An Algebra of Machine Learners with Applications

Nageswara S. V. Rao
Oak Ridge National Laboratory
Oak Ridge, TN 37831, USA
raons@ornl.gov

***Abstract*—Machine learning (ML) methods are increasingly being applied to solve complex, data-driven problems in diverse areas, by exploiting the physical laws derived from first principles such as thermal hydraulics and the abstract laws developed recently for data and computing infrastructures. These physical and abstract laws encapsulate, typically in compact algebraic forms, the critical knowledge that complements data-driven ML models. We present a unified perspective of these laws and ML methods using an abstract *algebra* $(\mathcal{A}; \oplus, \otimes)$, wherein the performance estimation and classification tasks are characterized by the *additive* $\oplus$ operations, and the diagnosis, reconstruction, and optimization tasks are characterized by the *difference* $\otimes$ operations. This abstraction provides ML codes and their performance characterizations that are transferable across different areas. We describe practical applications of these abstract operations using examples of throughput profile estimation tasks in data transport infrastructures, and power-level and sensor error estimation tasks in nuclear reactor systems.**

***Index Terms*—physical laws, abstract laws, machine learning, abstract algebra, reactor systems, data transport infrastructures.**

I. Introduction

Machine learning (ML) methods are increasingly being applied to solve complex, data-driven problems in diverse areas of interest including energy systems, and data and computing infrastructures. In the current "post black-box" period, various ML methods are being customized, sharpened, and composed in unprecedented ways to exploit the domain knowledge, including parameters and their relationships within the underlying system. Over the decades, physical laws have been developed, often in analytic forms, and exploited to customize ML solutions for reactor system analytics, methane hydrate exploration, and other applications. For example, a three-level fuser of linear and non-linear machine learners that reflects the structure of the coolant system of a nuclear reactor has been proposed for power level estimation using independent monitoring measurements [26]. In another direction, abstract laws were developed and incorporated into ML solutions for performance profiling of data and computing infrastructures; for example, the concave-convex profiles of data transport infrastructures and their individual sites, subsystems (e.g., file and IO), and components (e.g., TCP version) lead to the diagnosis of bottlenecks and optimization of throughput [32].

In general, physical and abstract laws encapsulate the critical knowledge — often in compact differential and algebraic forms — that complements datasets and ML models obtained from them. They have been synergistically exploited to compose powerful ML solutions, and also to derive generalization equations, for example, distribution-free error confidence bounds for sensor drift estimation in primary coolant systems [37] and concave-convex profile transitions in throughput profiles of data transport infrastructures [36]. In this paper, we unify these approaches under an abstract algebraic framework of ML methods and physical-abstract laws.

We propose an abstract *algebra* $(\mathcal{A}; \oplus, \otimes)$ of data-driven models wherein: (i) estimation and classification of performance profiles are characterized by *additive* $\oplus$ operations; and (ii) diagnosis and optimization are characterized by *difference* $\otimes$ operations. This abstraction provides codes and characterizations (subset of $\mathcal{A}$) transferable across multiple areas, including data and compute infrastructures, and thermal hydraulics, by exploiting the structure, relationships, and constraints from the physical and abstract laws.

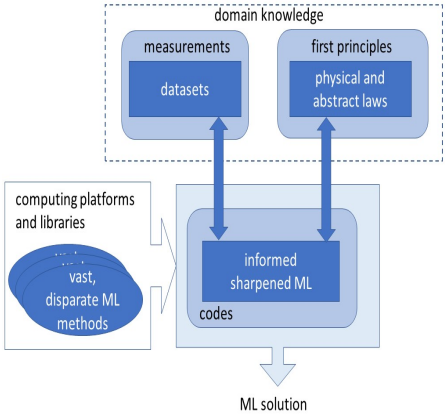
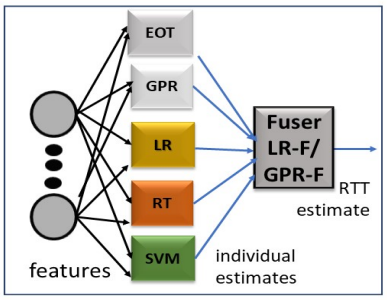


Fig. 1: *Algebra of ML models and physical-abstract laws for informed and sharpened solutions.*

We formulate ML problems that underlie these estimation and diagnosis tasks, and develop solutions by utilizing diverse ML methods combined with physical-abstract laws, as illustrated in Fig. 1. These ML solutions reflect the components and structure specified by the laws of the underlying system, and are composed to exploit the diversity of ML methods, namely, smooth and non-smooth, statistical and structural methods (Sections V and VI). We present examples of these abstract operations from exiting works on practical tasks in (i) data transport infrastructures [30] that are characterized by abstracts laws in terms of concave-convex throughput profiles, and (ii) power-level estimation of reactor systems using remote measurements of cooling towers of the secondary coolant system [26] and sensor error estimation of reactor's primary coolant systems



Regression Estimators:
EOT: Ensemble of Trees
GPR: Gaussian Process Regression
LR: Linear Regression
RT: Regression Trees
SVM: Support Vector Machine

Fusers:
LR-F: Linear Regression Fuser
GPR-F: Gaussian Process Regression Fuser

Fig. 2: *Fusion of diverse regression estimates improves the training and generalization error in estimation of RTT from MPI measurements [28]. Ensemble of trees and regression trees are non-smooth estimates; Gaussian process regression and support vector machines are smooth estimates; and they are fused using Gaussian process regression and linear regression methods.*

[37], both characterized by the underlying thermal hydraulic laws.

