EasyChair Preprint

№ 10528

Study of Overfitting by Machine Learning Methods Using Generalization Equations

Nageswara Rao

July 10, 2023

Study of Overfitting by Machine Learning Methods Using Generalization Equations

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

***Abstract*—The training error of Machine Learning (ML) methods has been extensively used for performance assessment, and its low values have been used as a main justification for complex methods such as estimator fusion and ensembles, and hyper parameter tuning. We present two practical cases where independent tests indicate that the low training error is more of a reflection of over-fitting rather than the generalization ability. We derive a generic form of the generalization equations that separates the training error terms of ML methods from their epistemic terms that correspond to approximation and learnability properties. It provides a framework to separately account for both terms to ensure an overall high generalization performance. For regression estimation tasks, we derive conditions for performance enhancements achieved by hyper parameter tuning, and fusion and ensemble methods over their constituent methods. We present experimental measurements and ML estimates that illustrate the analytical results for the throughput profile estimation of a data transport infrastructure.**

***Index Terms*—machine learning, over-fitting, hyper-parameter tuning, fusion and ensemble, generalization bounds, regression, throughput profile**

I. Introduction

Sophisticated Machine Learning (ML) methods have been developed to solve a variety of complex problems typically by minimizing the training error in various forms such as cross-validation, regularization, and augmented by information criteria. In addition to ML methods the employ models with a large number of parameters, approaches based on hyper-parameter tuning, ensembles and fusion of similar and disparate ML building blocks have been shown to dramatically reduce the training error. These approaches essentially increase the number of parameters compared to their constituent methods, which often increases the model space and results in the reduction of training error due to better fit to training data. Indeed, such lowering of training error has been a main justification for selecting the complex methods such as deep neural networks and ensemble methods with hyper parameter tuning, which utilize additional computations compared to their basic versions.

The ML methods with an expanded model space are known to over-fit the training data, particularly, when using a large number of parameters on small data sets. Consequently, their lower training error does not necessarily translate into their improved generalization which is essential for data not included in training. Currently, there are very few available methods that reliably identify over-fitting. Indeed, they often require carefully collected additional measurements to estimate the generalization error, which is not practical or viable in several cases. A complementary approach is to utilize the ML generalization equations [1]–[3] that utilize additional knowledge such as smoothness or finite total variation of the underlying quantities being estimated. The generalization equations capture the approximation and learnability properties of the underlying systems that are not usually represented by training data alone. In addition, they may also exploit the smoothness and algebraic properties of the parameters [4], for example, using thermal hydraulic equations of coolant systems [5] and concave-convex profiles of data transfer networks [6].

In this paper, we develop the approach of utilizing the generalization equations of hyper parameter turning, and estimator ensemble and fusion methods, for analyzing over-fitting in regression estimation problems. We present practical cases where independent tests show that the training error is more of a reflection of over-fitting than the generalization ability. In the first case, the throughput regressions of a data transport networks are estimated, and in the second case a low level radiation source is detected using regression and classification methods.

We derive the generalization equations based on training error of ML methods combined with their approximation and learnability properties that more accurately capture different contributions to generalization performance. We consider a generic form of the generalization equation wherein the expected error $I(\tilde{f})$ of the estimator $\tilde{f}$ chosen from a class $\mathcal{F}$ based on training sample of size l is below the precision parameter γ with confidence probability $1 - \delta(\gamma, l)$, expressed in the compact form

$$P\left\{I(\tilde{f}) < \gamma\right\} > 1 - \delta_{\mathcal{F}}(\epsilon, l), \tag{1}$$

when $\mathcal{F}$ satisfies suitable learnability and approximation properties [3]. This is best type of guarantee that can be given for

This research is sponsored in part by RAMSES project of Advanced Scientific Computing Research program, U.S. Department of Energy, and in part by the Office of Basic Energy Sciences, Division of Materials Sciences and Engineering, U.S. Department of Energy, 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. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

finite l when no restrictions are imposed on the probability distribution of the data, since it is not possible to achieve zero values for γ or δ. Ideally, $\tilde{f}$ that minimizes the training error based l-sample would satisfy the above generalization condition, but it is not possible in general based on training alone. For a deeper analysis, we establish the following more general version

$$P\left\{I(\tilde{f}) - I(f^*) < \epsilon + \hat{\epsilon}\right\} > 1 - \delta_{\mathcal{F}}(\epsilon, l), \qquad (2)$$

where $\hat{\epsilon}$ is the training error of $\tilde{f}$ and f^* minimizes the expected error over $\mathcal{F}$. In typical ML scenarios the training error $\hat{\epsilon}$ is used as a main criterion for choosing $\tilde{f}$, and Eq. (2) enables us to additionally account for the approximation error using $I(f^*)$ and the learnability using $\delta_{\mathcal{F}}(\epsilon, l)$.

Complex ML methods typically utilize a large class $\mathcal{F}$ to lower both $\hat{\epsilon}$ and $I(f^*)$, as a result of minimization of empirical and expected errors, respectively, over a larger set of functions. However, these larger classes also increase the learning parameters that in turn increase $\delta_{\mathcal{F}}$, which negatively effects the confidence in generalization. In particular, both hyper parameter tuning and fusion and ensemble methods increase $\mathcal{F}$, and do not necessarily achieve better generalization even if the training error is low. We present explicit cases that clearly illustrate this phenomenon using experimental measurements and a generic form $\delta(\epsilon, l) = Ae^{-B\epsilon^2 l}$ that several ML methods satisfy. We derive the conditions for a superior performance of hyper parameter tuning, and fusion and ensemble methods over their constituent methods for the regression estimation problem.

The organization of this paper is as follows. A generic decomposition of the generalization equations is derived in Section II. The two application scenarios are described in Section III. The generalization equations and measurements for hyper parameter tuning are described in Section IV. The fusers and ensembles of ML estimates are described for regression methods in Section V. Conclusions and future directions are presented in Section VI.

