Chapter 4
Predictive Situation Awareness and Anomaly Forecasting in Cyber-Physical Systems

Masoud Abbaszadeh, Weizhong Yan, and Lalit K. Mestha

4.1 Introduction

Cyber-physical systems' (CPS) security has become a critical research topic as more and more CPS applications are making increasing impacts in diverse industrial sectors. Due to the tight interaction between cyber- and physical components, CPS security requires a different strategy from the traditional Information Technology (IT) security. Cyber-Physical Systems (CPS) are an integral system featuring strong interactions between its cyber- (e.g., networks and computation) and physical components (Khaitan and McCalley 2014). CPS applications have been making great impacts on many industrial sectors, including energy, transportation, healthcare, and manufacturing. With the development of Internet of Things (IoT), more and more devices with potential security vulnerabilities are linked to CPS, which makes CPS susceptible to adversary attacks (Yan et al. 2019). While progress with machine and equipment automation has been made over the last several decades, and systems have become "smarter", the intelligence of any individual cyber-physical system to predict failures (e.g., equipment malfunction, sensor faults, etc.), outages, degradation or slow drift in performance, and cyber-threats in real time to provide early warning is difficult. Several methods have been proposed for anomaly forecast and prognostic in different industrial control systems (Abbaszadeh and Marquez 2010, 2007; Allegorico and Mantini 2014; Chandola et al. 2009; Clifton et al. 2014; Ehlers et al. 2011; Gupta et al. 2008; Lamedica et al. 1996; Pimentel et al. 2014; Rigatos et al. 2021; Sridhar and Govindarasu 2014; Xue and Yan 2007; Zaher et al. 2009; Zimek et al.

M. Abbaszadeh (✉) · W. Yan
GE Research, Niskayuna, NY, USA
e-mail: masoud@ualberta.net

W. Yan
e-mail: yan@ge.com

L. K. Mestha
Genetic Innovations Inc. (work performed while at GE Research), Honolulu, HI, USA

M. Abbaszadeh and A. Zemouche (eds.), *Security and Resilience in Cyber-Physical Systems*, https://doi.org/10.1007/978-3-030-97166-3_4

2012). Although technology exists to predict when systems fail, approaches used to predict failures from a Prognostics and Health Management (PHM) perspective are not directly applicable to situation awareness of cyber-incidents since they (1) do not model large-scale transient data incorporating fast system dynamics (i.e., have improper estimation models) and (2) do not to process multiple signals simultaneously to account for anticipated changes in future times in system behavior accurately based on current and past data (i.e., have inaccurate decision thresholds/boundaries) (Mestha et al. 2017). Especially, when it comes to forecasting cyber-attacks propagation and impact, the difficulty is further compounded by not knowing attackers' intention and their next move for exploiting weakness/vulnerabilities in the system.

There can be various types of known attacks that a system may be subjected to such as espionage attacks, eavesdropping, denial-of-service attacks, zero dynamics attack, deception attacks (e.g., covert/stealthy attack), false data injection attack, replay attack, and the like, which are just a short sampling of potential threats that exist to cyber-physical systems (Park et al. 2019). These attacks will exhibit different levels of disclosure, disruption, and knowledge to be executed successfully, corresponding to adversaries' recourses, expertise, and intent. Also, cyber-hackers always invent many new ways to create malicious code and disrupt the operation of the physical system. Present condition monitoring technology used for failure detection, prediction, and monitoring or the threat detection technologies included inside information and operational technologies (IT and OT) does not adequately provide forecasting to protect assets from such attacks. There are many examples in physical systems (e.g., electric grid, ventricular assist devices, etc.), wherein early warning of only a few seconds may be sufficient to take actions that would protect vulnerable equipment or loss of life (Kokkonen et al. 2016; Nateghi et al. 2018a, b; Skopik et al. 2015).

