

# Multi-Kernel-based Adaptive Support Vector Machine for Scalable Predictive Maintenance

Koushik A. Manjunatha and Vivek Agarwal

*Idaho National Laboratory, Idaho Falls, ID 83402, USA*

*koushik.araseethotamanjunatha@inl.gov*

*vivek.agarwal@inl.gov*

## ABSTRACT

Application of data-driven solutions across an industry is challenging, since the data are often stored locally, and increasing privacy and security concerns restrict access to the data. Because it is highly unlikely that all potential data patterns are captured in a single data source, machine learning (ML) models developed from a single source cannot be robust enough. An alternative is to train local ML model at each source and at the central location combine all the local models to generate a global model. In this work, we develop a proof-of-concept of distributed machine learning model, federated transfer learning, using a multi-kernel-based adaptive support vector machine. For federated learning, the multi-kernel approach enables feature-specific model aggregation under data heterogeneity; whereas for transfer learning the adaptive model enables utilization of an aggregated model from a different task. The proposed approach is validated using nuclear power plant vertical motor-driven pump data to predict the health condition of vertical motor-driven pumps as an anomaly detection. The efficiency of the proposed approach is also quantified and compared with neural network.

## 1. INTRODUCTION

Artificial intelligence is driving the advancement of technologies in the fields of healthcare, industrial automation, transportation, etc. (National Artificial Intelligence Initiative, 2020). The enormous amount of data generated at different locations from various sources such as sensors and Internet of Things devices requires high bandwidth to transmit data. Also, low-latency real-time decision making requires continuous network connection. Failure to meet bandwidth or network connection requirements would lead to unreliable decision making at a centralized location. This drives a localized decision-making capability the more appropriate solution. For localized decision making (i.e., edge computing) using

data-driven models, it is challenging to incorporate all possible data patterns into the data-driven models. Data collected from various sites provide a better estimation of the population than do data collected at a single site. However, privacy, security, legal, and commercial concerns restrict data sharing. For example, the U.S. Health Insurance Portability and Accountability Act (Edemekong, Annamaraju, & Haydel, 2018) requires that medical and individual data only be released with proper anonymization (Li & Qin, 2017). Similarly, in commercial terms, data related to processes and manufacturing are often a valuable business asset. However, the central accumulation of summaries or obfuscated models may be considered reasonable as long as the original data are not revealed. Thus, it is essential to enable privacy-preserving distributed mining of information and decision making. That said, a comprehensive model can be developed by combining all the local models into a single aggregated model. Realization of a comprehensive model is achieved via a central entity (server) with which all the distributed edge devices share their local model at customized intervals. This relies on performing distributed training on the edge devices and edge servers where the data is generated or collected, without necessarily sending data outside the enterprise firewall to the central server.

Often, edge devices have limited computational capabilities, meaning frequent transmitting and receiving will not be energy efficient. This challenge increases with complex models such as deep neural networks (Goodfellow, Bengio, & Courville, 2016). In addition, the more complex models are "black box" in nature, and there is always an added challenge in deriving the proper appreciation for the data-driven models from an enterprise/business perspective. In this research, we focus on a specific classifier: the support vector machine (SVM) (Suykens & Vandewalle, 1999). We leverage this popular classification model in machine learning (ML) to build a collaborative framework for distributed learning, along with a unique way to model aggregation.

Over the last decades, several collaborative SVM-based data mining approaches have been proposed in order to enhance

Koushik A. Manjunatha et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

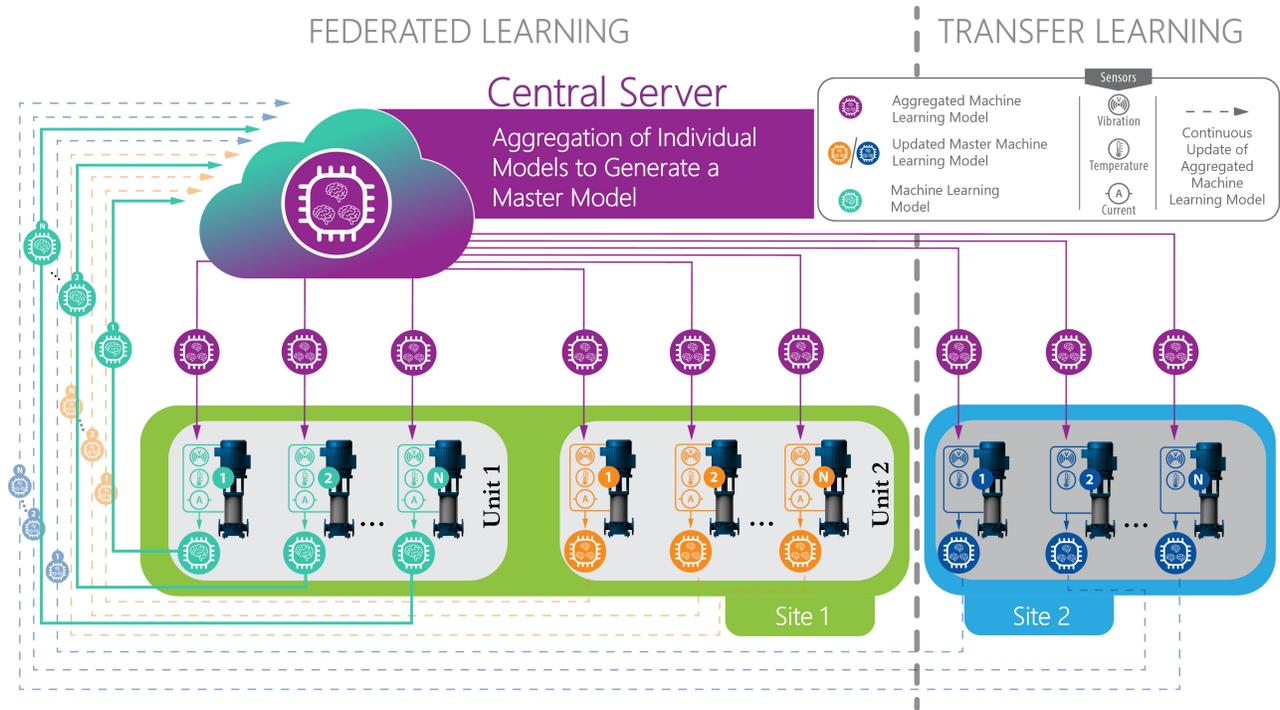


Figure 1. Federated transfer learning framework for condition assessments of vertical motor-driven pumps in the circulating water systems of two plant sites.

distributed learning. A privacy-preserving SVM classification was proposed in (Vaidya, Yu, & Jiang, 2008), constructing the global SVM classification model from data distributed at multiple parties without disclosing the parties' data to each other. Using a similar approach, a collaborative learning framework (Que, Jiang, & Ohno-Machado, 2012) was developed for SVM. Collaborative multi-kernel SVM (MK-SVM) with alternating direction method of multipliers was used to globally optimize the distributed sub-models (Chen & Fan, 2012). In this approach, the training matrix is partitioned into blocks in two different ways (i.e., column partitioning and row partitioning), and multiple kernels are extracted for each model and then aggregated. Federated learning (FL) with SVM was implemented in (Carlsson, 2020), with the kernel values being combined and then shared back to the local models.