II. Generalization Equations

Under the ML paradigm, a machine "learns" a functional relationship between two vector random variables [7] using a random sample $(X_1, Y_1), (X_2, Y_2), \ldots, (X_l, Y_l)$, typically drawn from an unknown joint distribution $\mathbb{P}_{X,Y}$ of feature vector X and output vector Y. We consider that X_i's take bounded real values and Y_i takes bounded real values for the regression problems and Boolean values for the classification problem. The task is to learn a predictor function f from a complex class $\mathcal{F}$ such that $f(X)$ is a good estimate of Y overall. A fundamental generalization result establishes that, under certain conditions, the best estimator f^* that minimizes the expected cost

$$I(f) = E\left[Q(f(X), Y)\right] = \int Q(f(X), Y) d\mathbb{P}_{X,Y}$$

where $Q(.)$ is a cost function, can be closely approximated with high probability by an estimate $\tilde{f}$ learned solely from the sample regardless of the complexity of $\mathbb{P}_{X,Y}$ [3]. For regression problem $Q(f(X), Y) = (f(X) - Y)^2$ and for classification problem $Q(f(X), Y) = f(X) \oplus Y$ where $\oplus$ is the exclusive-OR operation. Vapnik's generalization theory [2], [3] establishes that a "suitable" estimator, $\tilde{f}$, computed by an ML method M, ensures

$$\mathbb{P}^l_{X,Y}\left[I(\tilde{f}) - I(f^*) < \epsilon + \hat{\epsilon}\right] > 1 - \hat{\delta}_{\mathcal{F}_M}(\epsilon, \hat{\epsilon}, l) \qquad (3),$$

where $\mathcal{F}_M$ is its function class, $\epsilon > 0$ is the *precision* parameter, $0 < 1 - \delta_{\mathcal{F}_M}(.) < 1$ is the *confidence* function, and $\hat{\epsilon}$ is the *training error* associated with computing $\tilde{f}$. This condition ensures that "error" of $\tilde{f}$ is within $\epsilon + \hat{\epsilon}$ of optimal error (of f^*) with probability $1 - \delta_{\mathcal{F}_M}$, *irrespective* of the underlying measured, computed or completely unknown data distribution $\mathbb{P}^l_{X,Y}$. Furthermore, under these conditions, the confidence parameter $1 - \hat{\delta}_{\mathcal{F}_M}(\epsilon, \hat{\epsilon}, l)$ approaches 1 as the sample size l approaches infinity.

The joint distribution $\mathbb{P}_{X,Y}$ of data is complex, domain specific, and is only partially known in most cases. In our context, it depends on the finer details of the underlying software and hardware components, which may manifest as additional random variables. Typically, in ML scenarios, we only have training error $\tilde{\epsilon}$ which is used as a main criterion for choosing $\tilde{f}$. Then, to satisfy the more stringent criterion in Eq (1) two conditions have to be met: (i) $\hat{\epsilon} = 0$, which is not always possible since it requires a globally optimal learning algorithm but can be verified in principle, and (ii) $I(f^*) = 0$, which requires a suitably dense $\mathcal{F}$ but cannot be verified when the underlying distribution is not known. For simplicity of presentation, we use the following equivalent form of Eq. (3)

$$\mathbb{P}^l_{X,Y}\left[I(\tilde{f}) - I(f^*) > \epsilon + \hat{\epsilon}\right] < \hat{\delta}_{\mathcal{F}_M}(\epsilon, \hat{\epsilon}, l), \qquad (4)$$

It constitutes one of the most critical characterization of the generalization of ML method, namely, distribution-free guarantees on their prediction performance on future data. Over past decades, generalizations of this formulation have led to finite sample performance guarantees in a variety of methods and applications, including neural networks [8], regression trees [9], Support Vector Machines (SVM) [3], and kernel estimates [10].

We derive a version of Eq (4) in Section II-B that separates the effects of training error $\hat{\epsilon}$, and the approximation and learnability properties of the function class $\mathcal{F}$ reflected by $I(f^*)$ and $\hat{\delta}_{\mathcal{F}_M}$, respectively. Specifically, $\hat{\delta}_{\mathcal{F}_M} = \delta_{\mathcal{F}_M}$ does not depend on $\hat{\epsilon}$ as illustrated in examples described next.

A. Confidence Function Examples

If $\mathcal{F}_M$ has finite capacity [3], then under bounded error and for a sufficiently large sample, the condition in Eq (4) is guaranteed; a more general result ensures this condition under finite scale-sensitive dimensions [8]. For sigmoid neural networks, the sample size needed to ensure Eq (4) is linear in the number of neural network parameters, as opposed to having the quadratic dependence of previous bounds for unbounded weights [11]. In particular, the estimate is

$$\hat{\delta}_{NN} = 8\left(\frac{32W}{\epsilon}\right)^{h(d+2)} e^{-\epsilon^2 l/512}$$

for a sigmoid network with h hidden nodes and input dimension d with weights suitably bounded by W. It is important to note that several statistical estimates learned by current ML methods are smooth, including SVM with Gaussian kernels [10] and

radial basis functions [12], and several variables and parameters in practical applications are bounded which enables us to develop generalization equations. ML methods also employ non-smooth methods, such as ensemble tree methods [13], regression trees [9], Haar estimators, and Nadaraya-Watson estimators. In practice, the parameters are bounded, and the learned functions have a finite (often small) number of jumps, which leads to their bounded finite total variation $V < \infty$. In this case, we have

$$\hat{\delta}_V = 8h\left(1 + \frac{128V}{\epsilon}\right)e^{-\epsilon^2 l/2048},$$

for a suitable function h [8].

B. Generic Form of Generalization Equations

The underlying principle behind the generalization equation Eq. (4) is based on the uniform convergence of empirical measures to expectations [3]. Consider a class of real valued functions $\mathcal{G}$ of the form $g(X,Y)$ whose expectation is

$$E(g) = \int_{X,Y} g(X,Y)\mathbb{P}_{X,Y},$$

where $\mathbb{P}_{X,Y}$ is the joint distribution of X and Y. The empirical mean of g based on a iid sample $(X_1,Y_1),(X_2,Y_2),\ldots,(X_l,Y_l)$ is given by

$$\hat{E}(g) = \frac{1}{l}\sum_{i=1}^{l} g(X_i,Y_i).$$

Under certain boundedness conditions, typically satisfied by ML methods, the uniform convergence property is specified as

$$\mathbb{P}^l_{X,Y}\left\{\sup_{g\in\mathcal{G}}|E(g)-\hat{E}(g)| > \epsilon/2\right\} < \delta_{\mathcal{G}}(\epsilon,l).$$

