Modeling and Analytics for Cyber-Physical Systems in the Age of Big Data

Abhishek B. Sharma, Franjo Ivančić, Alexandru Niculescu-Mizil, Haifeng Chen, Guofei Jiang
NEC Laboratories America
{absharma,ivancic,alex,haifeng,gfj}@nec-labs.com

ABSTRACT

In this position paper we argue that the availability of "big" monitoring data on Cyber-Physical Systems (CPS) is challenging the traditional CPS modeling approaches by violating their fundamental assumptions. However, big data also brings unique opportunities in its wake by enabling new modeling and analytics approaches as well as facilitating novel applications. We highlight a few key challenges and opportunities, and outline research directions for addressing them. To provide a proper context, we also summarize CPS modeling approaches, and discuss how modeling and analytics for CPS differs from general purpose IT systems.

1. INTRODUCTION

Cyber-Physical Systems (CPS) have the potential to transform the way we live and work. Smart cities, smart power grids, intelligent homes with network of appliances, robot assisted living, environmental monitoring and transportation systems are examples of complex systems and applications [3, 18]. In a CPS the physical world is integrated with sensing, communication, and computing components. These four components have complex interactions.

The complexity of CPS has resulted in model-based design and development playing a central role in engineering CPS [6, 9, 12]. Naturally the sensing component of CPS is critical to the modeling and management of CPS because it provides real operational data. The goal of sensing is to provide *high quality* data with *good coverage* of all components at *low cost*. However, these goals may not always be achievable. E.g. we can use high fidelity sensors such as loop detectors and radars to detect the traffic on a road, but these sensors are expensive and hence, cannot be used to cover a large metropolitan area. Smart devices with GPS (e.g. smartphones or GPS on taxis) can provide good coverage but their data quality is low [19].

Traditionally, the design of the sensing component of a CPS focused on how to deploy a limited number of sophisticated, reliable, and expensive sensors to optimize coverage of an environment or physical phenomenon [19]. However, advances in sensing and communication technologies over the past 10-15 years have disrupted the traditional way. Sensors have become smaller and cheaper, and with the maturity of wireless networking, we can now deploy a large number of them to collect massive amount of data at low cost. The *Mobile Millenium* traffic information system fuses data from GPS-enabled phones, and GPS in taxis with data from sophisticated sensors such as radar and loop detectors to estimate traffic in the San Francisco metropolitan area [3].

Today, low cost, ubiquitous sensing is driving a paradigm shift away from *resource constrained sensing* towards using *big data analytics* to extract information and actionable intelligence from massive amount of sensor data [8, 19]. In this paper argue that the availability of *big monitoring data* on CPS creates *challenges for traditional CPS modeling* by breaking some of the assumptions, but also provides *opportunities to simplify the CPS model identification task, achieve proactive maintenance*, and *build novel applications* by combining CPS modeling with data driven learning and mining techniques.

We also argue that *combining CPS modeling with data analytics cannot be accomplished by simply transplanting approaches from the general purpose computing domain* – e.g. techniques for performance modeling and analytics in services like Facebook, Google's web services, etc. Though there are obvious similarities between CPS modeling and software engineering, a few important differences also exist. These stem from the marriage between computer science and control theory, and the special role of time in CPS area. Derler et al. note that in IT systems a task's execution time is a *performance* issue, and it is not *incorrect* to take longer to perform a task, unlike in a CPS where a task's execution time may be critical to it's correct functioning [9]. We discuss these point in more detail in Section 3.

The rest of this paper is organized as follows. We provide a brief overview of CPS in Section 2, and discuss how modeling and analytics for CPS differs from IT systems in Section 3. We then discuss the challenges to traditional modeling approaches (Section 4), and the opportunities for applying bid data analytics (Section 5). We summarize our position and conclude in Section 6.

2. CYBER-PHYSICAL SYSTEM MODELS

A CPS consists of a tightly coupled integration of computational elements with physical processes. The computational elements rely on sensors to monitor and control the physical environment and processes. The computing resource controls the physical processes using a variety of control objectives, where feedback loops impact computations as well. Before summarizing approaches to CPS modeling, we want to define our use of the terms *modeling* and *analytics* in the context of CPS. We use the term *modeling* to refer to a formal approach to designing and engineering CPS [9], whereas *analytics* denotes learning and mining approaches for extracting knowledge from monitoring data, and this knowledge can lead to actionable intelligence [8].