To address collaborative learning challenges through SVM, this work proposes multi-kernel-based adaptive SVM (MK-A-SVM)-based federated transfer learning (FTL), as shown in Figure 1. FTL is a combination of FL and transfer learning (TL). FL is a collaborative learning technique in which many clients collaboratively train a model under the orchestration of a central controller, without exchanging the party's original data. FL enables focused data collection and data minimization by reducing the systematic privacy risks and costs resulting from traditional, centralized ML. The FL process is typically driven by model engineers who develop ML models for particular applications. TL builds an effective model for

the target domain while leveraging knowledge from the other (source) domains. The main advantages of TL are that the training time is significantly reduced and only a very small amount of training data (or none at all) are required to leverage pretrained models. To enable FTL across heterogeneous data, a feature-group-based MK-SVM was developed. The features are grouped based on measurement type, with each group having its own kernel/parameter settings. The predicted category is the weighted sum of the contributions from each kernel. To enable TL, the MK-SVM is modified by integrating an adaptive SVM framework and redefining the optimization approach for SVM. The adaptive nature of the MK-A-SVM also enables the integration of other ML models (e.g., neural networks [NN] and Bayesian methods) with SVMs in TL. Unlike traditional SVM, MK-A-SVM supports (1) the adoption of multiple and feature-specific kernels, and (2) a resilient approach to dealing with missing measurements.

## 2. MULTI-KERNEL ADAPTIVE SUPPORT VECTOR MACHINE

SVM (Suykens & Vandewalle, 1999) is a discriminative classifier that finds, in a higher dimensional space, a hyperplane that distinctly classifies the data points. To separate two classes, there may be many possible hyperplanes. SVM finds the hyperplane of maximum margin with the longest distance between the data points of both classes. Support vectors are data points on the hyperplane that influence the position and orientation of the hyperplane. For the input feature vector

$x \in X$  with labels  $Y$ , the general expression for the soft-margin classifier in dual form with regularization parameter,  $C$  is given by (Suykens & Vandewalle, 1999)

$$\begin{aligned} \max_{\alpha} \quad & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j Y_i Y_j K(x_i, x_j) \\ \text{s.t.} \quad & \sum_i \alpha_i Y_i = 0, \\ & C \geq \alpha_i \geq 0, \forall_i, \forall_j \end{aligned} \quad (1)$$

where  $\alpha$  is the Lagrange multiplier and  $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$  is a kernel function.

## 2.1. MK-SVM

With the kernel function, the soft margin decision function for SVM is defined by:

$$f(x) = \sum_{i=1}^n \alpha_i y_i K(x_i, x_j) + b \quad (2)$$

Let  $[X]_{m \times n}$  represent a data matrix with  $m$  samples and  $n$  features. The sample set can be vertically partitioned based on feature type. The MK-SVM across the vertically partitioned sample set can be determined by computing a net kernel matrix (i.e., a gram matrix)  $K = K(x_i, x_j)$  from individual matrices determined from each vertically partitioned sample. In the case of two partitions, the  $[X]_{m \times n}$  data matrix can be vertically partitioned into  $X^1$  and  $X^2$ . Then  $X^1$  and  $X^2$  will have  $K^1 = K(X^1, X^{1T})$  and  $K^2 = K(X^2, X^{2T})$  as the gram matrix, respectively. The net gram matrix can next be determined as the linear combination of individual gram matrices. Let the  $(i, j)^{th}$  element of  $K$  represent  $K(x_i, x_j)$  and  $x_i^1$  and  $x_i^2$  be vertically partitioned vectors of  $x_i$  from  $X^1$  and  $X^2$ , respectively. Accordingly:

$$\begin{aligned} K(x_i, x_j) &= K(x_i^1, x_j^1) + K(x_i^2, x_j^2) \\ \therefore K(X, X^T) &= K^1 + K^2 = K(X^1, X^{1T}) + K(X^2, X^{2T}) \end{aligned} \quad (3)$$

The net gram matrix can also be obtained via the weighted summation of individual kernels. Hence, for  $G$  vertical partitions, equation (3) can be generalized as:

$$K(X, X^T) = \beta_1 K^1 + \beta_2 K^2 + \dots + \beta_G K^G = \sum_{i=1}^G \beta_i K(X^i, X^{iT}) \quad (4)$$

where  $\beta_i$  is the weight associated with each local gram matrix and  $\sum_i^G \beta_i = 1$ . Using the net kernel matrix applied in equation (1), the SVM model parameters can be obtained by solving a quadratic programming problem. Note that each gram matrix will be a square matrix.

Thus, using the MK approach, FL across  $P$  parties can be performed to generate a global (master) gram matrix as the weighted sum of net kernel matrices  $K_i$  (determined in equa-

tion [4]), for  $t = [1, 2, \dots, P]$ . The formulation to generate a global gram matrix is given by:

$$K_g(X, X^T) = \sum_{t=1}^P p_t K_t(X, X^T) \quad (5)$$

where  $p_t$  is the weight/importance associated with each net gram matrix and  $\sum_t^P p_t = 1$ . The importance of each net gram matrix can be considered equal ( $1/P$ ), or be based on the individual contribution to overall performance (e.g., prediction accuracy or  $F_{\beta}$ -score) denoted as  $ACC_t$  is determined as:

$$p_t = \frac{ACC_t}{\sum_{j=1}^P ACC_j} \quad (6)$$

As per equation (6), kernel aggregation is conducted by giving the highest importance to the most accurate individual model, and the lowest importance to the least accurate model. Then the  $K_g$  matrix, which captures patterns from all the individual models, is used to retrain each individual model. After retraining with  $K_g$ , all groups will have a global model, and the above process of individually updating the kernel matrix and constructing a global matrix from all the  $P$  parties continues. Substituting equation (4) for (5), we get:

$$K_g(X, X^T) = \sum_{t=1}^P p_t \sum_{i=1}^G \beta_i K_t(X^i, X^{iT}) \quad (7)$$

The global gram matrix will be shared with each party. Then, using the global gram matrix in equation (1), each party can obtain the SVM model parameters by solving a quadratic programming problem. Creating a feature-group-based MK-SVM model enables the aggregation of kernel matrices across individual parties with similar or partially heterogeneous features.