The organization of this paper is as follows. We describe the post-black-box efforts that lead to the proposed algebra in Section II. Basic concepts of ML-learnability and generalization equations are described in Section III. We develop an algebra that underlies the proposed proposed approach by building up on specific examples and use cases in Section IV. We present examples to illustrate the abstract additive algebraic operations in Section V, and the abstract difference operations in Section VI. Conclusions are presented in Section VII.

II. Post-Black-Box Era of Machine Learning: Origins of Algebra

Over the past decade, there has been a sea change in ML practice in part due to the availability of software products, including python libraries such as Numpy [5], Theano [10], Keras [2], and TensorFlow [9] and R modules such as CARET [1], randomForest, and nnet [4]. Furthermore, ML frameworks for special-purpose and high-performance computing systems, such as Nvidia's cuDNN [7] and NCCL [6], TensorFlow [9], Caffe2 [3], and Baidu's PaddlePaddle [8], demonstrate the unprecedented scale and scope of practical problems solved by ML methods [12], [21]. Consequently, such methods have been extensively applied to a broad set of applications; initially, however, their ease of use resulted in them being used as just "black boxes," often leading to solutions that were hard to interpret and explain. Currently, we are at the verge of a post-black-box era in which basic ML methods are being customized, sharpened, and composed in unprecedented ways to exploit domain knowledge, in particular, system and physical aspects. These developments in practical ML methods can be broadly classified into the following categories:

(a) *Strategic Deep Compositions:* The sharpened and customized ML solutions are composed in multilevel, tree and acyclic graph combinations, wherein ML methods at nodes are selected to match the underlying system components, for example, 3-level tree ML method for power-level estimation in nuclear reactors 4 [26].

(b) *Fusion of Diverse ML Methods*: Diverse ML methods, such as smooth kernels and support vector machines, non-smooth tree and forest estimates, and algebraic vector spaces, are combined to mitigate overfitting by ensuring diversity of design, for example, classifier fusers for dissolution events [27] and regression fusers for round trip time estimation using code execution time measurements [29].

(c) *Hyperparameter Harness*: ML solutions are wrapped inside a harness to explore, select, and tune higher-level parameters, for example, using an informed gradient search in the hyperparameter space.

These complex operations represent the beginnings of *an abstract algebra of ML* methods, as illustrated by the fusion of diverse classifiers and regression methods. Based on statistical classifier theory, there is no universally best classifier for finite samples [16]. Furthermore, there is no method to assert such property even if such classifier exists; the universality of k-Nearest Neighbor method (with dynamic k) in achieving Bayes error is valid only for infinite samples. These finite sample laws have a practical impact: newer ML classification methods continue to be developed using diverse methods, including, smooth and non-smooth, and structural and statistical. Indeed, ensemble methods take advantage of them by combining similar weak learners or disparate strong learners for improved performance, and they are abstracted by the "addition" operations over ML methods. For example, in estimating round trip time (RTT) from message passing interface (MPI) execution times of codes over wide-area networked nodes, the fusion of five regression estimates (shown in Fig. 2) improved the performance by 15% [29]. We show next that such abstract addition operations can be expanded beyond the simple aggregation of ML modules to reflect physical and abstract laws. Furthermore, additional tasks such as diagnosis and optimization can be abstracted as "difference" operations on ML methods and models. Together, these abstracted addition $\oplus$ and difference $\otimes$ operations on ML methods provide a higher-level algebraic structure that transcends the specific applications, as will be described subsequently in this paper.

We formulate the underlying ML models within the context of a system S characterized by (i) *law* $\mathcal{L}_S(P, Z)$ that relates the parameters and system variable vectors, P and Z, respectively, and (ii) *structure* $\mathcal{S}_S(\mathcal{C}_S)$ that specifies the relationships between its set of components $\mathcal{C}_S$.

III. ML-Learnability and Generalization

We now address the boundaries of what is possible under the ML paradigm based on the theory of generalization and computability and then add the additional expressibility requirements, as illustrated in Figure 3. We will then consider more practical limits of what can be inferred about ML solutions that have been trained using a finite sample.

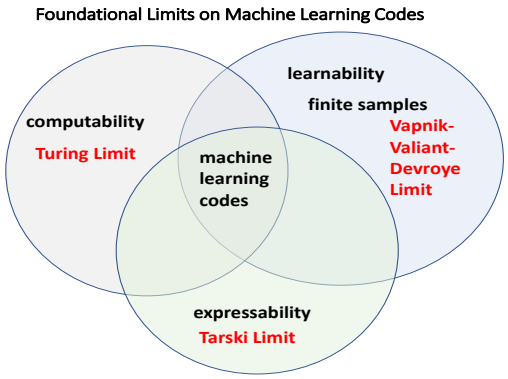


Fig. 3: *ML solutions must be computable and learnable and may also have to be expressible.*

A. Learnabiilty, Computability, and Expressibility

ML solutions that require performance guarantees and interpretable parameters are subject to three fundamental limits: learnability, computability, and expressibility. Learnability is defined by generalization properties such as finite scale-sensitive dimensions [11]. Computability in ML requires that the optimization problems needed for ML solutions be decidable in a Turing sense [17], [22]; for example, learning the equivalence of context-free grammars is undecidable. Expressibility of ML refers to the ability of learned models or parameters to represent critical domain properties, such as learning information that neural network weights capture feature correlations or hidden nodes represent critical regions of the image being processed.