Proper early warning generation could thwart an attack entirely or help neutralize its effects, such as damage to equipment or sustain the operation. The goal of this chapter is to provide an innovative predictive situational awareness framework in order to maintain high levels of reliability and availability, while continuing to retain expected performance against abnormalities created by the system faults or the adversary. Building upon our previous results on anomaly detection and forecasting (Abbaszadeh et al. 2018; Mestha et al. 2017; Yan et al. 2019), the predictive situational awareness framework developed in this chapter is based on dynamic weighted averaging of multi-model ensemble forecasts both for anomaly detection and isolation. Ensemble forecasting has been proven to be very efficient in forecasting complex dynamic phenomena such as wind and other weather conditions and Internet communication traffics (Cortez et al. 2012; Gneiting and Raftery 2005). In the context of an industrial control system, we use ensembles to cover the plant variations both in operating space and ambient conditions. The ensembles are selected using GMM clustering, which provides both centroid (i.e., respective operating points) and probability membership functions. A state-space model is developed for each ensemble of each monitoring node, which is used in an adaptive multi-step Kalman predictor to provide ensemble forecast in a receding horizon fashion. Then, the ensemble forecasts are fused via dynamic averaging. Dynamic model averaging has been shown to be superior to other ensemble methods such as Markov Chain Monte Carlo (MCMC) especially for large datasets (Koop and Korobilis 2012; McCormick et al. 2012; Raftery et al. 2010). It is an effective way for estimation of fusion of ensemble models.

We carry out all key processing in a high-dimensional feature space by analyzing time-series signals received from multiple system monitoring nodes (a combination of selected control system sensors and actuators), comparing the forecasted features with anomaly decision boundaries. The decision boundaries are computed for each individual monitoring node using machine learning techniques. We use Extreme Learning Machine (ELM) as our binary classification decision boundary. ELM is a special type of flashforward neural network recently developed for fast training (Huang et al. 2012). Numerous empirical studies and recently some analytical studies as well have shown that ELM has better generalization performance than other machine learning algorithms including Support Vector Machines (SVM) and is efficient and effective for both classification and regression (Huang et al. 2012; Huang 2014; Huang et al. 2006). It is worth mentioning that the framework presented here is not limited to using Kalman predictors or ELM classifiers and can be used along with other forms of linear or nonlinear time-series models, predictors, and classifiers.

The rest of the chapter is organized as follows. In Sect. 4.2, the overall forecasting framework is described. Sections 4.3 and 4.4 provide details of the ensemble modeling and receding horizon ensemble forecasting. In Sect. 4.4, we demonstrate our algorithm in a sensor attack of a gas turbine using a high-fidelity simulation environment, followed by conclusions in Sect. 4.5.

4.2 Forecasting Framework

In this section, we discuss the framework used for anomaly forecast and early warning generation. The framework is applicable to both cyber-driven and fault-driven incidents in a unified manner.

4.2.1 Digital Twin Simulation Platform

We demonstrate our approach on a utility-scale (250 MW maximum output) power-generating gas turbine. However, the methods and techniques presented in this work are applicable to any cyber-physical system. We have created both *normal* and *abnormal* (attack and fault) datasets using GE ARTEMIS high-fidelity power plant simulation platform. The simulation environment consists of a closed loop *Digital Twin* of a utility-scale power generation gas turbine, a very complex nonlinear and time-varying physics-based model with adaptive parameters and factors such as asset performance degradation due to ageing. The closed-loop system contains multiple control loops along with their interconnections as in a real gas turbine in the field.