## 2.2. Adaptive SVM

Adaptive SVM is useful for transferring a trained model from one system to another, or for performing domain adaptations. The adaptive SVM learns from the source model  $f^s(x)$  by regularizing the distance between the learned model (target model)  $f(x)$  and  $f^s(x)$ . These source models can be trained using any algorithm (e.g., SVM, decision tree, NN, and naive Bayes). Let  $[X]_{m' \times n'}$  represent a data matrix with  $m'$  samples and  $n'$  features for the target model. The distribution of the target data is likely to be different from the source data. Let  $f^s(x)$  represent the decision model available from the source data, our goal being to learn classifier  $f(x)$  using  $M$  different source models  $f^s(x)$  as per:

$$\begin{aligned} f(x) &= f^s(x) + \Delta f(x) \\ &= \sum_{k=1}^M t_k f_k^s(x) + \Delta f(x) \end{aligned} \quad (8)$$

where  $t_k$  is the weight associated with each source model. The objective is to determine a new decision boundary that is close to  $\sum_{i=1}^M t_k f_k^s(x)$  and accurately predicts on  $[X]_{m' \times n'}^T$ . Thus:

$$\left\| f - \sum_{i=1}^M t_k f_k^s \right\|^2 = \|\Delta f\| \quad (9)$$

Equation (9) is a more intuitive interpretation of regularization aimed to minimize the distance between  $f(x)$  and the source model  $f^a(x)$ . To learn the parameters, the objective function in equation (1) can be rewritten as follows (Aytar & Zisserman, 2011; Yang, Yan, & Hauptmann, 2007):

$$\begin{aligned} \max_{\alpha} \quad & \sum_i (1 - \lambda_i) \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j Y_i Y_j K(x_i, x_j) \\ \text{s.t.} \quad & \sum_i \alpha_i Y_i = 0, \\ & C \geq \alpha_i \geq 0, \forall_i, \forall_j \end{aligned} \quad (10)$$

where  $\lambda_i$  is defined as  $\lambda_i = y_i \sum_{i=1}^M t_k f_k^s(x_i)$ .

Given the  $\hat{\alpha}$  determined from equation (10) and the results from the MK-SVM in equation (7), the MK-A-SVM decision function using equation (8) can be written as:

$$\begin{aligned} f(x) &= \sum_{i=1}^M t_k f_k^s(x) + \Delta f(x) \\ &= \sum_{i=1}^M t_k f_k^s(x) + \sum_{i=1}^n \hat{\alpha}_i y_i K_g(x_i, x_j) + b \end{aligned} \quad (11)$$

### 3. NUMERICAL RESULTS

To validate FTL using MK-A-SVM, the circulating water systems (CWSs) at two nuclear power plants (NPPs) were selected as the identified plant assets. The CWS is an important non-safety-related system. As the heat sinks for the main steam turbine and associated auxiliaries, the CWSs at Plant Sites 1 and 2 are designed to maximize steam power cycle efficiency (NRC, 2009). Plant Site 1 (a two-unit pressurized-water reactor) features six circulators at each unit. Schematic representations of the main condensers for Plant Site 1 Unit 2 are shown in Figure 2a. Each pair of waterboxes in the condenser is named according to the following convention: Unit #, Condenser #A, and Unit #, Condenser #B. Plant Site 2 (a single-unit boiling-water reactor) has four circulators. A schematic representation of the Plant Site 2 CWS is shown in Figure 2b, and several distinct differences are seen when comparing it to the Plant Site 1 CWS. These include: (1) the water supply to the Plant Site 2 CWS comes from a cooling tower water basin, not directly from the river; (2) the Plant Site 2 CWS does not have traveling screens, but each circulator has a single-pump screen to prevent debris transmission to the waterboxes; and (3) the Plant Site 2 CWS has

---

#### Algorithm 1 : Federated Transfer Learning

---

**Require:** Feature matrix  $[X]_{m \times n}^t$ , and labels  $[Y]^t$ , for  $t \in P$

===== FEDERATED LEARNING =====

**Central Server do:**

- 1: Initialization: global model  $K_g^0$ .
- 2: **for** each global iteration,  $k$  **do**
- 3:     Determine number of participants,  $1 \leq C_k \leq P$
- 4:     **for** each party  $c \in C_k$  **do**
- 5:         #Get client improved/retrained model  
 $K_c^{k+1}, ACC_c^{k+1} \leftarrow \text{TRAINLOCALLY}(c, K_g^k)$
- 6:     **end for**
- 7:     #Determine model importance  $p_c$   
 $p_c = \frac{ACC_c^{k+1}}{\sum_{j=1}^{C_k} ACC_j^{k+1}}$
- 8:     #Update the global model  
 $K_g^{k+1} \leftarrow \sum_{c=1}^{C_k} p_c K_c^{k+1}$
- 9:     **end for**
- 10:    **if** Global\_model request **then**
- 11:        #Send global model to party  $p$   
 $\text{SendGlobalModel}(K_g^{k+1})$
- 12:    **end if**

**Component/Plant Site do:**

- 13: **function** TRAINLOCALLY( $c, K_g^k$ )
- 14:     **for** each local iteration,  $i, j$  **do**
- 15:         #Do local model training  
 $K_c^{ij}, ACC_c^{ij} \leftarrow \text{LocalUpdate}([X]_{ij}^c, K_g^k)$
- 16:     **end for**
- 17:     **return**  $K_c, ACC_c$
- 18: **end function**

