produced by a multitude of heterogeneous sensors, and often ultimately stored in files formatted according to a variety of standards, with varying degrees of compliance. Significant amounts of data may follow no standards whatsoever. Using standard specifications such as XML (eXtensible Markup Language) and JSON (JavaScript Object Notation) can help [14]. However, problems such as under specification can remain and leave ambiguities. For example, using an XML attribute called "time" means little if there is no definition for how the string is to be interpreted: absolute with respect to UTC (Coordinated Universal Time) or local time zone? Is the time relative to the start of a process or something else?

While standards can be helpful, they are not panaceas. For example, equipment and software from different vendors may use different standards. Data, while still standard-compliant, can lose fidelity during interchange. Standards frequently have different levels of compliance that users may choose. Even highly specific standards do not guarantee data are usable. For example, the machine tool standard, MTConnect, covers only one direction of communication; so, correlation to commands may not be present or need reconstruction with timing uncertainties. Equally important, MTConnect does not cover all possible types of data that a machine tool can generate. For example, MTConnect defines a fixed set of statistics for DataItems. Kurtosis, a measure of peakedness relative to a normal distribution, is in the set. Skewness, a measure of symmetry, is not in the set.

All of these choices of standards and specifications have reasons for existence. For example, different vendors may have different reasons for their choices. These can include historical issues, expense in tracking developing standards, and interactions with other software. For example, the choice of OWL (Web Ontology Language) variants depends on how much need there is for expressiveness – the breadth of concepts that can be represented [15]. But greater expressiveness brings with it a loss of computational guarantees [16].

Data standards may be descriptive (describing practices) or prescriptive (defining practices). Each carries with it downsides. For example, descriptive standards may prevent the use of innovative techniques that are too new to be incorporated in standards while prescriptive standards may be ignored when better technology solutions are discovered. These dilemmas are particularly apparent in rapidly changing and highly-competitive fields. To allow variations and technological advances, some standards intentionally leave areas of ambiguity with a resulting ambiguity in the data.

III. USES OF SYNTHETIC DATA

Development and maintenance of data analytics applications, models, and real factories can all make use of synthetic data, albeit in different ways. For example, data analytics applications can use synthetic data to test that training algorithms perform adequately. Factories can also use the data to experiment with proposed changes. For instance, it is not possible to test a replacement machine tool before it is purchased and installed; however, data synthesized by a machine tool simulator can allow the factory to be modeled and tested as if the machine tool was present. Similarly, policy or algorithm changes can be tested before deployment in a real factory. Lastly, synthetic data can be a resource for test suites that exercise the full range of states of a system – including both normal operation and error conditions.

In short, synthesized data can be used throughout the lifecycle of a factory – from initial brainstorming to development to maintenance. We focus on several uses of synthetic data in this section.

A. Machine-Learning Training

Machine learning models are not explicitly programmed or based on a physical model but primarily by analyzing data. For example, neural networks are trained on data and typically produce networks that make no attempts to model the physics. There is no use of equations that correspond to performance of any real machines. An appropriate amount of data is necessary to make sure that machine learning algorithms well perform.

While defining the right amount of data is one area of research, it is equally important to understand the limitation of synthetic data in this context. For instance, a neural network that is trained on synthetic data is unlikely to provide a better model than the model used to generate the synthetic data in the first place. For this reason, users (e.g.,

neural networks) of machine-learning algorithms should track the provenance of any generated data used to train those algorithms. Without such provenance, mistaken assumptions may arise over the quality of decisions generated by machine-learning algorithms.

B. Verification

The American Society of Mechanical Engineers (ASME) defines verification as the process of determining that a computational model accurately represents the underlying mathematical equations and their solution [17]. During development, verification is used to ensure that data analytics applications and models meet their design requirements and specifications. While direct methods are available in limited ways for verification, data can be used in verification, in some cases as input to be consumed and in some cases as output to be tested against.