We believe that modeling will continue to remain a criti-

cal part of CPS design, development and operations, i.e. big data analytics will complement not supplement CPS models. This is because models offer several advantages [9]: (1) they can have formal properties such as determinism that we can prove, (2) they can be used to capture a system's evolution, (3) they enable analysis and simulation to help us detect design defects, and in some cases, they can be used to automatically synthesize implementations (e.g. code generation). Academia and industry has built an array of tools for CPS modeling such as Matlab Simulink and the Ptolemy suite [4]. However, with advancements in sensing, communications, and cloud computing, the term CPS now also includes large, complex systems such as the Internet of things, smart cities, the power grid, transportation networks, etc. Modeling these systems is challenging and often involves making simplifying assumptions to achieve tractability. In the worst cases, it might not be feasible to build accurate models even when detailed and sufficient amount of monitoring data is available. Big data analytics can be useful in such scenarios [8].

System model. A large class of CPS are modeled as a finite collection of operational modes in a physical environment [9, 14, 15, 17]. In this view, a CPS switches between m different modes, and the n-dimensional system state is represented as a combination of continuous and discrete components. For example, the state of a moving car can be represented as a combination of the current gear (discrete part) and continuous metrics such as speed, torque, etc., while the operating modes could be separated into classes such as *stopped*, *driving-forward*, and *reverse-driving*.

A discrete-time (or sampled) model for CPS can be represented as a set of m functions

$$y_t = f_{s_t}(\vec{x}_t) + \epsilon_t \tag{1}$$

where y_t is a time-varying (scalar) property of the system that is of interest (e.g. the trajectory of a moving vehicle),

$$\vec{x}_t = \{u_{t-k}, \ldots, u_{t-1}, y_{t-l}, \ldots, y_{t-1}\} \tag{2}$$

is the continuous state of the system at time t consisting of (sensed) external environment inputs u_j, $j \in \{t-k, \ldots, t-1\}$ over the k previous time slots, such as wind speed for example, and outputs y_i, $i \in \{t-l, \ldots, t-1\}$ over the l previous time slots, and ϵ_t is the noise. The function f_{s_t} predicting the current output y_t given the continuous state $\vec{x}_t$ depends on the discrete state at time t, $s_t \in \{1, \ldots, m\}$. Note that formally, we can consider all discrete state components be encoded in the operating mode. However, for performance reasons, in CPS developement practice as well as implementation such a strict distinction between continuous components and discrete components of the system state is not always followed. That is, in general, all discrete system components are not hidden into the operating modes, in order to avoid an unecessary blow-up of these modes.

Model identification. A key factor determining the complexity of learning a model for a CPS is whether the discrete state sequence s_t is known or not, and whether it is completely or partially observable during system runs. If we know or can observe s_t for all t, then we only need to estimate the individual functions f_s, $s \in \{1, \ldots, m\}$ for the m different discrete states or operating modes to gain a complete characterization of the whole system. For example, if we model f_s as AutoRegressive with eXogenuous inputs (ARX) model, then we can estimate its parameter using regression [17]. Support Vector Regression (SVR) can be used to estimate nonlinear functions f_s [14].

A more realistic and challenging case is when the discrete state sequence is not known. Here we need to estimate both the discrete state sequence (or equivalently when the system switches from one operating mode to another) and the system evolution functions f_s for each mode. An unknown or hidden discrete state sequence presents two scenarios: (1) a change in discrete state depends on the continuous state, and (2) discrete state can change independently of the continuous state. The latter systems are called *switched systems*. Different CPS model identification techniques are needed for the two scenarios (see [14, 17] for more details).

3. CPS VS. IT SYSTEMS

In our view there are four key differences that make big data modeling and analytics for CPS different from similar issues in the general purpose computing domain. The first two are: (1) The tight interaction of computing elements with the physical world through feedback control loops, and (2) A rigorous engineering process for mission critical CPS, when compared to standard software engineering practice for IT applications, where engineers cannot always rely on constant software updates to patch earlier problems. These directly lead to a formalized *model-based design paradigm* [6, 12] exemplified by a variety of engineering tools such as *Simulink/ Stateflow*, Ptolemy [4], etc.