Let g^* and $\hat{g}$ minimize the expectation and empirical mean, respectively, such that $E(g^*) = \min_{g\in\mathcal{G}} E(g)$ and $\hat{E}(\hat{g}) = \min_{g\in\mathcal{G}} \hat{E}(g)$. Let the estimator $\tilde{g}$ have empirical error $\hat{\epsilon}$ such that $\hat{E}(\tilde{g}) = \hat{E}(\hat{g}) + \hat{\epsilon}$. Then, with probability $1-\delta_{\mathcal{G}}(\epsilon,l)$, we have

$$\begin{aligned} E(\tilde{g}) &< \hat{E}(\tilde{g}) + \epsilon/2 \\ &= \hat{E}(\hat{g}) + \epsilon/2 + \hat{\epsilon} \\ &< \hat{E}(g^*) + \epsilon/2 + \hat{\epsilon} \\ &< E(g^*) + \epsilon + \hat{\epsilon}, \end{aligned}$$

which in turn implies

$$\mathbb{P}^l_{X,Y}\left\{E(\tilde{g}) - E(g^*) < \epsilon + \hat{\epsilon}\right\} > 1 - \delta_{\mathcal{G}}(\epsilon,l).$$

By using $g(X,Y) = Q(f(X),Y)$ we obtain the following version of Eq (1)

$$\mathbb{P}^l_{X,Y}\left[I(\tilde{f}) - I(f^*) < \epsilon + \hat{\epsilon}\right] > 1 - \delta_{\mathcal{F}_M}(\epsilon,l),$$

where the right hand side does not depend on $\hat{\epsilon}$ and only on $\mathcal{F}_M$'s learnability properties. This confidence bound on $I(\tilde{f})$ is expressed as

$$\mathbb{P}^l_{X,Y}\left[I(\tilde{f}) > \epsilon\prime + \hat{\epsilon}\right] < \delta_{\mathcal{F}_M}(|\epsilon\prime - I(f^*)|,l), \qquad (1)$$

where $\epsilon\prime = \epsilon + I(f^*)$ is a precision parameter. The left hand side is entirely controlled by the ML method applied to random data of possibly unknown form, e.g., measurement and modeling errors of thermal hydraulics and computing loads, and physics of quantum channels. The right hand slide is entirely epistemic with two opposing effects: (i) $\mathcal{F}_M$ needs to be large to provide better approximation with lower $I(f^*)$ and hence higher confidence, and (ii) $\mathcal{F}_M$ needs to be small to provide better learnability, e.g., lower Vapnik dimension, with higher confidence. In effect, the larger $\mathcal{F}_M$ of a complex ML solution presents a trade-off: better precision due to smaller $\hat{\epsilon}$ and $I(f^*)$ versus lower confidence $1-\delta_{\mathcal{F}_M}$.

By using $\gamma = I(f^*) + \epsilon + \hat{\epsilon}$, this equation is rewritten as

$$\mathbb{P}^l_{X,Y}\left[I(\hat{f}) > \gamma\right] < \delta_{\mathcal{F}_M}(\gamma - \hat{\epsilon} - I(f^*),l).$$

Thus the effects of both $\hat{\epsilon}$ and $I(f^*)$, is to shift the $\delta_{\mathcal{F}_M}(.)$ to right by either quantity. Since it is a decreasing function of γ, their non-negative values lead to higher values of $\delta_{\mathcal{F}_M}(.)$, and hence lower confidence. Thus, learning algorithms with higher empirical error $\hat{\epsilon}$ and higher approximation error $I(f^*)$ both result in higher $\delta_{\mathcal{F}_M}$ and hence lower confidence. We use one of the common form $\delta_{\mathcal{F}}(\epsilon,l) = Ae^{-B\epsilon^2 l}$ for our illustrations, wherein lower values of A and higher values of B, ϵ and l all lead to higher confidence probability. In particular, it is a monotonically decreasing, differentiable function of ϵ with $D_{\mathcal{F}}(\epsilon) = \frac{\partial\delta_{\mathcal{F}}}{\partial\epsilon} = -2AB\epsilon l e^{-B\epsilon^2 l}$.

C. Regression Problem

In a generic regression estimation problem the feature vector $X \in \Re^d$ and the output vector $Y \in \Re$, the *expected error* of a regression function f is

$$I(f) = \int (f(X)-Y)^2\, d\mathbb{P}_{X,Y}.$$

The *expected best* regression estimator f^* minimizes $I(.)$ over $\mathcal{F}$, i.e., $I(f^*) = \min_{f\in\mathcal{F}} I(f)$. The *empirical error* $\hat{I}(f)$ based on training data $(X_i,Y_i), i = 1,2,\ldots,l$, is defined as

$$\hat{I}(f) = \frac{1}{l}\sum_{i=1}^{l}(f(X_i)-Y_i)^2$$

It is an approximation of $I(f)$ computed based on the training data. The *empirical best* regression estimator $\hat{f}$ minimizes $\hat{I}(.)$ over $\mathcal{F}$, i.e., $\hat{I}(\tilde{f}) = \min_{f\in\mathcal{F}} \hat{I}(f)$. For the *learned* regression estimator $\tilde{f}$, we have $\hat{I}(\tilde{f}) = \hat{I}(\hat{f}) + \hat{\epsilon}$.

III. Two Application Scenarios

We consider two different applications scenarios, namely, the throughput estimation of data transport network infrastructures formulated as a regression estimation problem, and the detection of low level radiation sources formulated as a classification problem. In both cases, measurements from structured experiments provide: (i) training data to estimate the underlying regression or classification function, and (ii) test data from additional independent experiments that enable the estimation of test error that more accurately reflects the generalization property than the training error.