Being able to generate indefinite amounts of data allows more extensive testing than would otherwise be possible through limited reference data produced by physical machines. In addition, parameters can be changed to generate different synthetic data that would be very expensive or impossible to obtain by using physical machines.

C. Validation

ASME defines validation as the process of determining the degree to which a computational model is an accurate representation of the real world from the perspective of the intended uses of the model [17]. Validation confirms that applications or models match the needs of the customer. Data analytics applications and models should be validated – ideally, during development and continuing through the lifecycle. Data are useful in this validation process, particularly in cases when physically realized applications, models, and physical instances will not yet be available. Having synthetic data – both input and output – as if the enterprise existed, can be used to test that requirements are valid. Throughout the life of applications, models, and real factories, synthetic data can be used to validate that proposed changes continue to result in valid results.

D. Optimization

Optimization is the process of improving algorithms or applications to produce the best results, and improve the system under study. (This term is often mistakenly used for the process of "improvement" which is more practical while "optimality" is aspirational.) Optimization is frequently performed by simulation.

Synthesized data can be used for optimization, such as in factory models. These models are not algorithmic machine-learning models (see earlier) but rather simulations of factories (or subsets) that can be tested for performance optimality. For example, work cell and machine tool arrangements of a factory floor can be simulated and tested prior to deployment in order to select the best layout. Since such simulations have no real counterpart, synthesized data can be used in the simulation. For example, part flow, machining time, machine breakdown, etc., can be used during simulation runs to produce and measure more realistic performance metrics.

E. Augmenting Real Data

For many reasons, real data can be missing or insufficient. For example, sensors may be incapable of collecting data quickly enough. Many techniques are possible to replace missing data values. For example, simple mathematical interpolation can produce data that fill gaps and reflects existing data. However, interpolation can distort data. This may sound counterintuitive but means that interpolated data give the mistaken impression that the data are smoother than it actually is.

Analytic applications can automatically fill in missing data values by essentially running algorithms in reverse producing data that is more representative of real data. For example, once a neural network has been trained on a data stream, the neural network can be used to produce more data. However, by its very nature, such synthetic data will not have any impact on analytics rendering it of no intrinsic value except perhaps for pro forma purposes such as creating more complete visualizations.

Missing data can be used during development (for example, for optimization) but is also useful when a factory is in production. In that case, there may be instances or periods when data are unavailable whether due to a broken sensor or communications problem. In this case, missing data can be replaced with synthetic data in real time.

IV. Challenges for Data Generation and Collection

While synthetic data have many uses, challenges exist, many of which may not be obvious until encountered. Thus, we describe some of most significant challenges in data generation.

A. Quantity Sufficiency

Algorithms for synthesizing data require some minimum amount of realistic data although the data may be of a different type. For example, generation of energy data may require a machine specification and material data. Some algorithms may generate data by using data similar to what is desired – for example, by increasing or decreasing the variance of an existing dataset.

Data analytics algorithms generally increase accuracy with more data when that data cover more of the descriptive space to be analyzed. Of course, more data can slow down algorithms worsening time performance. More data can also be unnecessary, essentially providing no new information. More data can even introduce artifacts that are irrelevant and hurt model performance. However, sufficient quantities of training data can be significant, for example during walk-forward testing while training neural networks. Knowledge of the quantity required for training can be as useful as the choice of the data itself and which data are more likely to be seen in the real system. Defining the right quantity might be possible by calculating the squared error of the data analytics model while training and testing to define if it is over or

under fitted. Defining the right quantity of data before training is, however, a more complex issue.

B. Timing

Closely related to quantity sufficiency are issues of timing. These issues can be complex and be impacted by many parts of an enterprise but sensor behavior is the simplest place to explore such issues.