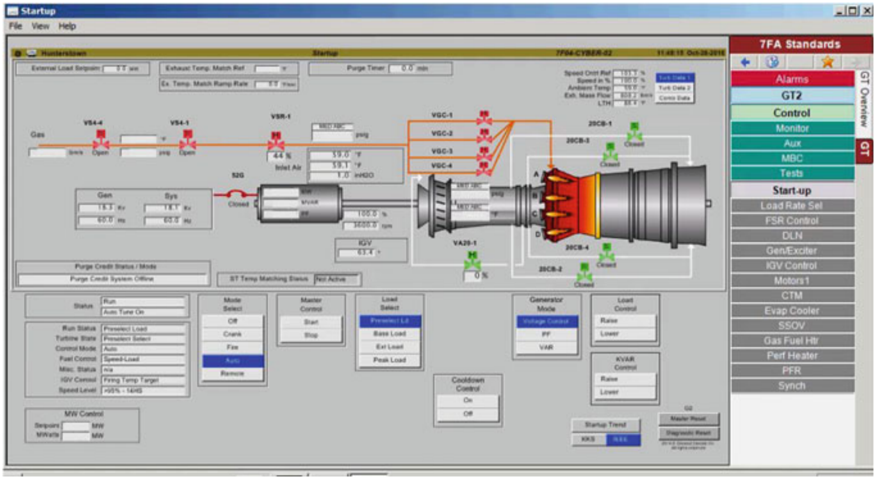

Fig. 4.1 Plant HMI used for dataset generation

The availability of such a platform enables realistic simulations of attack and fault scenarios, which, compared to normal operation data, are usually rare in the data collected from the field. This in turn enables deployment of high-performance supervised learning algorithms, as opposed to semi-supervised learning that only uses *normal* data. The *normal* dataset can be collected from the field, generated through simulations, or a combination of both. The *abnormal* dataset is synthesized utilizing the simulation platform. Our dataset consists of thousands of *normal* and *abnormal* time series of the monitoring nodes, resulting in over 2 million samples when projected into feature space. Figure 4.1 shows the HMI used for dataset generation.

4.2.2 Anomaly Forecasting Approaches

Depending on the scale of the system and outcome of the features dimensionality reduction, either the features or directly, the anomaly score may be forecasted. Each of these approaches have their pros and cons. Forecasting features make the forecasting framework independent of the decision boundary (i.e., the classifier), but it might be very difficult to do if the number of features is very large, they are highly non-linear, discontinuous, etc. On the other hand, forecasting the anomaly score directly simplifies the problem quite a bit, but makes the forecasting framework dependent on the particular anomaly classifier used (as will be described more).

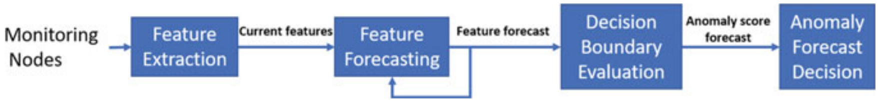

Fig. 4.2 Feature forecasting approach for anomaly prediction

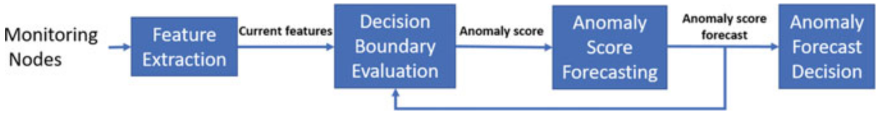

Fig. 4.3 Anomaly score forecasting approach for anomaly prediction

- **Forecasting Features:** In this approach, features are forecasted using dynamic models built for the time evolution of features, and the forecasted values are sent to classifier. A high-level depiction of the feature forecasting approach is shown in Fig. 4.2, where the feedback loop depicts the repetition of the forecasting for multi-step ahead prediction.
- **Forecasting Anomaly Score:** In this approach, the anomaly score is directly forecasted, as depicted in Fig. 4.3. Hence, instead of forecasting the features and sending the forecasted features to the classifier, the dynamic models are built for the anomaly score time series directly.

Note that assuming that there is only a single classifier for global detection and a single classifier for each local node, the anomaly score of each classification is a scalar, so such model would only have a single output. This significantly simplifies the dynamic models, reducing the number of model outputs from the number of features to only 1. Again, the anomaly score forecasting may be done both at the local and global levels. The states of the such a model may be the features or just the anomaly score. This method essentially simplifies the problem into forecasting a scalar. Note that as shown in Fig. 4.3, this brings the decision boundary into the forecasting loop. The dynamic models built in this approach will collectively represent the feature evaluation and the anomaly score evolution combined.

4.2.3 Dimensionality Reduction

Large-scale systems might have hundreds of monitoring nodes. Feature discovery techniques may lead to selection of several features for each node, resulting in a very large number of features to be forecasted. The following methods are used for dimensionality reduction in those large-scale systems.

4.2.3.1 Forecasting Features

In Feature Space

The number of features may be reduced using data dimensionality reduction methods such as PCA, ICA, and isoMap. This may be done for both the local and global levels. This enables the creation of scalable dynamic models.