Third, CPS systems exhibit many more operating modes compared to IT systems. We can think of a web service operating in different modes based on the traffic pattern, e.g. day vs. night, but often there is a small number of such modes. With few operating modes, we can combine domain knowledge and "brute-force" search to determine the state change points, and learn separate models. However, this approach will not scale to many discrete states that are not easy to distinguish, as is the case for complex CPS [9].

Fourth, we also expect the role and interpretation of analytics to be different in CPS. Often, when working with IT systems data, we can achieve good generalization with standard machine learning techniques by increasing the state space. E.g. instead of using only the number of request arrivals, we can also use its first and second derivatives as features. This may help us distinguish between different operational modes. However, with the paradigm shift to low cost, ubiquitous sensing such approaches become an *art* – how many and which variables to analyze? what do various transformations (e.g. derivatives, logarithms, etc.) represent? Ad-hoc analytics approaches are unlikely to make an impact in CPS area partly due to the need and well-established tradition of rigorous model-based design and development.

Based on these differences, we believe that big data modeling and analytics for CPS merits further independent research. We outline key challenges and opportunities, and discuss research directions to address them in the next two sections.

4. CHALLENGES AND OPPORTUNITIES

The Big Data paradigm shift, enabled by low cost, ubiquitous sensing, presents new challenges and opportunities in CPS area.

Challenge I: Partial knowledge of input-output relations. Models for CPS assume that the input and output variables (u_t and y_t in eq. 1-2) for each subsystem are known (see Section 2). This assumption does not always hold in practice. Often sensor data from a CPS is in the form of time series and today it is not uncommon to encounter datasets with thousands of metrics [5, 8]. In such cases we might have only partial information about the input-output relations: even domain experts and operators may not have complete information.

Similar situation is faced by large network and IT system operators. With distributed computing systems, several factors make it impossible for humans to understand the entire system: large scale, complex interaction across components, the demand to deploy a heterogeneous class of applications using limited resources while still maintaining high quality of service, etc. Analogous factors exist in case of CPS such as a large manufacturing plant, transportation networks, power grid, etc. The data deluge caused by 24x7 sensing in CPS is making this situation worse: our ability to measure at finer spatial (e.g. component level) and temporal granularity keeps improving; however the CPS modeling methods lag behind (with a few notable exceptions [19]).

Challenge II: Online processing of high dimensional, low-quality data. Hunter et al. demonstrate the need to tackle massive amount of high-dimensional data with low information content collected by CPS sensors in real time [19]. With the rise of industrial big data [5], and smart infrastructure (cities, transportation, homes, etc.), this is now a challenge for CPS modeling across multiple domains. Tackling this challenge requires an inter-disciplinary approach that combines performance modeling and system building with designing novel algorithms. For instance, Hunter et al. decompose computation on GPS-data streams into a series of small batch computations using the D-Streams programming model [20] deployed over Spark [2]. This enables them to use an Expectation Maximization (EM) algorithm to estimate travel times on a network of roads from noisy GPS data in real time.

Opportunity I: Modeling switched systems. In Section 2 we noted that learning CPS models when the discrete state sequence is unknown is challenging. Ubiquitous sensing presents an exciting prospect: *are there measurements or variables, hidden inside the massive amount of sensing data, that we can use to accurately infer discrete state changes?* If yes, then this can simplify the task of learning models for arbitrarily switched systems. Though this intuition is obvious, finding the relevant subset of variables to infer discrete state changes can be challenging in practice for two reasons:(1) high dimension of sensor data, (2) noisy or irregularly sampled measurements. We could use feature selection [10] to tackle high dimensional data but it is challenging to do this in an unsupervised setting, i.e. when no labels or context information is available. Noisy and/or irregularly sampled measurements make multivariate change point detection techniques less suitable for this problem.