Sensors operate in two modes. Some sensors run free meaning that as soon as they have finished reporting data, they begin collecting or sending additional data. They are constantly busy. This can be useful for algorithms that want to consume as much data as possible. Other sensors may be synchronized to an internal or external source. Such sensors artificially discretize readings. Being able to synthetically simulate such data sources can be useful for properly modeling real-world factories.

Generators may need to be capable of generating both types of data sources with the ability to generate data at arbitrarily large rates.

C. Dynamic Generation

Data analytics developers often do not need or want to store all data in advance of the need. The amount of data needed may be too large to store. Generating data dynamically can also better reflect the way data are consumed as well as leaving open the possibility for feedback that can change the data generation process. The converse is also true. Consider data coherency. Data coherency can be disrupted by updating data at the same time it is being consumed but without synchronization. By avoiding the storage of large amounts of synthetic data, data coherency is maintained with lower costs (e.g., locking protocol overhead). For these reasons, it can be useful to generate and immediately consume data dynamically whenever possible.

On the other hand, storage of large data sets may be useful for certain types of analysis. When time and space are not significant factors, data can be evaluated more thoroughly. Resulting models that are relatively static or are already optimal do not need to be continuously or frequently retrained if there is no benefit to doing so.

1) Feedback and Control

Real-time performance applications make decisions that are fed back to the system thereby affecting its performance. Naturally, changing the outcome of such decisions changes the performance of the system which in turn changes any performance data produced by the system that would ordinarily be fed back to the analytics applications.

It is desirable for data generators to be able to incorporate feedback in order to control future data generation. This is not necessarily easy as feedback control can introduce non-trivial synchronization issues. For example, lock-step synchronization is typically not necessary as there are lags in real factories when data-driven decisions affect controllers and similarly how quickly sensors can return feedback. Determining at what levels this is accounted for (e.g., generators, models, simulations) and with what accuracy can be challenging.

2) Speed

For physical data generation, data are generated at whatever speed the producers can generate it. For data analytics applications, that speed is generally not of interest. A data analytics developer is not consuming real data directly from a shop floor except in rare circumstances. Developers almost always use models or simulations that can run faster than real-time. For that reason, when data are generated dynamically, it is desirable to have data generators that can run faster than real-time, preferably as fast as the simulation itself.

In some contexts, data generation may need to be slowed down to real time to ensure that applications are able to provide feedback in a timely manner to impact factory performance. For example, if batch dispatching decisions occur every minute and an analytic application promises to improve system performance through better dispatching, the application should be able to execute and respond within sub-minute intervals. The capability of such an application to perform well in a real system should be tested with a manufacturing simulation that ensures real time performance.

D. Data Hiding / Suppression

For a variety of reasons, it can be useful to suppress or hide (i.e., to not use) synthetic data during analysis (or development and testing of analysis). It may be desirable to hide details of the implementation as well. Some of these considerations are presented in this section.

1) Intentional hiding

For various reasons, more data may be available than is desired. Thus, it may be useful to hide some of the data. For example, the analytic engines may be incapable of consuming all the data, particularly if time limits are an issue. This is a concern with the need for real-time results. Data sampling might be a solution to achieve this task without modifying the dimensions represented in the data set.

2) Walk-forward testing

Walk-forward testing consists in training a machine learning model with a subset of the data, and testing the trained model with an unseen subset of the data. It is another example where data must be hidden – at least initially. This may be repeated on many quantities of unseen data in order to ensure that systems are not over-trained. The aim is to create a model that is not necessarily expert at recognizing only the training data but capable of recognizing data that is also likely to be produced in similar scenarios.

3) Filtered

Data may be hidden because it is withheld (filtered) for a variety of reasons. For example, data that is clearly wrong or exceeds certain bounds may be suppressed depending on the needs of the data analysis applications. An analytics

application may be designed to run in a live system where data are guaranteed by database constraints as to its quality. During development, guarantees may be maintained when the generator is providing the data without any database filtering. Describing what should be filtered can be arbitrarily complex as there are many reasons for filtering and the reasons can be combined in complex ways. For instance, analytics software under development might be restricted from out-of-bounds data (previous example) as well as malfunctioning sensor readings that are within bounds but inaccurate.