In Dynamic State Space

Once the dynamic models are built, if the number of states (features and their lagged values) at each node or that of the global level is still large (normally > 50), dynamic model-order reduction techniques, such as balanced truncation or H_∞ norm-based model-order reduction, may be used to further reduce the dimensionality of the forecasting problem. The model-order reduction is performed using these two criteria:

- *Model Accuracy:* The error between the original model and the reduced-order model is less than a prescribed threshold using Hankel norm or H_∞ norm bounds. This determines the order of the reduced-order model. The error threshold may be selected by evaluating the forecasting accuracy of the reduced-order model or based on the preservation of the model observability (described below).
- *Model Observability:* The reduced-order model remains observable. In particular, in the original model, the features might be both the states and the outputs (i.e., an identity state to output mapping). Hence, the reduced-order model may have more outputs than states. The order and the model accuracy threshold then are selected in a manner to preserve the observability.

4.2.3.2 Forecasting Anomaly Score

If after dimensionality reductions in feature and/or state spaces, the order of the model is still high (normally > 50) or if the dimensionality reduction cannot be done in a way to properly satisfy the aforementioned criteria, then instead of forecasting the features, the anomaly score of the classifier is directly forecasted. In this approach, instead of forecasting the features and sending the forecasted features to the classifier, the dynamic models are built for the anomaly score time series directly. Note that the anomaly score is a scalar, so such model would only have a single output. This significantly simplifies the model reduces the number of model outputs (from the number of features to 1). Again, the anomaly score forecasting may be done both at the local and global levels. The states of such a model may be the features or just the anomaly score.

In the rest of this chapter, we will focus on forecasting using the feature forecasting approach, but the same tools and technique are applicable to the anomaly score forecasting as well.

4.2.4 Forecasting Process

The forecasting system is comprised of offline (training) and online (operation) modules. During the offline training, as shown in Fig. 4.4 the monitoring node datasets are used for feature engineering and decision boundary generation. To select the features, feature discovery techniques are used as described in Sect. 4.2.5. Then, state-space ensemble dynamic models are generated for the time evolution of features both at the global (for overall system status) and local (i.e., per monitoring node) levels as described in Sect. 4.3.1. At each level, dynamic forecasting models are generated for forecasting at three time scales, short term, mid-term, and long term, depending on the fundamental sampling time of the control system. Also, decision boundaries are computed both at the local and global levels as binary classifiers using machine learning techniques as described in Sect. 4.3.5.

The online module of forecasting system is shown in Fig. 4.5. First, each monitoring node signal goes through real-time feature extraction to create real-time feature time series. The features are computed using a sliding window over the monitoring node signals. In the next step, the extracted feature time series are inputted to multi-step predictors, both at the local and global levels. Using the models generated in the training phase and the multi-step predictors, future values of the feature time series are forecasted, both for local and global features, in three time scales:

1. **Short-term feature forecast:** future values of the global and local features (e.g., up to several seconds).
2. **Mid-term feature forecast:** future values of the global and local features (e.g., up to several minutes).
3. **Long-term feature forecast:** future values of the global and local features (e.g., up to several days).