A. Throughput Estimation of Data Transport Infrastructure

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. Network throughput measurements have been utilized to identify and isolate performance bottlenecks and provide ways to optimize the Transmission Control Protocol

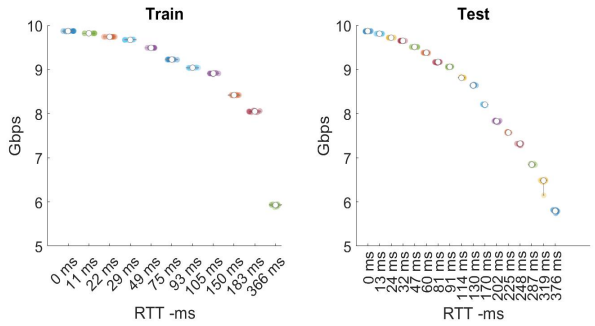
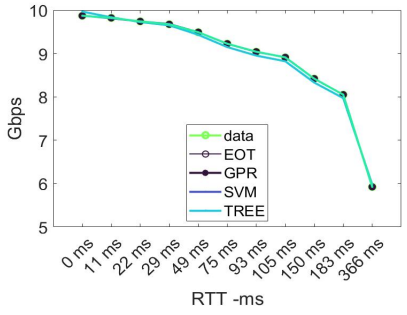
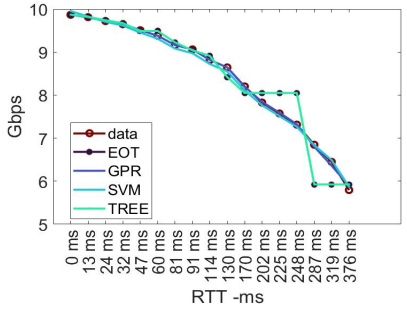


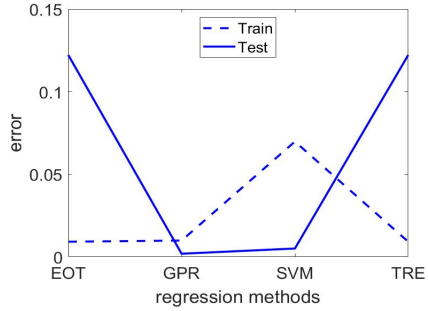
Fig. 1: Training and testing throughput measurements.



(a) *trained profiles*



(b) *predicted profiles*



(c) *train and test errors*

Fig. 2: Training and testing profiles and errors of EOT, GPR, SVM and TREE methods.

(TCP) parameters. The transport performance is characterized by the *throughput profile* as a function of connection Round Trip Time (RTT) τ. The throughput profiles can be analytically derived and also estimated using ML methods from measurements [6] for infrastructures with different RTT values between DTNs. For an infrastructure with 11 RTTs in 0-366ms range, and 17 RTTs in 0-376ms range, the throughput measurements are shown in Fig. 1, which have a smooth overall profile. We utilize the former to train different ML regression estimators, and use the latter to compare their predictions and compute the test error.

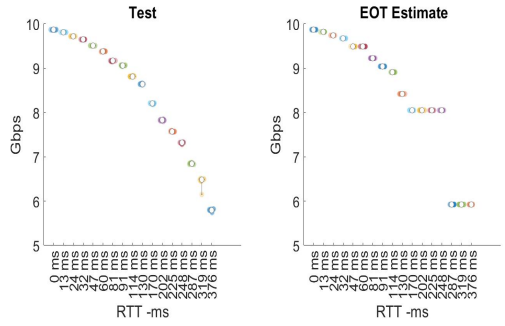
We consider two non-smooth estimators, Ensemble of Trees (EOT) and regression trees (TREE), and two smooth estimators, SVM and Gaussian Process Regression (GPR). Additionally, we consider their hyper-parameter tuning and selection versions Auto Tuning and Selection (AUTO), and two fusers using EOT and SVM methods that combine the outputs of individual regression estimators.

Among the four regression estimators, both non-smooth EOT and TREE estimators achieve low training error but their test error is higher than either smooth method as shown in Fig. 2(c). All four estimators are nearly identical for the training data (Fig. 2(a)) but both non-smooth estimators are not accurate at higher RTT values (Fig. 2(b)). The predicted throughput values show that both non-smooth estimators do not capture the smoothness required for accurate generalization, as shown in Figs. 3(a) and (d). On the contrary, both smooth estimators capture the continuous trend with respect to RTT as shown in Figs. 3(b)-(c), which results in lower test error that reflects their generalization property. Interestingly, SVM with the highest training error achieves nearly the lowest test error, primarily due it is smooth regression function.

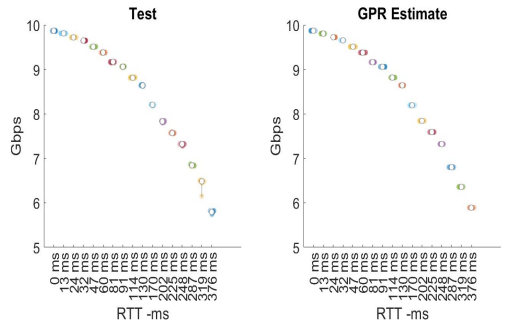
B. Detection of Low-level Radiation Sources

Signatures of low-level radiation sources arise in nuclear safeguards, non-proliferation and security tasks, and are studied using spectral measurements from gamma-ray detectors located at different distances from the source. We utilize data sets collected using detectors deployed over a 6 x 6 meters area in a formation of two concentric circles and one spiral, with the source located at the center (described in detail in [14]). The activity levels in spectral regions associated with possible ^{235}U signatures are estimated as counts at 1 second intervals, and are used as features to train different ML classifiers using the background and source measurements collected over multiple experimental runs. Two different methods are used for the source detection task. First, the distance to the source is estimated using a regression function which is thresholded to generate the Boolean detection decision. The taining and testing errors using EOT, GPR and AUTO ML methods are shown in Figs. 4(a) and (b), respectively, as a function of increasing detector distance from the source. The AUTO method achieved significantly lower training error for 7 farthest detectors from the source but its test error no lower than others.

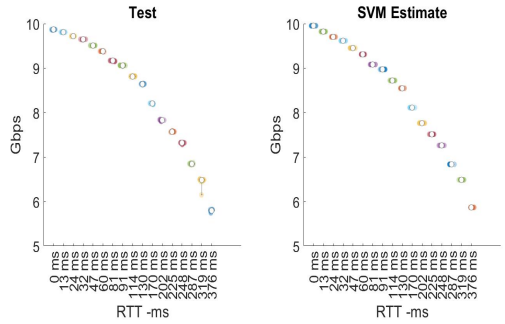
In the second method, eight classifiers that represent diverse designs, and six fusers that combine their outputs are considered. The classifiers are: AUTO, Classification Trees (CTREE), Error Correcting Output Codes (ECOC), Ensemble of Trees (EOT), k Nearest Neighbors (KNN), Naive Bayes (NB), Neural Network (NN), and Support Vector Machine (SVM). These classifiers are described in [14], and AUTO uses