4) Black box generators

A data generator or its underlying models and data may be treated as a black box. Hiding the implementation can prevent certain types of analytics that may give an unwarranted impression of a tool that is more universal than it is, simply because the implementer (or software) can "see" the implementation and use the implementation rather than the data as a basis for analytics.

Data hiding also prevents premature optimization as well as shortcuts that can overlook problematic data. For example, an optimizer that knows it will never see data outside a certain range may learn that it is not necessary to handle such data. When it is faced with such unexpected data, perhaps by a misbehaving process upstream, the optimizer is likely to behave inappropriately since it has not been trained on such data.

E. Data Quality

Sensors quantize data, have lags, fail, and have other issues. So, ideal data should never be expected from real enterprises. However, there is value in building ideal data generators. For example, certain types of algorithms require or perform much better when optimal goals are provided. This is the case with many non-heuristic approaches to nondeterministic polynomial (NP) problems such as identifying optimal routing [18]. Nonetheless, most interest for data analytics is in creating more realistic data. The following sections describe data quality issues in data generation.

1) Reliability

Physical systems can be unreliable so it is useful for synthetic data to be able to reflect that. This unreliability can be difficult to model as misbehavior can come about in so many ways. For example, sensors can behave erratically, communication can encounter interference, or power can be dirty. Each causes reliability issues.

Within each type of problem, there is a spectrum of unreliability. Reliability can also change over time, typically increasing but occasionally decreasing. In short, reliability is complex so generating realistic unreliable data is complex.

2) Accuracy

The accuracy of physical sensors must be accounted for by data generators. While an accuracy limit may suffice, it is more realistic to produce a range that models the physics of the sensor; however, this is often not possible. Instead a variety of ranges are used such as a Gaussian distribution, Poisson distribution, or Bayesian-based distribution.

3) Uncertainty

There are many sources of uncertainty. For example, machine-tool manufacturers themselves may not have a usable mathematical model for their products. Even if they do have a model, it may specifically omit aspects that are difficult or entirely unknowable, for instance. So for these and other reasons, physical device manufacturers often state only a range of accuracy leaving questions of distributions a difficult challenge between the designer of a generator which mimics that device and the user who configures the data generator. Quantifying and aggregating (epistemic and aleatory) uncertainty generated by different sources during the simulation is complex and requires to clearly identify and evaluate each source of uncertainty [19].

4) Adjusted

Sensor data can often be adjusted at intermediate processing nodes. For example, data can be joined at a network node that collects data from several sensors for each entry in a log. Adjustments can include normalization, scaling, and quantization. This means that analysis software may only see adjusted data rather than raw data. For this reason, data generators should be able to produce either raw data or data adjusted in a variety of ways.

F. High-Level Key Performance Indicators Without Low-Level Data

In many scenarios, the company is interested in high-level (i.e., enterprise or shop floor level) Key Performance Indicators (KPIs) or data and not in low-level (i.e., process or machine level) KPIs or data. For example, machine tool measurements may not be significant to higher-level processes such as machine-tool inventory predictions despite the basis on machine tool information. Many analyses only use high-level KPIs as input such as finished product cost, inventory cost, and goals such as minimizing late orders. Once the KPIs are created, the low-level data are never again used. In such scenarios, the low-level data are not necessary if the KPIs can be created independently.

Data generators that provide high-level KPIs may not need to model the underlying physics if the KPIs are good enough. Of course, "good enough" is challenging to define. Generating KPIs in a top-down approach (for example, driven by organizational goals) may be harder than generating low-level data and doing a higher-level simulation to arrive at KPIs. Whether this is true depends on the fidelity requirements of the KPI-consuming applications [20].