Opportunity II: new applications. The traditional goal of CPS modeling has been to predict some system output. This prediction can then be used for design verification, fault detection, etc. However, big data analytics combined with cloud based services enables new applications that require comparing data from two different systems or from the same system across two different operational runs. For instance, apart from predicting the current travel times, the data from the Mobile Millennium type projects can also to be used to compare traffic patterns across two different cities or at different times during the same city. One goal for such comparisons can be to learn *discriminative patterns* from data that distinguish one instance or dataset from another. This will enable novel applications in the CPS domain: (1) system-wide comparison – e.g. how does the traffic pattern during rush hour differ from the rest of the day? (2) *what-if* analysis – e.g. how will lane closures for repairs or due to accidents affect traffic patterns and travel times? (3) proactive maintenance – continuously collecting data from automobiles can enable proactive problem detection and maintenance scheduling [8].

5. RESEARCH DIRECTIONS

We outline machine learning and data mining techniques that can address the challenges and opportunities discussed in Section 4. These techniques are mainly from two areas: (1) graphical model structure learning, and (2) data mining algorithms for finding patterns and association rules in massive datasets. Several researchers are currently active in both the areas and these techniques are being applied to a diverse set of problems in medicine and biology (e.g. drug discovery, and disease diagnosis), analyzing social networks, and building new applications such as location-aware services.

Inferring input-output relationships. We can represent the input-output relationships in a CPS as a graph where nodes represent variables and an input-output relationship (more generally a dependency between variables) is represented as an edge. Hence, a simple CPS component can be thought of as a bipartite graph whereas a combination of components denoting a subsystem or the entire system can be represented as a tree, a DAG or a general graph with cycles. We can interpret these graphs as capturing either the dependency or correlations between variables or the knowledge that the state or value of a node (i.e. it's associated variable) depends only on its parents or neighbors.

The goal of graphical structure learning for CPS will be to infer such graphs using the monitoring data collected by CPS sensors. The complexity of this task depends on the nature of knowledge we want to extract and on the graph structure we want to learn. For instance, relevance or dependency networks only capture correlations between variables, and there exist efficient algorithms for inferring these networks from massive data [13]. However, they provide limited information about the true input-output relationships. In practice, they are used as a *quick and dirty* way for data exploration or for applications where correlations are informative [16]. More detailed information about the input-output relations can be obtained from the conditional independence relationships among the variables. Such relationships reveal not only that two variables are correlated, but also that the correlations can not be explained by any of the other measured variables. For example, on a hot day, there will be higher power consumption due to the use of AC, as well as a higher incidence of heat shock cases at local hospitals. A correlation based model will show an edge between power consumption and heat shock cases, as the two variables are correlated. Models based on conditional inde-

pendence on the other hand will not show an edge between these two variables because their correlation is explained by the temperature. The problem of inferring conditional independence relations from data has received significant attention in fields such as statistics and machine learning, and algorithms have been developed for binary, categorical, and time series data [13]; but applying these algorithms to the massive amount of CPS data can be computationally challenging. We believe that *graphical structure learning is a promising approach to infer input-output relationships in CPS*. As discussed in Section 4, a successful solution to this problem will enable us to apply existing hybrid system models to CPS monitoring data now being collected by pervasive low-cost sensors.

Inferring and predicting state switches. In Section 2 we noted that if the discrete state sequence (or switches between subsystems) are known for a CPS, then we can partition the measurement data based on the system's operating mode, and estimate the parameters for function f_s using only the data collected when the system is in state s. However, in general, these state changes are not known. This makes the problem of model identification for arbitrarily switched systems challenging.

With ubiquitous sensing, we can now collect data that can help us *infer* discrete state changes. For instance, the adaptive cruise control system in a car can be modeled as a combination of four states, cruising without obstacles in sight, cruising at a safe distance, decelerating because a car is in front, and emergency braking [1]. We can combine continuous measurements of speed, engine rpm, etc. with binary variable (cruise control on/off) to infer when a car is any of these states. GPS traces from driver's phone or car can also be utilized. A more challenging problem can be to predict the next state change. Such predictions can be useful for proactive maintenance and fault/anomaly detection [8]. However, combining massive amount of binary, categorical and continuous measurements to detect/predict state change points efficiently in real time is a challenging problem.