Expressibility of ML outputs is critical for trusting black-box solutions, and they are subject to Tarski's limit [20], [38], in particular solutions that require that critical properties be defined first in complex algebraic systems. In simple terms, Tarski's undefinability states that in any sufficiently strong formal system, truth in that standard model cannot be defined within the system, which in turn implies that examples provided within the context of a system are not guaranteed to lead to the discovery of truths about that system. The undefinability theorem shows that no sufficiently rich interpreted language can represent its own semantics. A corollary is that any meta-language capable of expressing the semantics of some object language must have expressive power exceeding that of the object language. The metalanguage includes primitive notions, axioms, and rules absent from the object language, so there are theorems that are provable in the metalanguage that are not provable in the object language. Consequently, problems formulated to "learn" such meta-properties are beyond the power of ML methods.

It is particularly important to investigate these results as more complex ML problems are being formulated to discover complex, high-level relationships. When considering the broad spectrum of scenarios where ML solutions are needed, it is critical to identify the problems whose solutions are beyond the power of ML based on these limits. On first glance, these problems might appear too abstract, but it is instructive to note that several practical problems such as virus detection [15] and ensuring the resilience of complex codes [25] are undecidable, and hence their analogs are also beyond the power of ML solutions.

B. Finite Sample Inference of ML Quality

Even for problems that can be provably solved by ML methods, such as the classification problem, there are fundamental limits on how general such methods can be and what can be inferred about them based on their performance. This is because learning is conducted using a finite number of samples. The incompleteness or slow convergence theorem of Devroye [18] shows that there is *no universally best machine learning method* that can be asserted using a finite number of samples. In other words, ML methods trained on finite samples do not have sufficient information to optimize over the infinite class of data distributions. Indeed, any ML method is reduced to a random guesser for certain data distributions that are tailored to the method, even if it performs optimally on a given training dataset. As a practical consequence, diverse ML methods will continue to be developed as newer learning problems arise in various disciplines. These methods employ powerful techniques to exploit the structure $\mathcal{S}_S$ and distributions of domain-specific parameters in $\mathcal{L}_S(A, Z)$. Collectively, they provide a set of rich, diverse ML solutions, but their performance can vary widely even for a single problem. Information fusion methods are an attractive alternative. They combine multiple disparate ML solutions to ensure a theory of generalization that is at least as good as their best subset using the projective fusers that utilize the lower envelope of error surfaces [24]. Furthermore, these fusers can be retrained to incorporate newer methods using training samples, which gives us a practical solution to continually fold in the power of newer ML methods.

Another limitation of an ML solution is the computational intractability [19], even if the ML problem is Turing decidable, because training a three-node neural network is NP-Complete [14]. Thus, practical ML computations can only guarantee approximations. For example, different invocations of the back propagation algorithm yield different approximations in sigmoidal neural networks. A nearest neighbor fuser combines these estimates to provide a solution superior to all of them [24]. In general, for a fuser class $\mathcal{F}_F$ that combines estimators $\hat{f}_i \in \mathcal{F}_i, i = 1, 2, \ldots, N$ and satisfies the isolation property, namely, containing functions that transfer individual inputs to output [23], we have

$$I(f_F^*) = \min_i^N I(\hat{f}_i) - \Delta,$$

where $I(f)$ is the expected error of estimator f, and Δ is the improvement in the expected error achieved by the optimal