G. Well-Described Scenarios

To generate data that are useful, scenarios must be identified to define the context in which the data applies. Finding and describing such scenarios can be difficult for several reasons. There are many variables that likely differ for every factory: products, machine tools, goals, and costs. These are almost always mixed. For example, a

manufacturer may have a mix of machine tools and the number, types, and layout will differ from one manufacturer to another. Similarly, one manufacturer's goals are likely to differ from another. One manufacturer may have contracts with suppliers and utilities while another manufacturer will have different suppliers and other constraints. Machine tools will be of different ages and exhibit different performance characteristics. Where semi-automation is an issue, human factors will be unique as well.

To accommodate differences such as constraints or goals, data generator parameters can be adjusted; however, thought should be given to whether these parameters are intrinsic to data generation. For example, while both machine tool performance characteristics and goals (such as "use minimal power") will affect the generated data, the former is intrinsic to the generated data while the latter can be considered a dependent variable that is only meaningful to a higher level of control.

H. Manufacturing Levels

It is customary to organize a hierarchy of manufacturing at different levels of operation, e.g., models for machine tools, workcells, factories, enterprises. Additionally, customers and supply chains may also be modeled. Developing simulators or analytics applications may be specific to a level or may include multiple interacting levels such as model supply chain level and its interaction with the factory level [21].

Data at any one level are likely to have a strong dependence on other levels of a hierarchy. Obviously, a factory model depends greatly on the performance of the operations within. Less obviously, a machine tool depends on the goals of its workcells or even higher levels. For example, a machine tool may wear less by running at a slower speed which is only acceptable if the goals of the factory or workcell permit.

I. Data Type Complexity

Many different types of data exist in a real factory. For example:

- Material Data: costs/characteristics/physics of material, water, energy
- Process Data: task time, customer demand, production schedules
- Product Quality: geometry, structural integrity, performance
- Manufacturing equipment: efficiency, reliability, spare capacity
- Employee data: salary, hours worked, employee skills

It is possible to generate all of these but the more types of data generated, the more work is required. For practical purposes, many of these are unnecessary and only depend on the particular goals of the analysis. Not all goals can be achieved simultaneously as many will always conflict. For example, it is generally impossible to achieve both minimal energy usage and time. There is always a tradeoff. For the same reason, there is no minimally optimal data set. The scope of data sets varies based on the objectives of the analysis and the level of abstraction desired by the analyst.

J. Repeatability, Reproducibility, and Provenance

Repeatability, reproducibility, and provenance are closely related, all having to do with the confidence in the ability to recreate generated data.

1) Repeatability

Repeatability of data generation from a physical enterprise is difficult. For synthetic data generation, repeatability is generally straightforward. Generators must be capable of publishing and accepting random number seeds. Generator code must not require anything else that could change the data output from one run to another. The code itself must be published in such a way that the code remains the same so that others can rerun the same data generator with identical results. This can be ensured with a signing procedure such as providing the results of a cryptographic hash function.

2) Reproductibility

While repeatability is straightforward, reproducibility can be more difficult since underlying libraries and computer hardware can cause output differences despite identical high-level code. Languages such as C are particularly notorious for this. These differences are not necessarily a bad thing but intended to give programmers more control over efficiency of an implementation. However, programmers are free to ignore (or may be unaware of) such subtleties which can lead to non-reproducibility. C is not alone. Many higher-level languages have subtle dependencies as well. In [22], the authors review correct approaches to assess the consistency of measuring process. These approaches can be similarly applied to data simulation to ensure consistent generation of data.

3) Provenance

Some data may not need reproducibility as long as it is suitable for its purpose and its provenance can be ascertained. Provenance through a digital signature provides a guarantee of the source of the data and that it has not changed. Only the signing of data and its metadata is necessary.

Closely related to provenance is traceability. While data may be provably shown to have come from one data supplier, it may be of no value if others cannot ever hope to build a factory that reproduces it.