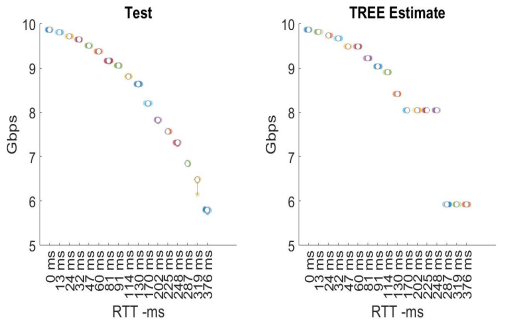
(a) *EOT*



(b) *GPR*



(c) *SVM*



(d) *TREE*

Fig. 3: Training and testing of throughput values using non-smooth EOT and TREE regressions and smooth GPR and SVM regressions.

the hyper-parameter searching of individual methods, CTREE, EOT, KNN, NB, and SVM, and chooses one among them based on training data. These classifiers represent the diversity of design, namely smooth and non-smooth, statistical, structural, and hyper parameter tuning and classifier selection methods [7], [15], [16]. The classifiers are fused in three different ways: (i) all eight classifiers are fused using EOT and SVM fusers, denoted by EF and SF, respectively; (ii) the hyper-parameter and selection classifier AUTO and two non-smooth classifiers CT and EOT are fused using EOT and SVM methods, denoted by ACEEF and ACESF, respectively; and (iii) the classifiers AUTO, KNN, NN are fused using EOT and SVM methods, denoted by AKNEF and AKNSF, respectively. The results show significant over-fitting by all fusers as they achieve low training error as shown in Fig. 4(c) but much higher test error as shown in Fig. 4(d).

IV. Hyper Parameter Tuning

The hyper parameter tuning corresponds to expanding the space of estimators $\mathcal{F}$ to $\mathcal{F}_H$ by including more (hyper) parameters. Thus, the training error is minimized by $\tilde{f}_H$ chosen from a larger class $\mathcal{F}_H$. Since $\mathcal{F} \subseteq \mathcal{F}_H$, we have

(i) $I(f_H^*) \leq I(f^*)$, where $I(f_H^*) = \min_{f \in \mathcal{F}_H} I(f)$,

(ii) $\hat{I}(\hat{f}_H) \leq \hat{I}(\hat{f})$, where $\hat{I}(\hat{f}_H) = \min_{f \in \mathcal{F}_H} \hat{I}(f)$, and

(iii) $\delta_{\mathcal{F}_H}(\gamma, l) \geq \delta_{\mathcal{F}}(\gamma, l)$.

Here, f_H^* and $\hat{f}_H$ are the expected best and empirical best hyper-parameter tuned estimators, respectively. For the computed estimates $\tilde{f} \in \mathcal{F}$ and $\tilde{f}_H \in \mathcal{F}_H$, we have

$$\hat{I}(\tilde{f}) = \hat{I}(\hat{f}) + \hat{\epsilon} \quad \text{and} \quad \hat{I}(\tilde{f}_H) = \hat{I}(\hat{f}_H) + \hat{\epsilon}_H.$$

Since the hyper parameter tuning reduces the training error, we typically have $\hat{\epsilon}_H \leq \hat{\epsilon}$. Then, the condition for higher confidence for $\tilde{f}_H$ compared to $\tilde{f}$ for the same precision γ is given by

$$\delta_{\mathcal{F}_H}(\gamma - \hat{\epsilon}_H - I(f_H^*), l) \leq \delta_{\mathcal{F}}(\gamma - \hat{\epsilon} - I(f^*), l) \qquad (5).$$

Since both $\delta_{\mathcal{F}}(\epsilon, l)$ and $\delta_{\mathcal{F}_H}(\epsilon, l)$ are increasing functions of precision parameter, a necessary condition is

$$\gamma - \hat{\epsilon}_H - I(f_H^*) \geq \gamma - \hat{\epsilon} - I(f^*)$$

or equivalently $\hat{\epsilon}_H + I(f_H^*) \leq \hat{\epsilon} + I(f^*)$, which is typically satisfied. However, this does not guarantee a superior performance of hyper-parameter tuned estimate. We illustrate a sufficiency condition using the derivatives of the confidence functions. Using Taylor expansion we have $\delta(\gamma - \alpha, l) \approx \delta(\gamma) + D_\delta \alpha$, where D_δ is the negative of derivative. Then, the condition for the superior performance in Eq. (5) is expressed as

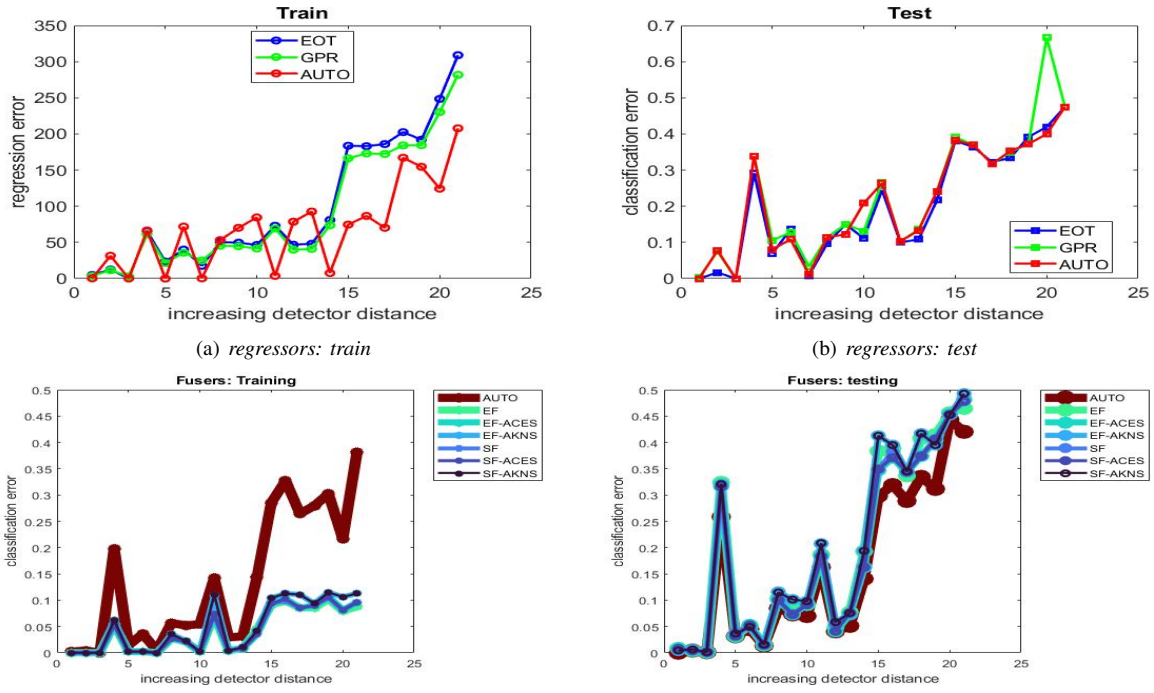
$$\delta_{\mathcal{F}_H}(\gamma) - \delta_{\mathcal{F}}(\gamma) \leq D_\delta[\hat{\epsilon} + I(f^*)] - D_{\delta_H}[\hat{\epsilon}_H + I(f_H^*)]$$