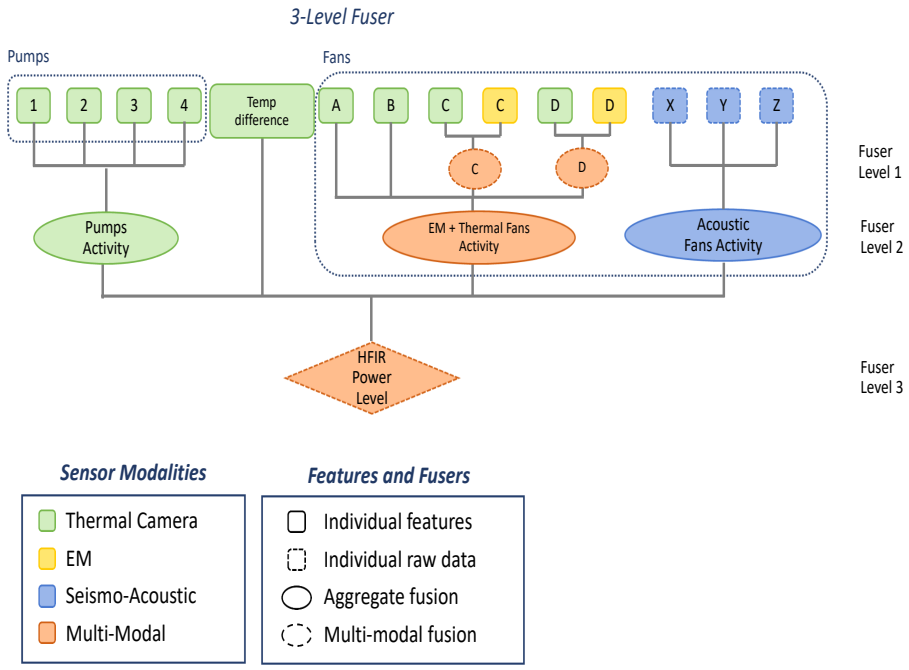


Fig. 4: *Composite ML method for reactor power estimation: tree structure derived from fluid dynamic equations is used to compose ML solution with linear and non-linear fuser nodes [26].*

fuser f_F^* over the best estimator. For the learned fuser $\hat{f}_F$ that minimizes the training error, the confidence function δ_V provides the probability that the error is within ϵ of the optimal. It is expressed in terms of $V = \sum_{i=1}^{N} V_i$, where V_i is the total variation of $\mathcal{F}_i$, and improves with the training sample size independent of the underlying distribution [24], which is complex and often unknown. These aggregation operations over ML models provide a basis for abstract additive operations $\oplus$ that are rigorously supported by such distribution-free guarantees [39].

IV. An Algebra of Machine Learners

The physical and abstract features, along with their relationships, captured by laws play a crucial role in the design and analysis of ML solutions to the underlying classification and estimation problems. By identifying generic features, we exploit the properties of ML model decision surfaces and laws to develop custom solutions. We focus on two overall areas, which are described next at a high-level and illustrated with practical examples in the following two sections.

(a) *Additive algebraic $\oplus$ operations—aggregation and information fusion:* The knowledge from physical and abstract laws can be combined to customize, sharpen, and compose solutions. (a) *Sharpening and customization:* Basic methods are fine-tuned by identifying and optimizing their structure and parameters by exploiting system and physical aspects. The sharpened and customized solutions can be composed in multilevel tree or acyclic graph combinations in which nodes are selected to match the underlying system components. (b) *Fusion of diverse sources and methods:* Diverse data (simulations and experiments) and methods (e.g., smooth kernels and support vector machines, non-smooth tree and forest estimates, algebraic vector spaces) are combined, and the solutions are wrapped inside a harness to explore, select, and tune higher level parameters indicated by the laws.

(b) *Difference algebraic $\otimes$ operations—reconstruction, diagnosis and optimization:* Physical and abstract laws that characterize ideal configurations and conditions (e.g., fully constructed 3D image, and optimized data transport infrastructure) are used to compute the differences with respect to ML models derived from measurements. (a) *Reconstruction and Diagnosis via differences:* Under ideal conditions, the differences will be negligible, but under operational conditions, they indicate incompleteness or errors, such as partial macro-level material models, sensor drifts in energy systems and under-performing of data transport (e.g., convex throughput profiles). (b) *Performance optimization parameters:* By relating the differences to the underlying systems laws, the parameters and components can be identified to support further steps (e.g., parameters for next experiments, re-calibrating sensors or fine-tuning file systems).

The physical and abstract laws capture the critical knowledge of the underlying systems, often in compact forms, by abstracting important parameters and relationships. Data sets collected from operational or emulated systems and ML models obtained from them typically provide complementary knowl-

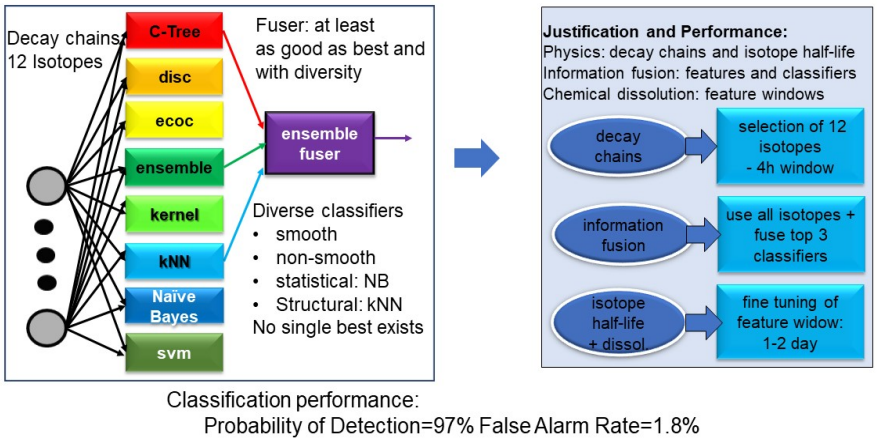


Fig. 5: *Classifier Fusion informed by underlying physics [27]. Fusion of diverse classifiers improves the training and generalization error [28]. Ensemble of trees and classification trees are non-smooth methods; kernel and support vector machines are smooth methods; k-nearest neighbor method is based on structural properties and naive Bayes classifier is based on statistical principles; and they are fused using ensemble of trees method.*