K. Model Type Choice and Provenance

There are a variety of model types used for data generation. There may also be hybrid combinations of these models.

1) Physics-based Models

Physics-based models are based on equations that reflect our understanding of what physically happens in real life. In theory, physics-based models are the optimal way of modeling any manufacturing process since they capture all

information and effectively produce our best understanding of a process.

However, these equations can be difficult to obtain, relying on, for example, a machine tool manufacturer which may have little incentive to provide specific types of equations or explanations. Alternatively, the equations may exist but fail for a variety of reasons to correlate with what is observable. For example, machine tool wear may be a factor in the equation but is effectively unmeasurable because it requires destructive testing or machine disassembly that voids a warranty.

For practical reasons, it is rarely the case that we truly have accurate equations that hold in all situations. For example, it would not be sensible to have quantum-level models when Newtonian models are sufficient for almost all purposes. Similarly, while physics-based models are often implemented using continuous simulation, the ability to achieve arbitrary levels of resolution and scale is generally overkill for most data generation needs.

The result is that physics-based models are always idealized and require adaptation to be used for realistic data generation.

2) Empirical Models

Empirical models are based on data, ideally from physical systems. These data are then used to build machine-learning models using techniques such as neural networks and Gaussian networks. Such empirical models are generally used with discrete simulation techniques although this is not mandatory.

The result is that empirical models are adequate to provide good data generation for some situations. However, for situations that they have not been trained for such as edge cases and other unusual events, empirical models can fail to generate suitable.

3) Special-purpose Models

Special-purpose models can be based on other techniques besides those based in statistics or physics. For instance, a data generator could be used to produce intentionally bad data specifically to test the behavior of analysis tools. It may suffice to use a stream of zeros or just open a file of random data. Similarly, an ideal data generator (see 4.5) can be used to test minimal fitness or conformance. Such data can be based on published or nominal specifications from a manufacturer with no regard to a statistical or physical model. Data could also be aspirational, referring to a goal for which no known algorithms can achieve but represent provable limits.

L. Integration, Interoperability & Standards

While it is important that data be meaningful, it is also important that data are in a form that allows it to be easily used.

1) Data Interchange Standards

It is desirable for data to be in a form that uses well-recognized standards. A collection of standards, often considered as a stack or hierarchy, may be used together although rarely is the delineation clean. For example, standards such as XML or JSON may be used for low-level syntactic formatting while International Standards Organization (ISO) 10303 for product manufacturing information and ISO 22400-2:2014 (KPIs) apply to higher-level manufacturing semantics [23], [24], [25]. Many standards exist to address other concerns. For example, CMSD (Core Manufacturing Simulation Data) and CSPI (COTS (Commercial Off-The-Shelf) Simulation Package Interoperability) are standards for facilitating the use of simulation models such as shop floor configurations [26], [27]. Standards such as MTConnect and Open Platform Communications Unified Architecture (OPC-UA) may be used to exchange manufacturing information from machine tools while Representational State Transfer (REST) and Simple Object Access Protocol (SOAP) are examples of standards for carrying out communications [28],[29].

Data generators must be able to produce data in a form that either conforms to the relevant standards or is readily adaptable to them.

2) Plug-in interoperability

Data generators should be able to serve as plugins to other systems such as domain-specific testbeds and model design software that may provide other services such as model transformation or optimization. By using standards, plug-in capability can be more easily supported and generators can more likely be incorporated into additional systems. In the reverse sense, generators should also be able to use models based on a plugin architecture. For example, a data generator only capable of generating the results of a milling machine using aluminum would see limited use. Allowing the plugging in of models of tools and materials and parameterizing factors such as feed rates, a generator would be much more useful, leading to more of a widely applicable generator.

V. Summary and Concluding Notes

Limited access to real-world data is a significant impediment to advanced manufacturing. Use of virtual data is a promising approach to make up for the lack of access. We have presented issues, challenges, and desirable features in the generation of virtual data. These can be used by researchers to build virtual data generators and gain experience that will provide data to data scientists while avoiding known or potential problems. This, in turn, will lead to better requirements and features in future virtual data generators.