===== TRANSFER LEARNING =====

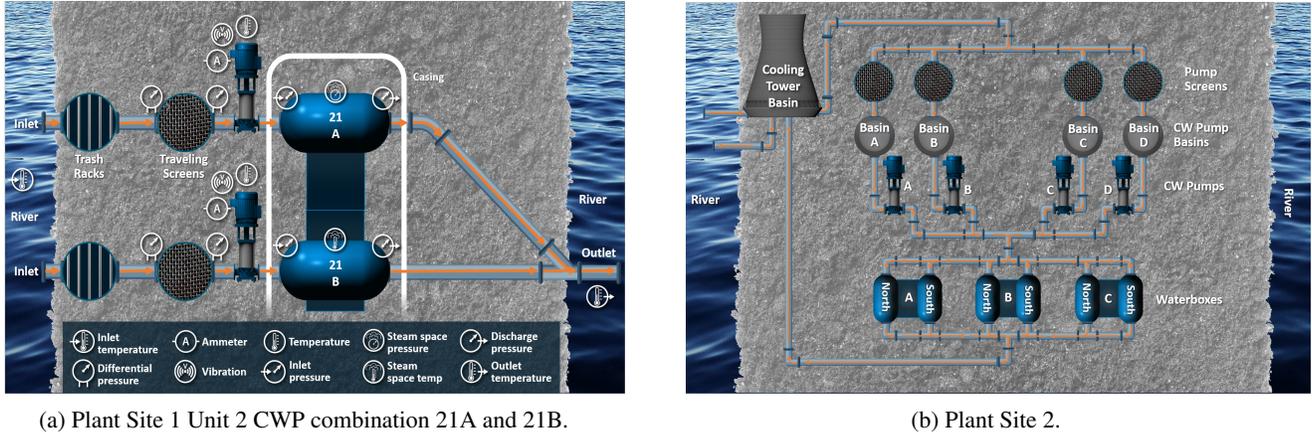
**New Component/Plant Site do:**

- 19: #Receive global model from central server  
 $f^s(x) \leftarrow \text{ReceiveGlobalModel}()$
  - 20: #Adopt global model  
 $f(x) \leftarrow f^s(x) + \Delta f(x)$
  - 21: #Determine Model Performance  
 $ACC_T \leftarrow \text{ModelScore}(f(x))$
  - 22: **if** send\_model\_to\_server **then**
  - 23:      $\text{SendModel}(f(x), ACC_T)$
  - 24: **end if**
- 

four circulators feeding six waterboxes via a common header, unlike the Plant Site 1 CWS, in which each waterbox has its own circulator.

#### 3.1. Fault Signatures of the NPP Asset

Fault signatures enable informed decision making to prevent potential failure of a plant asset. They can also be used for root cause analysis if failure occurs. In theory, the different fault modes associated with a plant asset (e.g., the CWS) have unique, consistently identifiable fault signatures. In practice, fault signature identification and diagnosis are not straightforward and can benefit from analyses of historical data. Each detected fault signature for a particular degradation mode should have enhanced feature verification and confidence by selecting additional process and condition monitoring data



(a) Plant Site 1 Unit 2 CWP combination 21A and 21B.

(b) Plant Site 2.

Figure 2. Schematic representation of CWS.

that provide complementary information.

Of the faults of interest examined, only waterbox fouling caused numerous instances of CWP shutdowns, even though this fault is not a pump/motor fault but rather a system fault whose symptoms may affect pump performance. Because waterbox fouling occurred so commonly, enough data was available to allow development and testing of condition-based monitoring algorithms. Fault types that caused only a single instance of CWP shutdown provided limited information for developing fault signatures and training ML algorithms. The potential fault signatures contained within the data are not readily resolvable at this time. A potential way of addressing this sparseness in some of the fault signatures is to leverage simulated data generated from the first-principles model of the CWS motor and pump (M&P) set. It is anticipated that, as ML technologies mature for operational plant applications, these subtle faults will be identified.

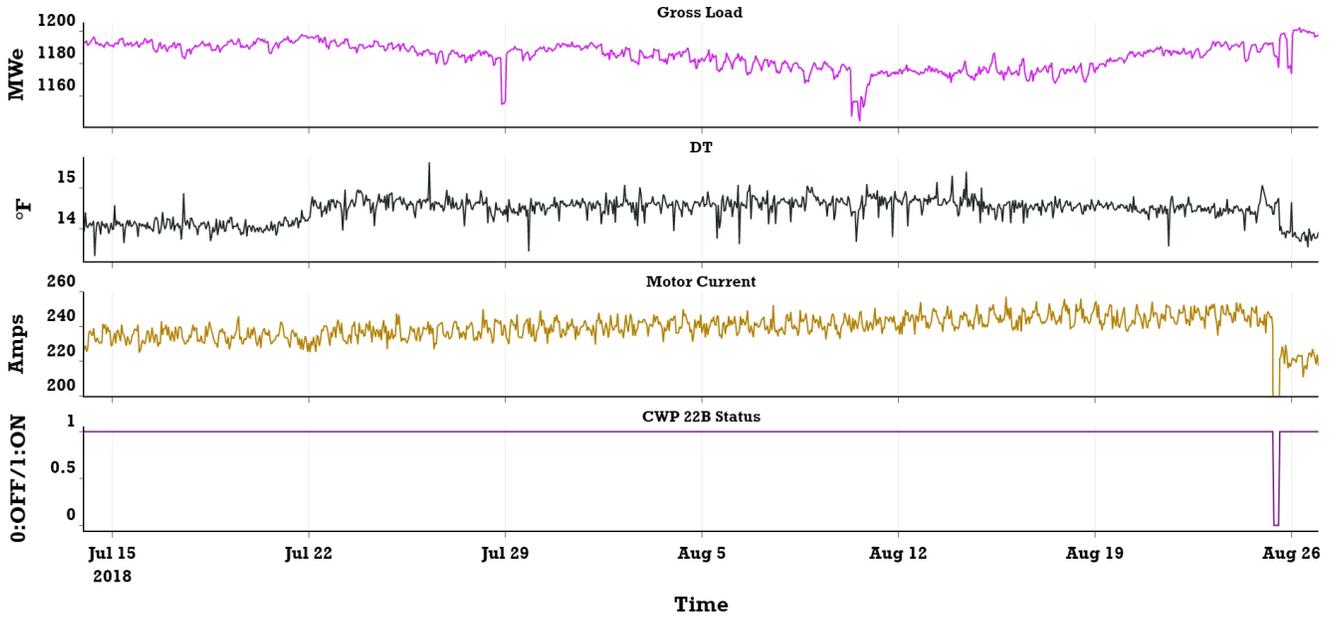
This section discusses two examples of waterbox fouling. The first is from Plant Site 1 and the second from Plant Site 2. These two examples highlight the similarities and differences in the fault signatures for waterbox fouling, and are a perfect lead-in as to why FTL is required for predictive modeling. The primary issue noted with the Plant Site 1 CWS is waterbox fouling, which typically occurs due to accumulation of grass/debris in the waterboxes and causes condenser tube blockage and reduced circulator water flow. This is a unique and frequent issue at Plant Site 1, since the Plant Site 1 CWP intake comes directly from the river, which produces a significant quantity of grass/debris. The grassing season typically occurs between February 1 and May 31 (NRC, 2009). Grassing often emerges from the river during high-wind conditions associated with storms. During these periods, the motor current can oscillate with river level changes. Operations monitors the waterbox motor current and inlet pressure, and schedules waterbox cleanings based on deviations in motor current and inlet pressure when compared against historical baseline data. Waterbox fouling is typically identified via motor cur-

rent increase (also, though far less frequently, motor current decrease), inlet pressure increase, waterbox differential temperature ( $DT$ ) increase, and condenser thermal performance loss.