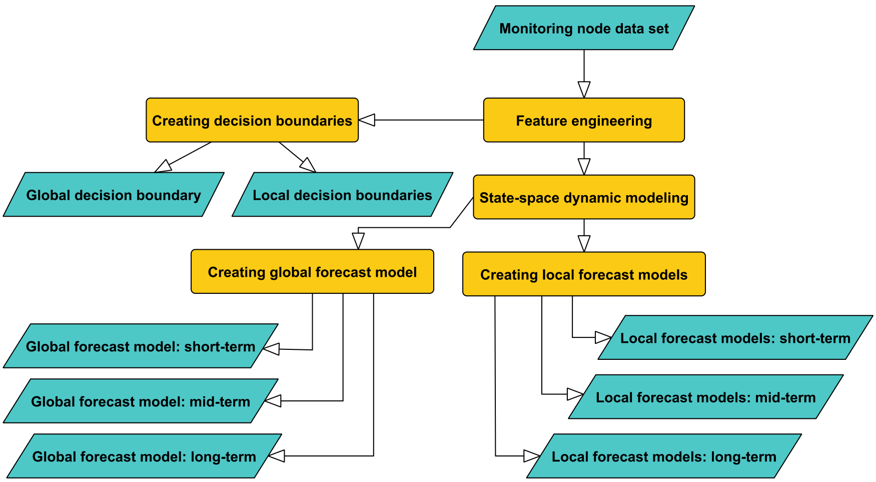

Fig. 4.4 Anomaly forecast systems: offline training

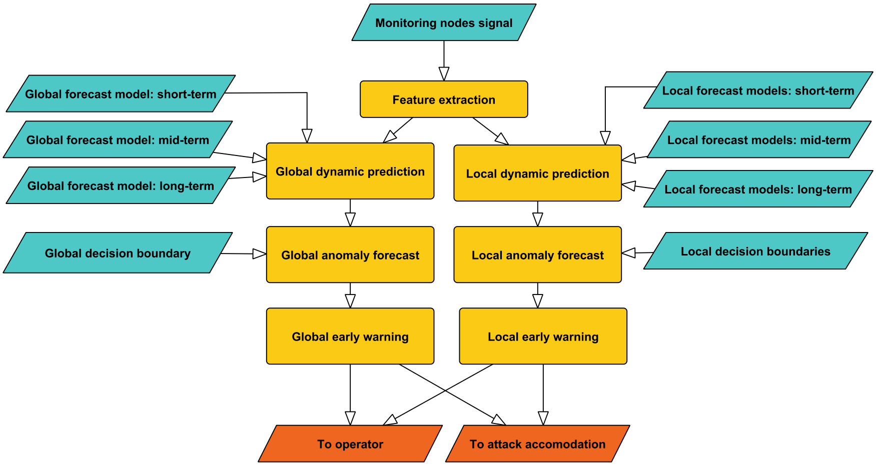

Fig. 4.5 Anomaly forecast systems: online operation

While the short-term forecast is useful for rapid detection of the incipient and transient faults and cyber-attacks, mid-term and long-term forecasts are helpful in early detection of stealthy cyber-attacks as well as component failures due to degradation. The forecasted outputs of models (*aka*, future values of the features) are compared to the corresponding decision boundaries for predictive anomaly detection. While comparing the feature vectors to the decision boundary, estimated time to cross the decision boundary will provide information for future anomaly. If a future anomaly is detected, an early warning is generated in the operator display with anticipated time to reach anomalous state and a message is sent to the automatic accommodation system (such as an attack-tolerant or fault-tolerant resilient control mechanism) for potential early engagement. The current values of the features along with the decision boundaries provide a deterministic decision of the current status of the system, while the forecasted features provide a probabilistic decision on the future system status. The global feature forecast is used for system-level anomaly detection (overall system health status) and the local feature forecasts are used for anomaly isolation (locate the abnormal nodes of the system). Using this framework a predictive situation awareness is established for the system.

4.2.5 Feature Discovery

The proposed sensing approach should handle many types of inputs from multiple heterogeneous data stream in complex hyper-connected systems. Signals from time domain are converted to features using multi-modal multi-disciplinary (MMMD) feature discovery framework employed as in machine learning discipline (Yan and

Yu 2015). A "feature" may refer to, for example, mathematical characterizations of data and is computed in each overlapping batch of data stream. Examples of features as applied to sensor data can be classified broadly into knowledge-based, shallow, and deep features.

Knowledge-based features use domain or engineering knowledge of physics of the system to create features. These features can be simply statistical descriptors (e.g., max, min, mean, variance), and different orders of statistical moments, calculated over a window of a time-series signal and its corresponding FFT spectrum as well. Shallow features are from unsupervised learning (e.g., k-means clustering), manifold learning, and nonlinear embedding (e.g., isoMap, locally linear embedding), low dimension projection (e.g., principal component analysis, independent component analysis), and neural networks, along with genetic programming and sparse coding. Deep learning features can be generated using deep learning algorithms which involve learning good representations of data through multiple levels of abstraction. By hierarchically learning features layer by layer, with higher level features representing more abstract aspects of the data, deep learning can discover sophisticated underlying structure and features. Still other examples include logical features (with semantic abstractions such as "yes" and "no") and interaction features.

Several methods have been proposed in the literature for feature selection and features ranking of ELM for classification and regression problems)(Wang et al. 2018; Yin et al. 2017). Machine learning-based attack and fault-detection algorithms can in general incorporate large number of features, with the number of features selected based on the Receiver Operating Characteristic (ROC) curve analysis to optimize the detection and false alarm rates. Different number of features might be selected for each individual monitoring node, however, from a systems engineering perspective, to streamline the design, it is preferred to choose the same type and number of features for all nodes, except if a particular node needs special treatment. In this work, for each monitoring node of the gas turbine, we have selected five features which are a combination of statistical and temporal features. At the system level, we have also selected multivariate features which consist of cross-correlations between critical measurements.