Three areas for future research and development are of particular interest and deserve increased attention and effort.

A. Test Data Repositories

Many people experimenting with data analytics would benefit from repositories of both real data and synthetic data. Such repositories would allow multiple academic researchers or commercial companies to be confident that they are using the same data in creating and testing software to deal with what are intended to be common scenarios. It is also

desirable to provide configuration data to reproduce the raw data in the repositories so that it can be reproduced as well as modified.

Data repositories should also have other aspects such as areas for algorithms and models that have been proposed to address data sets in the same repository. Ideally, documentation areas and discussion forums would be helpful as well. For example, observations or questions about particular data or data configurations would enable others to make progress by more easily re-using earlier results.

Repositories have been established that incorporate some of these ideas. For example, Bosch has created a challenge that includes measurement data produced from production lines [30]. One example challenge is to "predict which parts will fail quality control." The Bosch data sets and competitions are hosted on Kaggle, a service for general data science challenges [31]. Another example is the NIST Smart Manufacturing Systems Test Bed which makes available data from a manufacturing facility which resembles a small manufacturing shop [32]. Sets of data can be downloaded or queried. In addition, dynamic data streams can be monitored using MTConnect.

B. Standards

Standards development is already an area of intense interest. However, there are gaps in standards that would facilitate data generation and publication of synthetic data. For example, PFA (Portable Format for Analytics) is a useful specification in which to express data analytics [33]; however, while a PFA-enabled host can generate data usable to other software, PFA lacks the ability to control the seeding of its random number generators which limits its flexibility and repeatability. More work is needed on standards to better support data generation.

DISCLAIMER

No approval or endorsement of any commercial product by the National Institute of Standards and Technology (NIST) is intended or implied. Certain commercial software systems are identified in this paper to facilitate understanding. Such identification does not imply that these software systems are necessarily the best available for the purpose.

AKNOWLEDGMENTS

David Lechevalier's work on this effort was supported by National Institute of Standards and Technology's Foreign Guest Researcher Program.

REFERENCES

[1] M. Saadoun and V. Sandoval, "Virtual Manufacturing and its Implications." Virtual Reality and Prototyping, 1999.

[2] A. M. Berg, S. T. Mol, G. Kismihok, and N. Sclater, "The Role of a Reference Synthetic Data Generator within the Field of Learning Analytics," J. Learn. Anal., vol. 3, no. 1, pp. 107–128, 2016.

[3] J. W. Anderson, K. E. Kennedy, L. B. Ngo, A. Luckow, and A. W. Apon, "Synthetic data generation for the internet of things," presented at the 2014 IEEE International Conference on Big Data, 2014, pp. 171–176.

[4] W. Terkaj and M. Urgo, "Virtual factory data model to support performance evaluation of production systems," in Proceedings of OSEMA 2012 workshop, Graz, Austria, 2012, pp. 24–27.

[5] "Arena Simulation Software." Rockwell Automation, 2017.

[6] G. Shao, S. Jain, and S.-J. Shin, "Data Analytics Using Simulation for Smart Manufacturing," in Proceedings of the 2014 Winter Simulation Conference, Savannah, GA, 2014.

[7] M. Albert, "STEP NC - The End of G-Codes?," Mod. Mach. Shop, Mar. 2006.

[8] P. Warndorf, "MTConnect Institute Releases Version 1.3.0 of the MTConnect Standard." 2014.

[9] S. Jain, D. Lechevalier, J. Woo, and S. Seung-Jun, "Towards a virtual factory prototype," Winter Simulation Conference (WSC), 2015, pp. 2207–2218.

[10] "ISO 10303-238 (2007) Industrial automation systems and integration - Product data representation and exchange - Part 238: Application protocol: Application interpreted model for computerized numerical controllers." Geneva: International Organization for Standardization, May-2007.