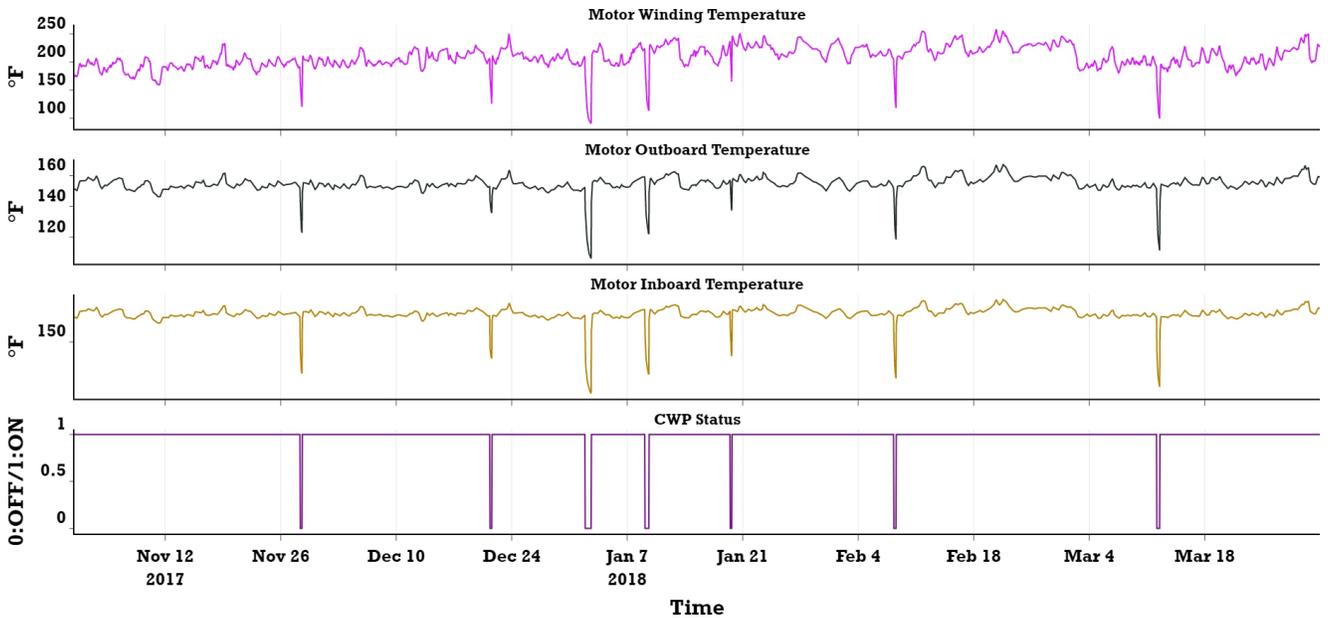
Figure 3a shows an instance of waterbox fouling diagnosed in Plant Site 1 Unit 2's CWP 22B. An upward drift in  $DT$  and motor current was identified on July 23, 2018. Consequently, the gross load began to dip. Note that, in Figure 3a, the CWP 22B motor current increased from 231 to 245 amps, and the  $DT$  increased from 14 to 16°F, with the gross load not trending as expected. The motor current and  $DT$  decreased to 220 amps and 14°F, respectively, following the waterbox cleaning on August 25, 2018, resulting in a 30–40 MWe improvement in gross load. The waterbox fault and approximate date of the shutdown were found by searching the work order database and narrative log information.

For Plant Site 2, waterbox fouling is not a major fault, yet remains of interest. The cause of waterbox fouling at Plant Site 2 is once again debris (limited grassing) in the water circulating in and out of the cooling tower basin. Figure 3b shows an instance of waterbox fouling in Plant Site 2 waterbox A. Under normal operating conditions with no faults, the differential pressure ( $DP$ ) across the Plant Site 2 CWPs averages 40–41 PSIG. Note that, in Figure 3b, on around December 23, 2017, CWP A's  $DP$  began trending upward, exceeding 43 PSIG. Following the  $DP$  trend, the  $DT$  across the north and south ends of waterbox A also trended upward in the same time period. These slow, steady increases in  $DP$  and  $DT$  trends are indicative of waterbox fouling. Following a waterbox cleaning on around January 23, 2018, the  $DP$  reduced to near 41 PSIG, and the  $DT$  also stabilized.

These two examples show that different fault features can indicate the same fault. Developing a comprehensive fault signature for each fault mode is key to achieving scalable, accurate predictive models. For other CWP fault signatures, see (Agarwal et al., 2021).



(a) Plant Site 1 Unit 2's CWP 22B.



(b) Plant Site 2's waterbox A.

Figure 3. Example changes to the CWS process data both before and after waterbox fouling.

### 3.2. Feature Extraction

To develop a FTL-based predictive model, features were extracted based on identified fault signatures.

#### 3.2.1. Plant Site 1

From the CWS-associated plant operational data, the following features were extracted for each M&P set:

- $DT$  was calculated as the difference between the outlet

water temperature associated with the M&P set and the inlet river temperature

- The measured motor in-board ( $MIB$ ) temperature, motor outboard ( $MOB$ ) temperature, and motor stator ( $MS$ ) temperature
- From historical CWS M&P replacement/refurbishment dates; the M&P run-hours from one replacement to the next were considered in calculating the motor age ( $M_{Age}$ ) and pump age ( $P_{Age}$ )

- To consider the seasonal effects on the data, week of the year was calculated for every timestamp and then used as a feature.

Thus, a total of seven features were extracted from the CWS plant operational data for each M&P set. Detailed information on feature extraction from plant operation data—as well as from vibration data—can be found in (Agarwal et al., 2021). For model development, plant operational data after 2016 were considered, because Plant Site 1 first adopted a new six-year CWP replacement PM strategy at that time. Since 2016, each Plant Site 1 unit has had its CWPs periodically replaced as per the updated PM strategy. The age of the M&P set is estimated based on the date of replacement of each CWP. If any faults in the M&P are identified post-replacement, the data corresponding to that fault and time period is labeled either *unhealthy* or *healthy*.

### 3.2.2. Plant Site 2

From the Plant Site 2 CWS-associated plant operational data, the following features were extracted for each M&P set:

- *DT* as an average of the *DT*s measured at the north and south condensers (the *DT* at each condenser was calculated as the difference between the respective condenser inlet and condenser outlet temperatures)
- The measured *MIB* (thrust) temperature, *MOB* temperature, and motor winding (stator) (*MS*) temperature
- To consider the seasonal effects on the data, week of the year was calculated for every timestamp and then used as a feature.

Thus, five features were extracted from the Plant Site 2 data. As there was no historical CWS M&P replacement/refurbishment date information available for Plant Site 2, motor age ( $M_{Age}$ ) and pump age ( $P_{Age}$ ) were not calculated.

Figures 4 and 5 show the distribution plot of each extracted feature in the *healthy* and *unhealthy* classes for each CWP (Plant Site 2) or combination thereof (Plant Site 1). The figures make it evident that *DT* is the most significant feature for the classification algorithm. This makes sense because most of the fault data captured in the *unhealthy* class are associated with waterbox fouling, which is best reflected by *DT* information rather than the other six features discussed above.