edge, often reflecting measurement uncertainties and errors and finite sample limits. They have been synergistically exploited to compose powerful ML solutions to estimation and diagnosis problems in a variety of ways, ranging from incorporating physical constraints, utilizing physical law violations as objective functions to minimize, structuring ML solutions to reflect physical laws, and combining multiple disparate estimators of parameters. In particular, the physical and abstract features, along with their relationships, captured by laws play a crucial role in the design and analysis of ML solutions to the underlying estimation and diagnosis problems in the two application domains, namely, reactor coolant systems and computing-data complexes. Broadly speaking, these solutions are based on utilizing the additive and differential knowledge between ML models and their counterpart first principle laws: solutions to estimation problems, including classification and regression problems, typically require the additive knowledge, and those to diagnosis problems typically require the differential knowledge. Furthermore, analytical properties such as bounded derivatives of smooth laws and bounded total variations of non-smooth laws combined with empirical errors of ML models lead to performance characterization and guarantees in the form of generalization equations with practical insights. We describe in next two sections such specific cases and identify the underlying concepts behind the abstract additive and difference operations within the applications context.

V. Additive Operations: Aggregation and Information Fusion

The addtive operation in RTT estimation problem discussed in Section II is a regression problem, and its solution involves fusing five disparate regression estimates shown in Figure 2. In particular, two smooth methods are Gaussian Process Regression (GPR) and Support Vector Machine (SVM) with Gaussian kernels, and two non-smooth estimators are Ensemble of Trees (EOT) and Regression Trees (RT), and the fuser is based on GPR method. Here, the individual regression estimates are chosen to represent their design diversity since the MPI execution time measurements have a complex relation to the RTT of connection over which they are collected and it is not clear if the underlying regression is smooth or non-smooth. We will discuss next the additive operations that are guided by the underlying physics laws or information.

1) Structural Aggregation Based on Physical Laws

When physical models or their sensitivity estimates are available, ML methods are customized using violations of physical laws as errors to be minimized [13], [35]. These methods have been developed for smooth and nonsmooth laws [34]. Recently, more complex ML solutions have been developed by exploiting the structure $\mathcal{S}_S$ of the law. A specific example in which the structure derived from fluid dynamic equations is used to compose ML solution involves the estimation of the power level of a reactor based on external monitoring sensors [26]. In this scenario, the activity of a reactor is monitored using measurements from four fans, four pumps, three acoustic sensors, and temperature sensors. The resultant a three-level ML solution is shown in Figure 4. The pumps, fans, and acoustic sensors collectively correspond to total power and hence are individually combined using the linear weighted method at level 2, and the outputs of different modalities are combined by training non-linear SVM at level 3. Thus, this ML solution is based on tree topology.

2) Physics-Informed Fusion of Classifiers

Inferring of dissolution events at a radiochemical processing facility using monitoring measurements is an important classification task [27]. For dissolutions scenarios, the classifier over-fitting is addressed by fusing three promising ones from eight disparate classifiers, as shown in Figure 5. These eight classifiers are chosen from four complementary classes, namely, smooth and non-smooth, and statistical and structural methods. smooth methods are SVM and kernel classifiers, and non-smooth methods are classification trees (C-trees) and EOT;

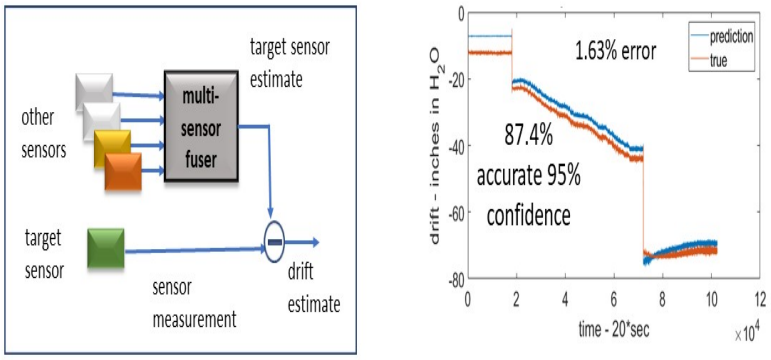


Fig. 6: *Difference between the measurement from a target sensor and its estimate obtained by fusing the measurements from other sensors is an estimate of the sensor error [37]*

statistical method is the Naive Bayes classifier, and structural method is the k Nearest Neightbor (kNN). from eight disparate classifiers chosen from four classes, namely, smooth and non-smooth and also statistical and structural Here, Ensemble of Trees Fuser (ETF) combines classifier outputs, thereby incorporating their design diversity. In addition, by using physics laws detection probability of 97% is achieved at a false alarm rate under 2% compared to 76% detection at 18% false alarm rate of original ETF solution. Specifically here, the physics-based decay chains provided the isotope features to be used as input to classifiers, and their half-life estimates provided feature window-size to achieve improved performance beyond the simple classifier fusion.