which means that the difference in confidence values at γ must be overcome by the sum of deductions in $\hat{\epsilon}$ and $I(f^*)$ appropriately scaled by the derivatives. Under condition $D_\delta \geq D_{\delta_H}$, it simplifies to the sufficiency condition

$$\frac{1}{D_\delta}[\delta_{\mathcal{F}_H}(\gamma) - \delta_{\mathcal{F}}(\gamma)] \leq [\hat{\epsilon} - \hat{\epsilon}_H] + [I(f^*)) - I(f_H^*)],$$

which shows that the reduction in training error and approximation error have an additive effect in improving the generalization error. This condition is true for example when $\delta_{\mathcal{F}}(\gamma, l) = Ae^{-B\gamma^2 l}$ and $\delta_{\mathcal{F}_H}(\gamma, l) = A_H e^{-B\gamma^2 l}$, $A_H \geq A$.

The results of hyper-parameter tuning for EOT, GPR, SVM and TREE are summarized in Fig. 5; the differences between the estimates of EOT and SVM and their hyper-parameter tuned versions are shown in (a) and (b), respectively. The difference between physical measurements and estimates of tuned EOT and SVM are shown in Fig. 5(c), where the latter are much



(a) *regressors: train*

(b) *regressors: test*

(c) *classifier fusers: train*

(d) *classifier fusers: test*

Fig. 4: Training and test detector errors of regression method, and classifiers and fusers for radiation source detection.

smaller. The SVM method has higher reduction in the training error as a result of hyper-parameter tuning, and the lowest test error, as shown in Figs. 5(b)-(d). On the other hand, the hyper-parameter tuning of EOT did not reduce the training error, and resulted in higher test error, as shown in Figs. 5(a),(c)-(d). These results illustrate the above necessary conditions on reductions in training and approximation errors for improved generalization.

V. Fusion and Ensemble of Regressions

We consider the fusion and ensembles of ML regression estimators wherein the former typically refers to using different methods and latter uses similar methods. To simplify the presentation in this section we use the term fuser to refer to both. We study two basic approaches:

(i) A *selection* fuser chooses one of its constituent estimators possibly by using hyper-parameter tuning. For throughput estimation application, we consider three ML estimates: AUTO using EOT, GPR, SVM and TREE as constituent estimators; its smooth version using GPR and SVM; and its non-smooth version using EOT and TREE.

(b) A *combination* fuser combines the outputs of constituent estimators using another ML estimator. For the application, we consider non-smooth EOT fuser and smooth SVM fuser, both using EOT and SVM as constituent estimators.

The training and test errors of three selection fusers, AUTO and its smooth and non-smooth versions, and two combination fusers, EOT and SVM, are shown in Fig. 7; for comparison, errors of constituent estimators are also shown. Among the fusers, the test error of SVM fuser (S-F) and smooth version of AUTO (A-S) are lower than their training errors, as is the case with smooth SVM and GPR estimates. The AUTO selection fuser (AUT) using both smooth and non-smooth constituent estimators has the highest test error. The differences between AUTO non-smooth (A-N) and smooth (A-S) versions are evident in the respective smooth and non-smooth estimates shown in Figs. 6(a) and (b), respectively. For combination fusers, the estimate of EOT fuser (F-E) is less smooth compared to SVM fuser (F-S), shown in Figs. 6(c) and (d) respectively, even though both use the same constituent EOT and SVM estimators.

We consider the fuser class $\mathcal{F}_F$ used in fusing the estimators $f_A \in \mathcal{F}_A, A \in \mathcal{A}_I$. Let f_F denote the regression function of a selection or combination fuser obtained by composing f_A's using a fuser function from $\mathcal{F}_F$. The *error reduction* Δ_F of the fused estimate f_F over the best individual classifier is

$$\Delta_F = \min_{A \in \mathcal{A}_I} I(f_A) - I(f_F).$$

A fuser class $\mathcal{F}_F$ satisfies the isolation property if it contains a function that simply transfers each of its input to output [17], and this property ensures $\Delta_F \geq 0$. This condition is satisfied by the selection fusers and not necessarily so by combination fusers. The best expected error reduction is given by

$$\Delta_F^* = \min_{A \in \mathcal{A}_I} I(f_A^*) - I(f_F^*).$$

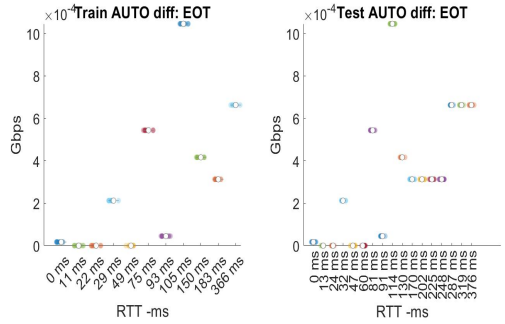
and its estimate based on a sample is given by

$$\tilde{\Delta}_F = \min_{A \in \mathcal{A}_I} \hat{I}(\tilde{f}_A) - \hat{I}(\tilde{f}_F).$$

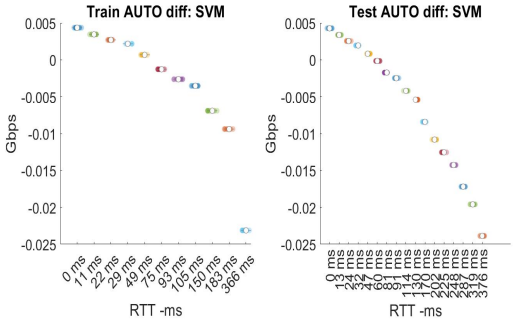
Consider that there exists $\delta_{\mathcal{F}_A}(\epsilon - \hat{\epsilon}_A, l)$ such that based on i.i.d. l-sample, we have

$$\mathbb{P}^l_{X,Y}\left[I(\tilde{f}_A) - I(f_A^*) > \epsilon\right] < \delta_{\mathcal{F}_A}\left(\epsilon - \hat{\epsilon}_A, l\right). \qquad (2)$$