### 3.3. FL-based CWP Motor Health Prediction

FL was demonstrated using an MK-SVM (Chen & Fan, 2012) that classifies whether a CWP is in a *healthy* or *unhealthy* state. FL was demonstrated on the Plant Site 1 data, with each local model being developed for a pair of CWPs connected to a common waterbox, as shown in Figure 2a. Since there are three waterboxes for each Plant Site 1 unit, this gives six local models that will be combined into a master model via the FL approach (see Figure 1). The samples that were grouped

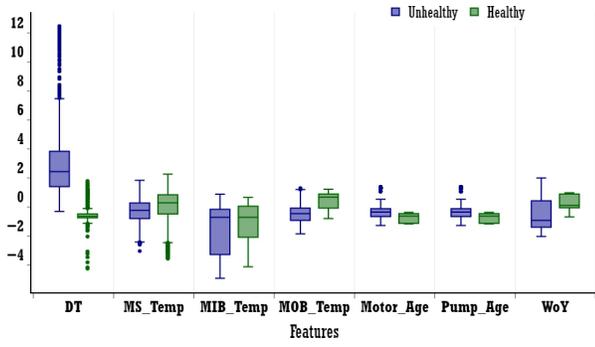
based on CWP combinations are then split into training and test samples in accordance with a 80:20 ratio, as shown in Table 1.

Data Group	Training Samples	Test Samples
Group 1 (CWP 11A & CWP 11B)	6174	974
Group 2 (CWP 12A & CWP 12B)	8303	1309
Group 3 (CWP 13A & CWP 13B)	4366	822
Group 4 (CWP 21A & CWP 21B)	1720	288
Group 5 (CWP 22A & CWP 22B)	2356	476
Group 6 (CWP 23A & CWP 23B)	1496	358

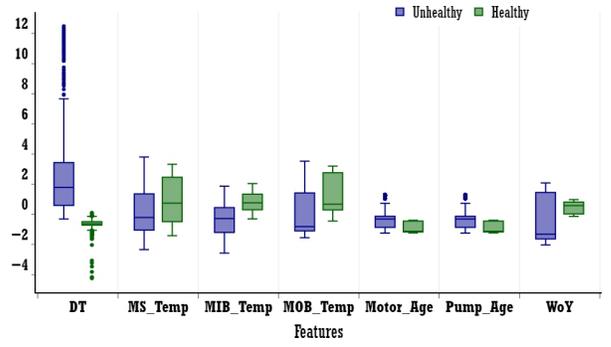
Table 1. Data split into training and test sets per each of the selected groups for predicting CWP conditions.

As per equation 11, the extracted features are grouped into three categories, an individual kernel matrix is built for each group, and the final decision  $f(x)$  is determined as a weighted combination of predictions from each kernel associated with each feature group, as shown in Figure 6. The selection of optimal hyperparameters  $C$  and  $\gamma$  for MK-A-SVM with radial basis function kernels is performed using grid-search cross validation to predict CWP conditions. Since there are three feature groups, the MK-A-SVM takes three parameters of  $\gamma$  ( $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$ ), each of which must be optimally tuned. In this work, for the sake of simplicity, only one  $\gamma$  parameter is tuned, and it is set as  $\gamma = \gamma_1 = \gamma_2 = \gamma_3$ . The parameter  $\gamma$  was varied from  $10e-3$  to  $10e3$ , and the regularization parameter  $C$  was varied from  $10e-3$  to  $10e2$ . For predicting CWP conditions,  $\gamma = 100$  and  $C = 0.001$  achieved the highest prediction accuracy value: 95.58%. Note that determining the optimal hyperparameter values separately for each feature group (and weight  $\beta_i$  associated with each feature group) is beyond the scope of this work. The results of the MK-A-SVM based individual learning and FL on Plant Site 1 are mentioned in Table 2.

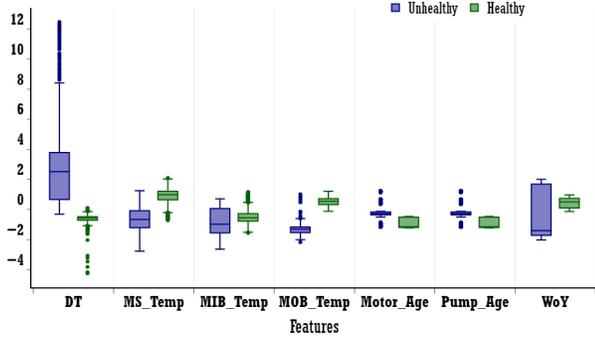
From Table 2, it is seen that individual models from each group achieved a performance of close to 100% in most of the MK-SVM models. This is a clear indication of overfitting in individual models, with the models being unable to predict other datasets or unseen data with the same level of accuracy. In addition, for some models, the accuracy of the test samples is higher than that of the training samples, since the test data were sometimes easier for the model to predict than the training data. Since the data labels are highly imbalanced, the F1 scores for all the models were above 98% (during individual training and after FL aggregation), indicating the prediction is not biased toward healthy class labels. After applying FL-based model aggregation and retraining each individual model, the accuracy levels decreased for most of the models, though performance remained at acceptable levels. As a comparison, FL aggregation was performed based on NNs, and the results are closely comparable with MK-A-SVM model. FL aggregation over several iterations can further improve overfitting, while also maintaining acceptable



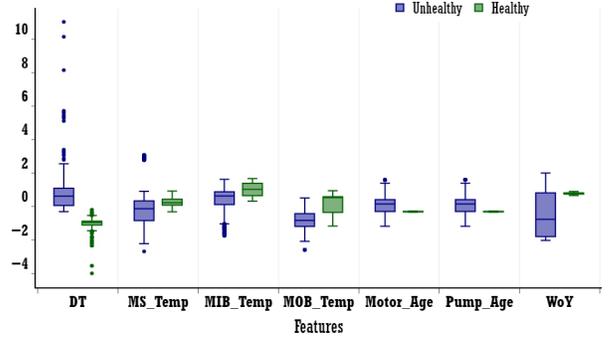
(a) Group 1: CWP 11A and 11B.



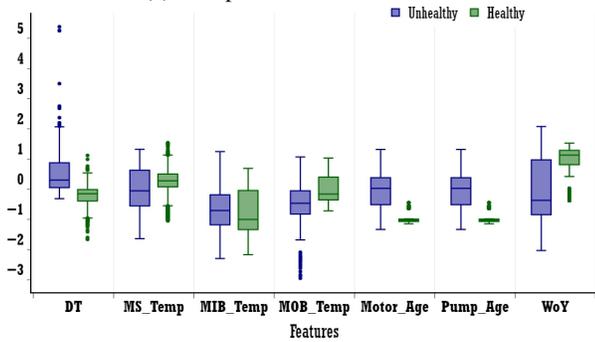
(b) Group 2: CWP 12A and 12B.