VI. Difference Operations Over Laws and Models

Physical and abstract laws that characterize ideal operating conditions (e.g., drift-free reactor dynamics and fully optimized data transport infrastructure) are used to compute the differentials with respect to ML models derived from measurements from operational, emulated, or simulated systems. Under ideal conditions, the differentials will be negligible, but under operational conditions, they indicate errors, such as sensor drifts in reactor monitoring sensors and under performance of data transport (e.g., convex throughput profiles). Furthermore, by relating the differentials to underlying systems laws, the components can be identified to support actions (e.g., recalibrating sensors or fine-tuning file systems) as described next.

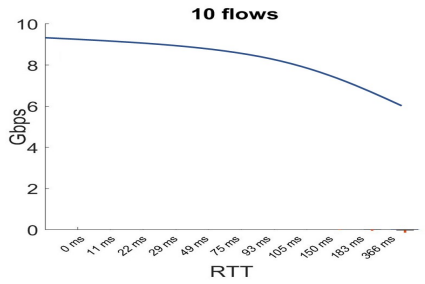
1) Diagnosis by Differences

The sensors of a power plant measure variables that are typically related to each other through the underlying physics laws. Such relationships provide regressions that are smooth or non-smooth with bounded variation property, which are conducive to estimation by machine learning methods [35]. These relationships are exploited to learn a regression function for a target sensor using measurements from other sensors that measure the same or different variables. The regression estimate is subtracted from its actual measurement to obtain a sensor drift estimate, as illustrated in Figure 6. The key idea is that when the target sensor measurement is subjected to drifts, this difference reflects its drift error [37]. The root mean square error of drift estimate is below 12.6% with 95% confidence, which are inferred by deriving the generalization equations for the error estimation [39]. The generalization equation derived in [37] does not impose any conditions on the underlying error distributions and yet provides practically useful information; the error and confidence estimates are derived by exploiting the large measurement data sets (typical of this domain) and domain-specific customization [37].

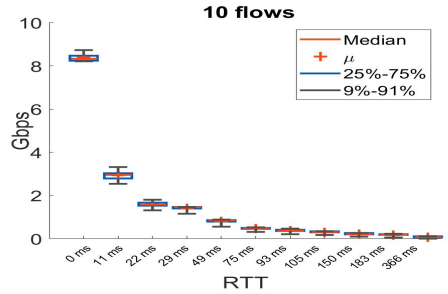
2) Performance Optimization

Network throughput measurements have been utilized to identify and isolate performance bottlenecks and provide ways to optimize the TCP parameters [33]. A mathematical method for characterizing transport performance that uses estimates of throughput profiles is the *concave-convex* analysis. Concave profiles indicate near-optimal throughput, and convex profiles indicate bottlenecks from factors such as IO buffers, file system throughput, mismatched IO-network couplings, and suboptimal protocols and transfer tools. The transport performance is characterized by the *throughput profile* $\Theta(\tau)$ (Fig. 7) as a function of RTT τ defined as $\bar{\Theta}_O(\tau) = E[\Theta_O(\tau)]$ where the underlying distribution is too complicated to calculate with simple models. Compared to the "physical laws" addressed in the previous section, law $\mathcal{L}_S(P, Z)$ is abstract in terms of its concave-convex property.

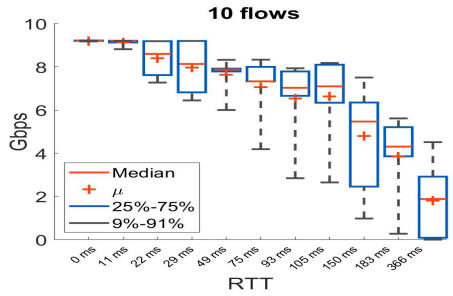
A data transport complex consists of special servers called data transfer nodes (DTN) that are optimized for high performance network, IO and file operations. They are connected over wide-area networks. DOE laboratory sites house DTNs connected over ESnet to support data transfers for several science projects. The throughput profiles can be analytically derived [31] and also estimated using ML methods from mea-



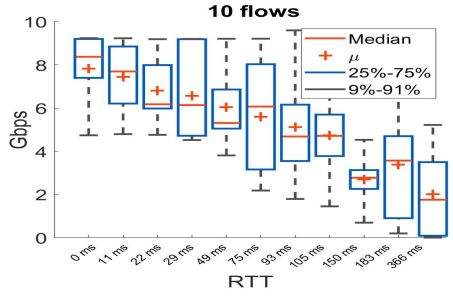
(a) ideal analytic profile $\Theta_I(\tau)$ from abstract law



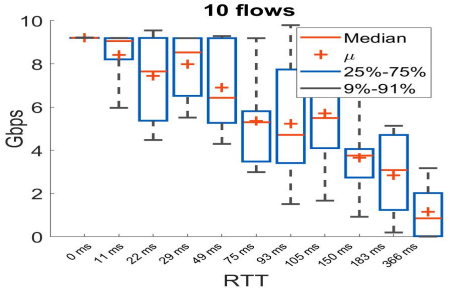
(b) estimated default profile $\hat{\Theta}_D(\tau)$