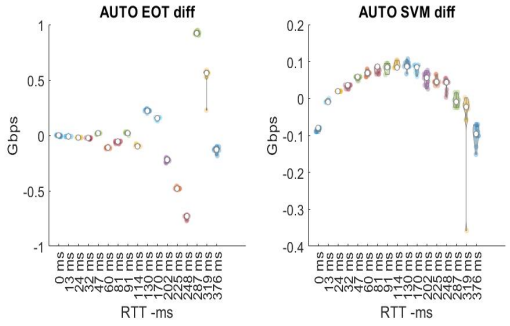
for individual estimators $A \in \mathcal{A}_I$, $N_{\mathcal{A}_I} = |\mathcal{A}_I|$ such that $\delta_A(\epsilon -$



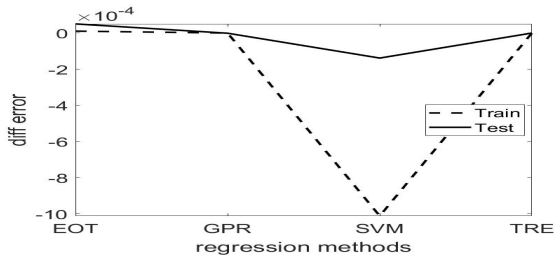
(a) *EOT fit difference*



(b) *SVM fit difference*



(c) *Physical errors of EOT and SVM*



(d) *error reduction due to hyper*

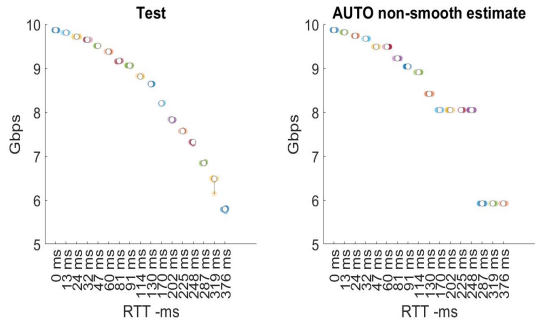
Fig. 5: Hyper parameter tuning of regression estimators.

$\hat{\epsilon}_A, l) \to 0$ as $l \to \infty$, and similarly for fuser F we have

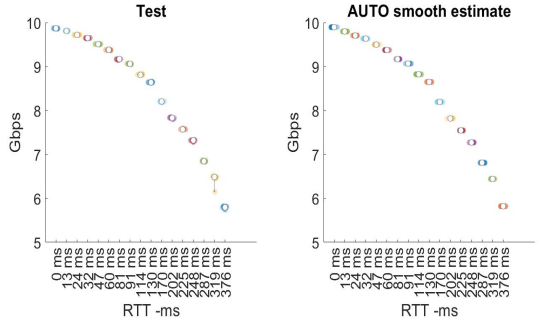
$$\mathbb{P}^l_{X,Y}\left[I(\tilde{f}_F) - I(f^*_F) > \epsilon\right] < \delta_{\mathcal{F}_F}\left(\epsilon - \hat{\epsilon}_F, l\right). \quad (3)$$

The estimate $\tilde{\Delta}_F$ is shown to be within ϵ of the optimal Δ^*_F with a probability that improves with l independent of $\mathbb{P}_{Y,X}$ [6]. In particular, the probability that the closeness between $\tilde{\Delta}_F$ and Δ^*_F is within ϵ is bounded as

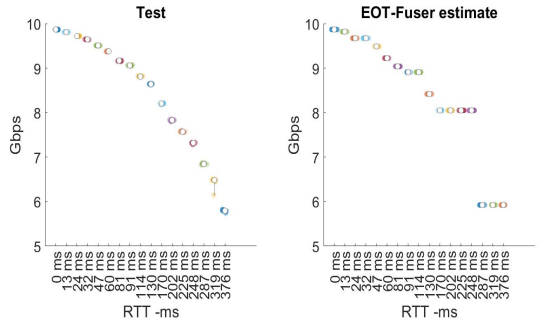
$$\mathbb{P}^l_{X,Y}\left[|\tilde{\Delta}_F - \Delta^*_F| < \epsilon\right] > 1 - \delta_{\mathcal{F}_F}\left(\epsilon/2 - \hat{\epsilon}_F, l\right) - \sum_{A\in\mathcal{A}_I} \delta_{\mathcal{F}_A}\left(\epsilon/(2N_{\mathcal{A}_I}) - \hat{\epsilon}_A, l\right).$$



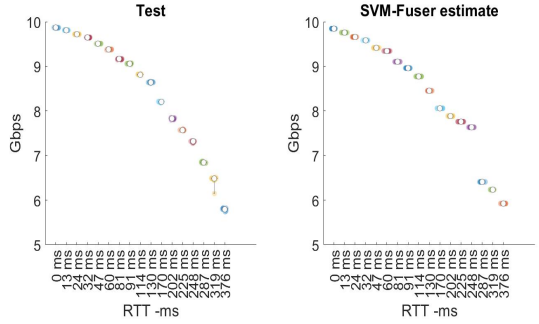
(a) *AUTO non-smooth A-N*



(b) *AUTO smooth A-S*



(c) *EOT fuser F-E*



(d) *SVM fuser F-S*

Fig. 6: Training and testing profiles and error of AUTO fusers of smooth and non-smooth estimates, and EOT and SVM fusers.