[11] D. Lechevalier, S.-J. Shin, S. Rachuri, S. Foufou, Y. T. Lee, and A. Bouras, "Simulating a virtual machining model in an agent-based model for advanced analytics," Journal of Intelligent Manufacturing, Sep. 2017.

[12] Jain, Sanjay, and David Lechevalier. "Standards based generation of a virtual factory model." In Proceedings of the 2016 Winter Simulation Conference, pp. 2762-2773. IEEE Press, 2016.

[13] D. Libes, S. Shin, and J. Woo, "Considerations and recommendations for data availability for data analytics for manufacturing," in 2015 IEEE International Conference on Big Data (Big Data), pp. 68–75, 2015.

[14] "JSON: The Fat-Free Alternative to XML." JSON.org.

[15] "OWL Web Ontology Language Reference." World Wide Web Consortium.

[16] B. Motik, B. Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz, "OWL 2 Web Ontology Language Profiles." W3C.

[17] "Guide for verification and validation in computational solid mechanics." ASME, 2009.

[18] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. New York: W.H. Freeman, 1979.

[19] Oberkampf, William L., Sharon M. DeLand, Brian M. Rutherford, Kathleen V. Diegert, and Kenneth F. Alvin. "Error and uncertainty in modeling and simulation." Reliability Engineering & System Safety 75, no. 3: pp. 333-357, 2002

[20] D. Kibira, K. C. Morris, and S. Kumaraguru, "Methods and Tools for Performance Assurance of Smart Manufacturing Systems," J. Res. Natl. Inst. Stand. Technol., vol. 121, pp. 287–318, 2016.

[21] S. Jain, E. Lindskog, J. Andersson, and B. Johansson, "A Hierarchical Approach for Evaluating Energy Trade-offs in Supply Chains," Int. J. Prod. Econ., vol. 146, no. 2, pp. 411–422, 2013.

[22] Watson, P. F., and A. Petrie. "Method agreement analysis: a review of correct methodology." Theriogenology 73, no. 9: pp. 1167-1179, 2010.

[23] A. Katzenbach, S. Handschuh, and S. Vettermann, "JT Format (ISO 14306) and AP 242 (ISO 10303): The Step to the Next Generation Collaborative Product Creation," NEW PROLAMAT, pp. 41–52, Jan. 2013.

[24] "ISO 10303-1:1994 Industrial automation systems and integration -- Product data representation and exchange -- Part 1: Overview and fundamental principles." ISO, Dec-1994.

[25] "ISO 22400-2:2014 Automation systems and integration -- Key performance indicators (KPIs) for manufacturing operations management -- Part 2: Definitions and descriptions." ISO, Jan-2014.

[26] Y.-T. T. Lee, "A Journey in Standard Development: The Core Manufacturing Simulation Data (CMSD) Information Model," J. Res. Natl. Inst. Stand. Technol., vol. 120 (2015).

[27] "SISO-STD-006-2010 Standard for Commercial-off-the-shelf Simulation Package Interoperability Reference Models." Simulation Interoperability Standards Organization (SISO), Inc., Mar-2010.

[28] W. Mahnke and S.-H. Leitner, "OPC Unified Architecture - The future standard for communication and information modeling in automation," ABB Rev., vol. 3/2009, pp. 56–61, Mar. 2009.

[29] J. Mueller, "Understanding SOAP and REST Basics And Differences." Smartbear.com, Jan-2013.

[30] "Bosch Production Line Performance." Bosch, Aug-2016.

[31] "Kaggle: Your Home for Data Science." Kaggle, Inc.

[32] T. Hedberg and M. Helu, "Smart Manufacturing Systems (SMS) Test Bed." National Institute of Standards and Technology.

[33] J. Pivarski, "Portable Format for Analytics: moving models to production." KDnuggets, Jan-2016.57670.