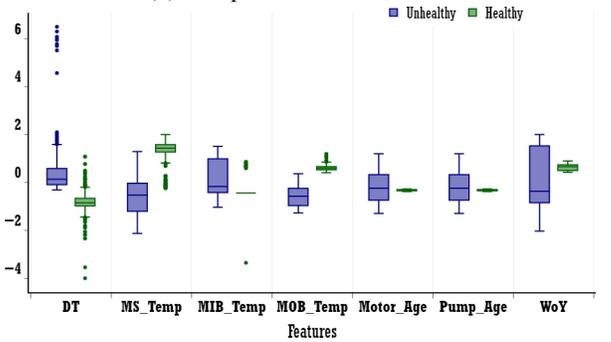
(c) Group 3: CWP 13A and 13B.



(d) Group 4: CWP 21A and 21B.

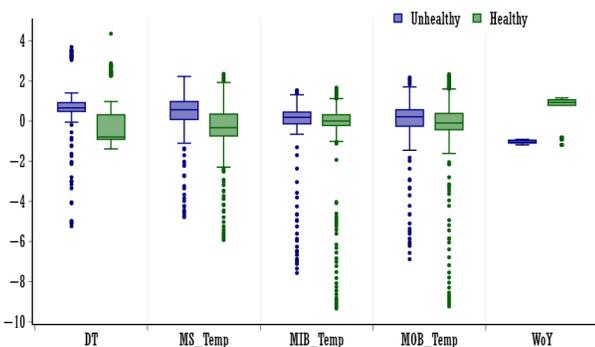


(e) Group 5: CWP 22A and 22B.

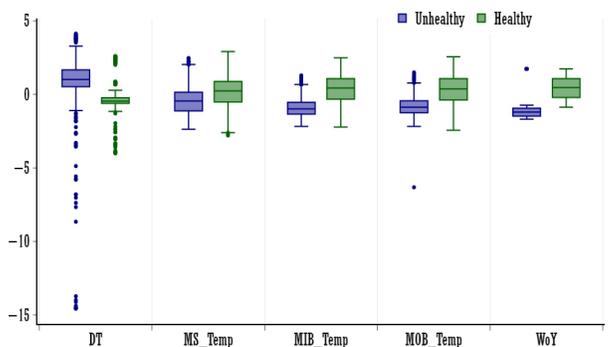


(f) Group 6: CWP 23A and 23B.

Figure 4. Distributions of “healthy” and “unhealthy” class labels for different CWP combinations from Plant Site 1.



(a) CWP A.



(b) CWP C.

Figure 5. Distributions of “healthy” and “unhealthy” class labels for different CWP combinations from Plant Site 2.

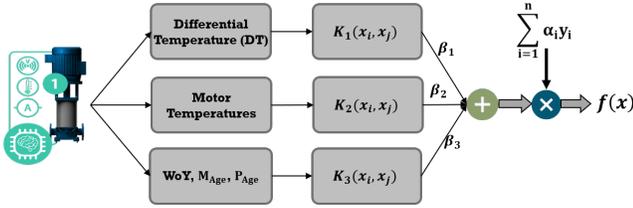


Figure 6. Feature-group-based MK-A-SVM framework for Plant Site 1.

performance of the diagnostic model. This exemplifies how FL-based model aggregation enables aggregation of diagnostic models from the component level to the plant level. The fact that the models are trained with limited datasets also impacts FL performance, which is anticipated to improve with larger training datasets.

### 3.3.1. MK-A-SVM Performance Analysis

Further work must be done to build more robust individual models that generalize more readily to previously unseen data. However, as noted above, sufficiently high accuracy is obtainable while retaining a uniform (and thus simpler) architecture for all individual models.

The FL approach showed much stronger test set performance for the CWPs than was seen in the individual phase. The added information afforded by examining all the pump data *en masse* provided clear advantages to the federated-model building process.

CWP Group	MK-A-SVM				NN			
	Individual Learning		Federated Learning		Individual Learning		Federated Learning	
	Train	Test	Train	Test	Train	Test	Train	Test
Group 1	97.1	95.6	98.3	93.7	100	88.3	98.0	96.3
Group 2	100	99.9	100	99.9	93.8	97.3	98.7	96.0
Group 3	100	98.9	100	98.9	100	97.6	100	94.3
Group 4	100	99.3	100	99.3	100	75.0	100	96.9
Group 5	100	98.9	98.6	91.8	100	75	100	98.1
Group 6	100	98.3	99.3	99.2	100	82.1	98.5	99.7

Table 2. FL performance (in %) on Plant Site 1 data, using MK-A-SVM and NN. Values are percent accuracy.

### 3.4. TL-based CWP Motor Health Prediction

TL aims to train a model on data from one domain (Plant Site 1), and then adapt that model to another (Plant Site 2) by partial or full retrain. To match the features used in the FL model, the *DP* was dropped, leaving only five features for TL. The features were again grouped based on measurement type for kernel selection, as discussed in Section 3.3. Thus, only three kernels were used in the TL for MK-SVM. For demonstrating TL on Plant Site 2 data, binary classification was considered, with the *healthy* and *unhealthy* class labels being considered

based on plant operation data. For the *unhealthy* state, waterbox fouling fault data were extracted. Data prior to the occurrence of waterbox fouling were considered *healthy*. Healthy and unhealthy samples extracted from CWPs A and C. From CWP A 1916 training samples and 966 testing samples were used while from CWP C 6502 training samples and 2558 testing samples were used. For the extracted samples, the master model from FL was used (both with and without retraining for comparison) to predict CWP conditions by using the Plant Site 2 data. Note that for TL without retraining, the extracted data are not split into training and test data; instead, all the data are considered test data, and the FL model (see Section 3.3) is used to predict the labels on complete test data.

For MK-SVM-based TL, the overall performance on both CWP A and CWP C data is around 80%. This approach involves using all the samples from CWP A and CWP C as test data in order to classify health using the master model from the FL framework. The performance dictates that the MK-SVM parameters must be further optimized to improve the prediction accuracy. Typically in TL, a small set of sample data is used to retrain the transferred model in order to fine-tune the model parameters for the new environment (i.e., Plant Site 2). For example, only 10–20% of the total number of samples will be used to retrain the model and optimize the parameters of the MK-SVM for the Plant Site 2 data. After retraining with 20% of the data, CWP A's performance did not improve, whereas CWP C's performance significantly improved (to higher than 95% accuracy). The performance of CWP A with TL indicates there were insufficient samples for building the ML model.