Patterns in CPS data. As discussed in Section 4, data mining techniques when applied on CPS big data can enable a new class of applications. For instance, in the *proactive management* application in Section 4, instead (or in addition to) comparing the measurements from a car against the predictions from a model, we can also compare *patterns* in its measurement data against data from other cars of same model. (This assumes a infrastructure for processing and archiving data from a large number of cars). Das et al. proposed a similar approach for anomaly detection in commercial aircrafts [8]. However, in addition to applications like maintenance scheduling, and detecting faults and anomalies, this *data driven* approach can also be applied to design, testing and verification as well as system-wide comparisons of CPS.

Data mining researchers have proposed efficient algorithms for learning frequent patterns, sequential patterns, and discriminative patterns [11] from symbolic data. Since CPS data is often in the form of time series, we can use techniques like temporal abstractions [7] to convert them to symbols and then mine for patterns or motifs in them. We believe that combining *pattern mining techniques* with traditional CPS modeling can help us extract actionable intelligence from CPS monitoring data that will lead to exciting new applications.

6. CONCLUSION

In this position paper we discussed the impact of "big" monitoring data collected from CPS using ubiquitous sensing with respect to CPS *modeling* and *analytics*. We discuss new challenges for CPS modeling, and opportunities related to CPS management and novel applications arising from big data. We also point out the key differences between CPS and IT systems that lead us to argue for further research into combining CPS modeling with analytics. Finally, we outline a few exciting research directions.

7. REFERENCES

[1] Adaptive Cruise Control System. http://www.globaldensoproducts.com/dcs/accs/.

[2] Spark: Lightning-Fast Cluster Computing. http://spark-project.org/.

[3] The Mobile Millennium Project. http://traffic.berkeley.edu.

[4] The Ptolemy Project. http://http://ptolemy.eecs.berkeley.edu/.

[5] The Rise of Industrial Big Data. GE whitepaper. http://www.ge-ip.com/library/detail/13170/.

[6] B. Selic. The pragmatics of model-driven development. *IEEE Software*, 20 (5), 2003.

[7] I. Batal, D. Fradkin, J. Harrison, F. Moerchen, and M. Hauskrecht. Mining Recent Temporal Patterns for Event Detection in Multivariate Time Series Data. In *Proceedings of the KDD*, 2012.

[8] S. Das, B. L. Matthews, A. N. Srivastava, and N. C. Oza. Multiple Kernel Learning for Heterogeneous Anomaly Detection: Algorithm and Aviation Safety Case Study. In *Proceedings of the KDD*, 2010.

[9] P. Derler, E. A. Lee, and A. S. Vincentelli. Modeling Cyber-Physical Systems. In *Proceedings of the IEEE, 100 (1)*, 2012.

[10] I. Guyon. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[11] J. Han and M. Kamber and J. Pei. *Data Mining: Concepts and Techniques, 3rd ed.* Morgan Kaufmann, 2011.

[12] J. Sztipanovtis and G. Karsai. Model-integrated computing. *IEEE Computer*, 34 (4), 1997.

[13] K. P. Murphy. *Machine Learning: a Probabilistic Perspective*. MIT Press, 2012.

[14] F. Lauer and G. Bloch. Switched and PieceWise Nonlinear Hybrid System Identification. In *Proceedings of HSCC*, 2008.

[15] V. L. Le, F. Lauer, L. Bako, and G. Bloch. Learning Nonlinear Hybrid Systems: From Sparse Optimization to Support Vector Regression. In *Proceedings of HSCC*, 2013.

[16] K. Nagaraj, C. Killian, and J. Neville. Structured Comparative Analysis of System Logs to Diagnose Performance Problems. In *Proceedings of the NSDI*, 2012.

[17] S. Paoletti, A. L. Juloski, G. Ferrari-trecate, and R. Vidal. Identification of hybrid systems: a tutorial.

[18] L. Sha, G. Gopalakrishnan, X. Liu, and Q. Wang. Cyber-Physical Systems: A New Frontier. In *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, 2008.

[19] T. Hunter and T. Das and M. Zaharia and P. Addeel and A. M. Bayen. Large Scale Estimation in Cyberphysical Systems using Streaming Data: a Case Study with Smartphone Traces. *arXiv.org*, 1212.3393v1, 2012.

[20] M. Zaharia, T. Das, H. Li, S. Shenker, and I. Stoica. Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters. In *Proceedings of HotCloud*, 2010.