For the forecasting at the *global* level (i.e., the system level), the global feature vector is formed by stacking up the local feature vectors of the individual monitoring nodes. For large-scale systems with many monitoring nodes, the size of the global feature vector might be very large, and thus it can be reduced by dimensionality reduction techniques such as Principal Component Analysis (PCA).

4.3 Ensemble Forecasting

The forecasting framework described in the previous section is based on ensemble models which are used in adaptive Kalman predictors to provide ensemble feature forecasts. The ensemble feature forecasts are then averaged using dynamic weights to provide the overall feature forecast. The process described in the section is applied separately and in parallel to the local features of each individual monitoring node, as well as to the global feature vector.

4.3.1 Ensemble Modeling in Feature Space

The forecasting models at each time scale (short term, mid-term, and long term) consist of a collection of ensemble models, each providing an ensemble forecast of the features. These ensembles ensure coverage of whole operating space with operational and ambient condition variations. The operating space is partitioned through Gaussian Mixture Model clustering. A mixture model is a statistical model for representing datasets which display behavior that cannot be well described by a single standard distribution. It allows a complex probability distribution to be built from a linear superposition of simpler components. Gaussian distributions are the most common choice as mixture components because of the mathematical simplicity of parameter estimation as well as their ability to perform well in many situations (Dempster et al. 1977).

Gaussian mixture models can be used for stochastic data clustering. To select the operating point associated with each ensemble model, we use GMM clustering in the feature space. The GMM clustering partitions the operating space (projected into feature space) into multiple clusters each represented by a multivariate Gaussian process described by a mean (centroid) and a covariance matrix. The centroid of each cluster represents the operating point for each ensemble model, while its covariance matrix establishes a probabilistic membership function. The Expectation Maximization (EM) algorithm is a maximum likelihood estimation method that fits GMM clusters to the data. The EM algorithm can be sensitive to initial conditions, and therefore we repeat the GMM clustering multiple times with randomly selected initial values and choose the fit that has the largest likelihood.

Since GMM is a soft clustering method (i.e., overlapping clusters), all points in the operating space belong to all clusters with a membership probability. As an example, Fig. 4.6 shows the GMM clustering at the global level for our gas turbine dataset, where the horizontal axis is the number of clusters and the vertical axis is the Bayesian Information Criterion (BIC) computed for different covariance structures per number of clusters. BIC provides a right trade-off between model accuracy and complexity, thus avoiding over-fitting to the training dataset. The model with the lowest BIC is selected. As seen in the figure, the optimal clustering is achieved with seven clusters with Gaussian models having full (i.e., non-diagonal) unshared (i.e., individual) covariance matrices.

Remark 4.1 Note that at the local node level, GMM clustering can be done for each monitoring node separately, resulting in different number of ensembles for each monitoring node.

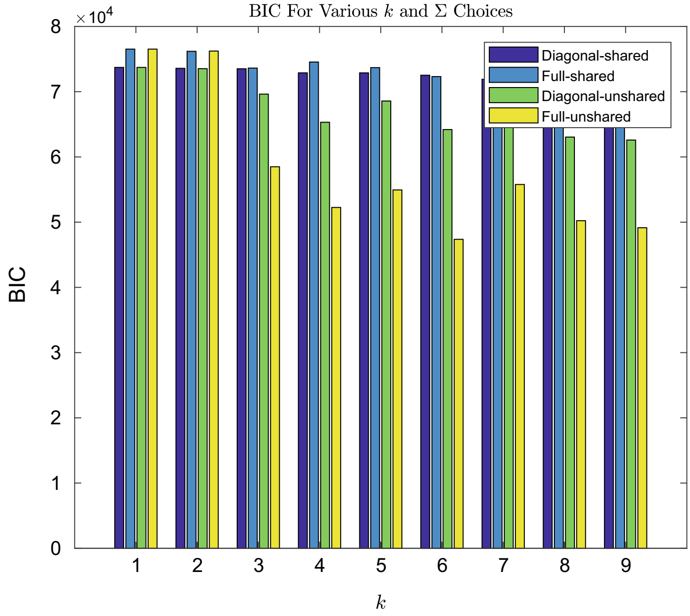

Fig. 4.6 BIC for GMM clustering. ©2018 IEEE. Reprinted, with permission, from (Abbaszadeh et al. 2018)