Then, the generalization equation for fused estimate $\tilde{f}_F$ is [17]

$$\mathbb{P}^l_{X,Y}\left[I(\tilde{f}_F) - \min_{A\in\mathcal{A}_I} I(f^*_A) < \epsilon - \Delta^*_F\right] > \delta_{\mathcal{F}_F}\left(\epsilon/2 - \hat{\epsilon}_F, l\right) + \sum_{A\in\mathcal{A}_I} \delta_{\mathcal{F}_A}\left(\epsilon/(2N_{\mathcal{A}_I}) - \hat{\epsilon}_A, l\right),$$

for fuser F. In comparison with the generalization equation of a constituent estimator, the left hand side of this equation

indicates an improved precision due to its reduction by Δ_F^* which is non-negative under the isolation property, in particular, for selection fusers. But, the right hand side represents decrease in confidence, which becomes larger with more constituent estimators, as illustrated for AUTO with four estimators (AUT) in Fig. 7. Then, the generalization equation for fused estimate $\tilde{f}_F$ is given by

$$\begin{aligned}\mathbb{P}^l_{X,Y}&\left[I(\tilde{f}_F) > \gamma\right] \\ &< \delta_{\mathcal{F}_F}\left[(\gamma + \Delta_F^* - I^*]/2 - \hat{\epsilon}_F, l\right) \\ &+ \sum_{A\in\mathcal{A}_I} \delta_{\mathcal{F}_A}\left(\frac{\gamma + \Delta_F^* - I^*}{2N_{\mathcal{A}_I}} - \hat{\epsilon}_A, l\right),\end{aligned}$$

where $I^* = \min_{A\in\mathcal{A}_I} I(f_A^*)$. For a fixed precision parameter γ, higher confidence is achieved by lower values of the first operator ϵ of $\delta_{\mathcal{F}}(\epsilon, l)$ and less of such terms. Thus, the larger training error of fuser $\hat{\epsilon}_F$ and constituent estimators $\hat{\epsilon}_A$ increase the right hand side, there by reducing the confidence; thus, fusers with lower training error provide better generalization. In terms of the epistemic parameters, larger error reduction due to fusers Δ_F^* leads to improved confidence but the larger minimum error $I^* = \min_{A\in\mathcal{A}_I} I(f_A^*)$ and the number of constituent estimators $N_{\mathcal{A}_I}$ both have the opposite effect. By comparing with the generalization equation of a constituent estimate $\tilde{f}_C$ given by

$$\mathbb{P}^l_{X,Y}\left[I(\tilde{f}_C) > \gamma\right] < \delta_{\mathcal{F}_C}\left(\gamma - I(f_C^*) - \hat{\epsilon}_c, l\right),$$

we obtain the condition for superior confidence in fuser's generalization given by

$$\begin{aligned}&\left[\delta_{\mathcal{F}_F}(\gamma/2) + \sum_{A\in\mathcal{A}_I}\delta_{\mathcal{F}_A}(\gamma/2N_{\mathcal{A}_I})\right] - \delta_{\mathcal{F}_C}(\gamma) \\ &< D_C\hat{\epsilon}_C - \left[D_F\hat{\epsilon}_F + \sum_{A\in\mathcal{A}_I} D_A\hat{\epsilon}_A\right] \\ &+ D_C I(f_C^*) - I^*(D_F + \bar{D}_A)/2 + \Delta_F^*(D_F + \bar{D}_A)/2,\end{aligned}$$

where $\bar{D}_A = \frac{1}{N_{\mathcal{A}_I}}\sum_{A\in\mathcal{A}_I} D_A$. In essence, the positive effects of $\hat{\epsilon}_F$ and Δ_F^* (lower and higher, respectively) must be large enough to offset the negative effects due to the increased number of terms and their effects on the right hand side. When the fuser utilizes hyper-parameter tuning of constituents, $\hat{\epsilon}_A$'s on the right hand side need to be replaced by their corresponding versions described in previous section, which in turn requires suitably smaller $\hat{\epsilon}_F$ and larger Δ_F^* to outperform the constituents.

VI. Conclusions

We presented two practical applications where independent tests illustrated over-fitting by ML methods, wherein a low training error in some cases is a misleading indicator of their generalization ability. We presented a generic decomposition of the generalization equations that separates the training error terms from the structural approximation and learnability terms, thereby providing a mechanism to account for both in analyzing and ensuring the generalization performance. We discussed conditions for superior performance of hyper parameter tuning and fusion methods over their constituents for regression estimation,

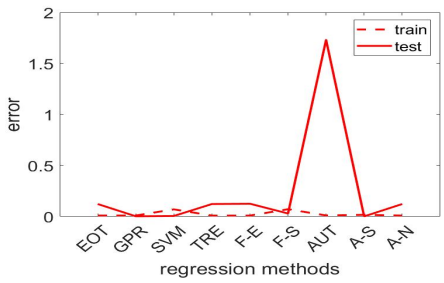


Fig. 7: Errors of AUTO, AUTO for smooth (A-S) and non smooth (A-N), and EOT (F-E) and SVM fusers (F-S).

and illustrated them using experimental results for throughput profile estimation of a data transport infrastructure.

Future directions include expanding the scope to include the classification problems, and application areas that rely on algebraic properties such as Hilbert spaces for quantum channel tomography, non-smooth settings of compute-throughput profiles of cyber infrastructures, and probabilistic settings such as detection and estimation based on gamma spectra.

References

[1] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. The MIT Press, 2018. second edition.

[2] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

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

[4] N. S. V. Rao, "Generalization equations for machine learners based on physical and abstract laws," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2021.

[5] 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, H. H. Hesse, M. Maceira, C. Chai, O. Marcillo, T. Karnowski, and R. Wetherington, "Reactor power level estimation by fusing multimodal sensor measurements," in *International Conference on Inforamation Fusion*, 2020.

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

[7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2001.

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

[9] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Wadsworth and Brooks, 1984.

[10] B. Scholkopf, C. J. C. Burges, and A. J. Smola, eds., *Advances in Kernel Methods*. MIT Press, 1999.

[11] W. Mass, "Agnostic PAC learning of functions on analog neural nets," *Neural Computing*, vol. 7, pp. 1054–1078, 1995.

[12] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[13] K. Tumer and J. Ghosh, "Error correlation and error reduction in ensemble classifiers," *Connection Science*, vol. 8, no. 3, pp. 385–404, 1996.

[14] N. S. V. Rao, D. Hooper, and J. Ladd-Lively, "Study of classifiers for u-235 source signatures using gamma spectral measurements," in *Institute of Nuclear Materials Management Annual Meeting*, 2022.

[15] E. Alpaydin, *Introduction to Machine Learning*. MIT Press, 2020. fourth edition.

[16] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley and Sons, Inc., 2001. Second Edition.

[17] N. S. V. Rao, "Measurement-based statistical fusion methods for distributed sensor networks," in *Distributed Sensor Networks* (S. S. Iyengar and R. R. Brooks, eds.), Chapman and Hall/CRC Publishers, 2011. 2nd Edition.