For comparison with TL, individual models were also trained on the Plant Site 2 data, with an 80:20 split between the training and test data. Individual model performance—particularly for CWP A—clearly shows the same overfitting trend as seen in the FL case. More samples for training are required in order to generalize the model and avoid overfitting. In comparison, the performance for NN is higher in MK-A-SVM for both CWP A and CWP C. This is also due to the fact that further enhancements to the optimization is necessary in terms of optimizing weights associated with multiple kernels.

Model	CWP Group	Individual Learning		TL without Retrain	TL with Retrain	
		Train	Test	Test	Train	Test
MK-A-SVM	CWP A	99.9	66.1	80.74	81.1	80.8
	CWP C	99.32	93.1	79.92	97.9	95.8
NN	CWP A	95.6	95.1	79.8	95.8	95.5
	CWP C	99.4	99.5	74.3	99.1	99.4

Table 3. TL performance (in %) on Plant Site 2 data, using MK-A-SVM and NN models from FL

#### 4. PRACTICAL CONSIDERATIONS

The practical applicability of the proposed FTL using MK-A-SVM can be considered for mainly three scenarios. Firstly, for a distributed asset/entity scenarios in which the data sharing is challenging due to privacy, communication bandwidth, and economical competence issues. Secondly, to implement data-driven decision-making on a new asset/entity by transferring previously trained model. Finally, to develop a model which could capture all the patterns of an asset/equipment behavior.

#### 5. CONCLUSION

A MK-A-SVM is developed to demonstrated FTL framework by considering the application of predicting CWP health conditions in NPP CWSs. The Plant Site 1 CWS fault signatures were used to develop FL based on MK-SVM. The federated models developed for Plant Site 1 were then used to estimate the state of health of the Plant Site 2 CWS (a process referred to as TL). The results obtained were comparable to predictive models individually trained on Plant Site 2 data. This demonstrated the significance of the FTL approach and the development feasibility through MK-A-SVM. The performance results were also compared with NN algorithm.

As a path forward, we continue to update and improve the MK-A-SVM framework to continuously optimize weight parameter associated with multiple kernels. In addition, future work will also expand the approach to multi-class classification.

#### 6. ACKNOWLEDGEMENTS

This research was made possible through funding from the U.S. Department of Energy's Light Water Reactor Sustainability program under the contract DE-AC07-05ID14517. We are grateful to William Walsh of DOE and Bruce P. Hallbert and Craig A. Primer at Idaho National Laboratory for championing this effort.

#### REFERENCES

- Agarwal, V., Manjunatha, A., Smith, J. A., Gribok, A. V., Yadav, V., Palas, H., . . . others (2021). *Machine Learning and Economic Models to Enable Risk-Informed Condition Based Maintenance of a Nuclear Plant Asset* (Tech. Rep.). Idaho National Lab.(INL), Idaho Falls, ID (United States).
- Aytar, Y., & Zisserman, A. (2011). Tabula Rasa: Model Transfer for Object Category Detection. In *2011 international conference on computer vision* (pp. 2252–2259).
- Carlsson, R. (2020). *Privacy-preserved federated learning: A survey of applicable machine learning algorithms in a federated environment*.
- Chen, Z.-Y., & Fan, Z.-P. (2012). Distributed customer behavior prediction using multiplex data: a collaborative mk-svm approach. *Knowledge-Based Systems*, 35, 111–119.
- Edemekong, P. F., Annamaraju, P., & Haydel, M. J. (2018). Health insurance portability and accountability act.
- Goodfellow, I. J., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Cambridge, MA, USA: MIT Press. (<http://www.deeplearningbook.org>)
- Li, X.-B., & Qin, J. (2017). Anonymizing and sharing medical text records. *Information Systems Research*, 28(2), 332–352.
- National Artificial Intelligence Initiative. (2020). *American Artificial Intelligence Initiative:Year One Annual Report*. Available at: <https://www.nitrd.gov/nitrdgroups/images/c/c1/American-AI-Initiative-One-Year-Annual-Report.pdf>.
- NRC. (2009). *General Electric Systems Technology Manual: Chapter 11.1 - Circulating Water System, Rev. 09-11*. Available at: <https://www.nrc.gov/docs/ML1125/ML11258A375.pdf>.
- Que, J., Jiang, X., & Ohno-Machado, L. (2012). A collaborative framework for distributed privacy-preserving support vector machine learning. In *Amia annual symposium proceedings* (Vol. 2012, p. 1350).
- Suykens, J. A., & Vandewalle, J. (1999). Least Squares Support Vector Machine Classifiers. *Neural processing letters*, 9(3), 293–300.
- Vaidya, J., Yu, H., & Jiang, X. (2008). Privacy-preserving SVM Classification. *Knowledge and Information Systems*, 14(2), 161–178.
- Yang, J., Yan, R., & Hauptmann, A. G. (2007). Cross-domain Video Concept Detection using Adaptive SVMs. In *Proceedings of the 15th ACM International Conference on Multimedia* (pp. 188–197).

#### BIOGRAPHIES



**Koushik A. Manjunatha** received the B.E degree in electronics and communication engineering from the Visvesvaraya Technological University, Belgaum, India, in 2013, the M.S degree from the Hong Kong University of Science and Technology, Hong Kong, in 2014 and the Ph.D. degree in electrical and computer engineering from the University of Alabama, Tuscaloosa, USA, in 2018. He is currently a Staff Research Scientist with Idaho National Laboratory (INL), Department of Instrumentation, Controls, and Data Science. He is also supporting the INL Wireless Security Institute (WSI) in various projects. His current research includes wireless communication and networking, industrial IoT, predictive maintenance, and condition-based monitoring. He has authored and co-authored more than 10 research articles with a U.S Patent.



**Vivek Agarwal** received the B.E. degree in electrical engineering from the University of Madras, Chennai, India, the M.S. degree in electrical engineering from the University of Tennessee, Knoxville, TN, USA, in 2001 and 2005, respectively, and the Ph.D. degree in

nuclear engineering from Purdue University, West Lafayette, IN, USA, in 2011. He is currently a Senior Research Scientist with INL, Department of Instrumentation, Controls, and Data Science. His current research interests include interdisciplinary, ranging from prognostics health management, wireless sensor networks, instrumentation and controls, mathematical modeling, computer vision, signal/image processing,

and nuclear engineering.

He is on the Editorial Board of the Journal of Pattern Recognition Research. He is a Publication Chair of the American Nuclear Society's Human Factors and Instrumentation and Controls Division and a Secretary of the American Society of Mechanical Engineer's Non-Destructive Prognostics and Diagnostics Division. He has authored or co-authored a book chapter and over 90 peer reviewed conference and journal publications.

He was awarded 2019 Presidential Early Career Awards for Scientists and Engineers; 2019 American Nuclear Society Ted Quinn Early Career Award; and 2016 Laboratory Director Early Career Achievement Award.