(c) estimated profile $\hat{\Theta}_C(\tau)$ for CUBIC



(d) estimated profile $\hat{\Theta}_S(\tau)$ for SCTP



(e) estimated profile $\hat{\Theta}_Y(\tau)$ for YEAH

Fig. 7: *Different infrastructure TCP throughput profiles.*

surements [29] for infrastructures with different RTT values between DTNs of the infrastructures as shown in Fig. 7 based on testbed measurements. For an infrastructure with 11 RTTs in 0-366ms range, an ideal throughput profile $\Theta_I(\tau)$ can be derived in closed algebraic form shown in Fig. 7(a). The concave profiles have been confirmed using experimental testbeds that employ highly optimized DTNs, and larger TCP buffers and more parallel streams have been shown to expand the concave regions of the profiles. We estimate profiles using testbed measurements, which entail random quantities with joint distribution dependent on complex factors like capacities of edge systems and networks, loads on files systems, transfer software, host systems, and connections. The profile estimate under default host parameters is shown in Fig. 7(b) for CUBIC TCP version; it has an overall convex profile and the throughput rapidly drops as RTT is increased. After optimizing host transport parameters, the profile for CUBIC improves as shown in Fig. 7(c). It has an overall concave profile and throughput is sustained compared to default case as RTT is increased, and throughput measurements have significant variations due to non-linear TCP dynamics but an overall slower decreasing trend. Throughput profile estimates for Scalable TCP (SCTP) and YEt Another Highspeed (YEAH) TCP versions are shown in Fig. 7(d) and (e), respectively, and they show overall trends similat to CUBIC. The difference between ideal analytical model and estimated ML model, namely, $\Theta_I(\tau) - \hat{\Theta}_A(\tau)$, $A = C, D, S, Y$ indicates the optimization level of data transport and can be used to decide TCP parameters and versions for the infrastructure.

Furthermore, performance bottlenecks such as under-tuned TCP parameters and limits lead to convex profiles as shown in Fig. 7(b) for untuned TCP parameters. Then, under performance of the infrastructure is diagnosed by estimating profiles from measurements using custom ML methods [36] and using them to obtain their difference from best possible profiles indicated by analytical models [31]. Infrastructure profiles can also be learned for individual sites and sub-systems such as DTNs, file systems and TCP version and stack [30]. Their deviation can be used to identify candidates for performance improvement and optimization. In one example, they can be used to identify sites with insufficient buffers, which can be increased to improve the infrastructure throughput performance. In another example, throughput profiles of the infrastructure can be assessed for different TCP versions.

VII. Conclusions

We presented an abstract algebra of data-driven models and physical and abstract laws that provides a unified view of recent approaches that combine disparate ML methods to efficiently solve complex problems from diverse areas such as nuclear systems and cyber infrastructures. We described practical scenarios where the performance estimation and classification tasks are characterized by abstract additive operations, and the diagnosis, reconstruction, and optimization tasks are characterized by abstract difference operations. This abstraction enabled us to develop ML codes and their performance characterizations that are transferable across multiple seemingly unrelated areas, such

as data and compute infrastructures, thermal hydraulics and material science areas, by exploiting the structure, relationships, and constraints from the physical and abstract laws. Specific examples of data transport infrastructures, power-level and sensor error estimation of reactor systems are described, wherein effective ML codes are developed.

Several future research and development directions can be pursued in applying these abstract operations to practical problems by exploiting the underlying physical and abstract laws. These results are early indicators that the knowledge of physical and abstract laws can utilized to develop efficient and effective ML solutions together with analytic performance characterizations for a broad class of ML problems. Specific future directions include the following:

- expanded problem space to include more detailed physical and other sensor information;
- explainable solutions wherein ML solutions reflect the structure of underlying laws; and
- integrated platforms wherein analytics equations and proofs are co-developed with codes under a single platform such as Jupyter Notebook.

Acknowledgments

This work is funded by Cyber Physical Networks and RAMSES projects, Office of Advanced Computing Research, U.S. Department of Energy. It utilizes published results form other projects, and is performed at Oak Ridge National Laboratory managed by UT-Battelle, LLC for U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

REFERENCES

[1] Caret. http://topepo.github.io/caret/index.html, 2018.

[2] Keras. https://keras.io/, 2018.

[3] A new lightweight, modular, and scalable deep learning framework. https://caffe2.ai/, 2018.

[4] Nnet. https://cran.r-project.org/web/packages/nnet/nnet.pdf, 2018.

[5] Numpy. http://www.numpy.org/, 2018.

[6] Nvidia collective communications library (nccl). https://developer.nvidia.com/cudnn/, 2018.

[7] Nvidia cudnn. https://developer.nvidia.com/cudnn/, 2018.

[8] Paddlepaddle: An easy-to-use, efficient, flexible and scalable deep learning platform. https://github.com/PaddlePaddle/Paddle, 2018.

[9] Tensorflow. https://www.tensorflow.org/, 2018.

[10] Theano. http://www.deeplearning.net/software/theano/, 2018.

[11] M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[12] A. A. Awan, H. Subramoni, and D. K. Panda. An in-depth performance characterization of cpu-and gpu-based dnn trainingon modern architectures. In *Proceedings of the Machine Learning on HPC Environments(MLHPC'17)*, 2017.

[13] J. Barhen and N. S. V. Rao. Information fusion method for system identification based on sensitivity analysis. In *Int. Conf. on Information Fusion*. 2000.

[14] A. L. Blum and R. L. Rivest. Training a 3-node neural network is NP-complete. *Neural Networks*, 5:117–127, 1992.

[15] F. B. Cohen. Computational aspects of computer virus. *Computer & Security*, 8:325–344, 1989.

[16] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGraw-Hill Book Co., New York, 1990.

[17] M. D. Davies and E. J. Weyuker. *Computability, Complexity, and Languages*. Academic Press, Inc, 1983.

[18] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, 1996.

[19] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., San Francisco, 1979.

[20] J. Murzi. The inexpressibility of validity. *Analysis*, 74(1):65–81, 2014.

[21] D. K. Panda. Scalable and distributed dnn training on modern hpc systems: Challenges and solutions. In *High Performance Machine Learning Workshop at CCGrid2019*, 2019. Keynote Talk.

[22] C. H. Papadimitriou. *Computational Complexity*. Addition-Wesley Publishing Co., 1994.

[23] N. S. V. Rao. On fusers that perform better than best sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):904–909, 2001.

[24] N. S. V. Rao. Measurement-based statistical fusion methods for distributed sensor networks. In S. S. Iyengar and R. R. Brooks, editors, *Distributed Sensor Networks*. Chapman and Hall/CRC Publishers, 2011. 2nd Edition.

[25] N. S. V. Rao. On undecidability aspects of resilient computations and implications to exascale. In *Resilience 2014: Seventh Workshop on Resiliency in High Performance Computing with Clouds, Grids, and Clusters*, 2014.

[26] N. S. V. Rao, C. Greulich, S. Sen, K. Dayman, J. Hite, W. Ray, R. Hale, A. Nicholson, J. Honson, M. R. Chatin, K. M. Buckley, R. D. Hunley, J. Johnson, Haley H. Hesse, M. Maceira, C. Chai, O. Marcillo, T. Karnowski, and R. Wetherington. Reactor power level estimation by fusing multi-modal sensor measurements. In *International Conference on Inforamation Fusion*, 2020.

[27] N. S. V. Rao, C. Greulich, S. Sen, J. Hite, K. J. Dayman, A. D. Nicholson, D. E. Archer, M. J. Willis, I. Garishvili R. D. Hunley, J. Johnson, A. J. Rowe, I. R. Stewart, and J. M. Ghawaly. Classification of dissolution events using fusion of effluents measurements and classifiers. In *Institute of Nuclear Materials Management Annual Meeting*, 2020.

[28] N. S. V. Rao, N. Imam, Z. Liu, R. Kettimuthu, and I. Foster. Machine learning methods for connection rtt and loss rate estimation using mpi measurements under random losses. In *International Conference on Machine Learning for Networking (MLN'2019)*, 2019.

[29] N. S. V. Rao, N. Imam, Z. Liu, R. Kettimuthu, and I. Foster. Machine learning methods for connection rtt and loss rate estimation using mpi measurements under random losses. In E. Renault and S. Boumerdassi, editors, *Machine Learning for Networking*. Springer, 2020.

[30] N. S. V. Rao, Q. Liu, Z. Liu, R. Kettimuthu, and I. Foster. Throughput analytics of data transfer infrastructures. In *13th EAI International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2018)*, 2018.

[31] N. S. V. Rao, Q. Liu, S. Sen, J. Henley, I. T. Foster, R. Kettimuthu, D. Towsley, and G. Vardoyan. TCP throughput profiles using measurements over dedicated connections. In *ACM Symposium on High-Performance Parallel and Distributed Computing*, Washington, DC, July-August. 2017.

[32] N. S. V. Rao, Q. Liu, S. Sen, Z. Liu, and R. Kettimuthu. Measurements and analytics of wide-area file transfers over dedicated connections. In *International Conference on Distributed Computing and Networking*, 2018.

[33] N. S. V. Rao, Q. Liu, S. Sen, D. Towsley, G. Vardoyan, I. T. Foster, and R. Kettimuthu. Experiments and analyses of data transfers over wide-area dedicated connections. In *26th International Conference on Computer Communications and Network*, 2017.

[34] N. S. V. Rao, D. B. Reister, and J. Barhen. Fusion method for physical systems based on physical laws. In *Proceedings of 3rd International Conference on Information Fusion*, 2000.

[35] N. S. V. Rao, D. B. Reister, and J. Barhen. Information fusion methods based on physical laws. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):66–77, 2005.

[36] N. S. V. Rao, S. Sen, Z. Liu, R. Kettimuthu, and I. Foster. Learning concave-convex profiles of data transport over dedicated connections. In E. Renault, P. Muhlethaler, and S. Bourmerdassi, editors, *Machine Learning for Networking*. Lecture Notes in Computer Science 11407, Springer, 2019.

[37] N. S.V. Rao, C. Greulich, P. Ramuhalli, A. Gurgen, F. Zhang, and S. M. Cetiner. Estimation of sensor measurement errors in reactor coolant systems using multi-sensor fusion. *Nuclear Engineering and Design*, 375:111024, 2021.

[38] A. Tarski. *Logic, Semntics, Metamathematics*. Oxford University Press, 1956.

[39] V. N. Vapnik. *Statistical Learning Theory*. John-Wiley and Sons, New York, New York, 1998.