4.3.2 Adjusting Cluster Centroids to Physical Points

The GMM clustering may select centroid of the clusters as any arbitrary real-valued vector in the feature space. However, since centroids are deemed as operating points to create state-space models, they need to be associated with actual physical operating points of the system. This can be achieved in two ways:

- **Mixed-integer programming for EM:** GMM clustering uses Expectation Maximization (EM) algorithm for cluster optimization. Rather than running the standard EM, one can use a modified EM to enforce searching for centroids only among the points given in the training dataset (which are readily physical points of the systems). This is essentially similar to running k-medoids clustering rather than k-means clustering but in a GMM framework. This normally requires mixed-integer programming and is feasible for small- and medium-sized datasets.
- **Heuristics based:** Adjust the centroids of GMM into closest point in the dataset in post-processing. This is particularly efficient for large datasets. Moreover, since large datasets comprise of high granularity data, the distance of the initial centroid to the closest point in the data is often small and negligible. This can be further validated by putting a threshold on such point adjustments. As a result of centroid adjustment, the covariance matrices of each GMM clusters are also adjusted. Suppose that μ_i and Σ_i are the centroid and covariance of the i-th cluster, respectively,

and the closest point to μ_i is $\bar{\mu}_i$ whose Euclidean distance to μ_i in feature space is d_i, i.e., $\bar{\mu}_i - \mu_i = d_i$. Then, we have

$$\mu_i \to \bar{\mu}_i = \mu_i + d_i, \tag{4.1}$$

$$\Sigma_i \to \bar{\Sigma}_i = \Sigma_i + d_i {d_i}^T, \tag{4.2}$$

which means that the Gaussian model associated with the i-th cluster is adjusted from $\mathcal{N}(\mu_i, \Sigma_i)$ to $\mathcal{N}(\bar{\mu}_i, \bar{\Sigma}_i)$.

In this work, since we have a large-scale dataset with high resolution, we use the heuristics-based method described above to adjust the cluster centroids to the nearest physical point as needed.

4.3.3 Dynamic Modeling

Once the number and structure of the clusters are determined, the cluster centroids are selected as the representative operating points of the system, and a dynamic model is developed for the time series of each monitoring node of each operating point (aka ensemble models). The time-series dynamic modeling can happen using different linear or nonlinear time-series modeling techniques. The choice of linear versus nonlinear modeling can be made by assessing the feature time series through linearity tests such as those described in Harvey and Leybourne (2007). For linear time-series data, Vector Autoregressive (VAR) models are proved to be a powerful tool. For nonlinear time-series modeling, nonlinear autoregressive models, Volterra series, or recurrent neural networks (such as LSTM) could be used. In this work, due to the good fit of the feature time-series data in the linear space, the time series are modeled as VAR models. A VAR model is a multivariate autoregressive model that relates the current value of the time series to its previous values through a linear mapping plus a constant bias term. Essentially, this is not an input–output modeling but a time-series output modeling, assumed to be derived by an unknown stochastic input. VAR models are vastly used for modeling of time-series signals (Johansen 1995), similar to what we measure here from our monitoring nodes. The number of lags required for each VAR model is again determined using BIC. This determines the order of the models, which could be different among the ensembles. The parameters of the VAR models are identified, and the models are then converted into the standard state-space form for each ensemble, as follows:

$$x[k+1] = Ax[x] + Bu[k] + Qe[k], \tag{4.3}$$

$$y[k] = Cx[k] + v[k], \tag{4.4}$$

where x is the vector of monitoring node features and their lagged values, u is a fictitious Heaviside step function capturing the bias term of the VAR model, e is a zero-mean Gaussian white noise with Identity covariance, $E[ee^T] = I$, and Q is the process noise covariance. The model outputs y here are the monitoring node features with some assumed measurement noise v, whose covariance R is adaptively updated, as will be described later.

If the model is VAR(1), i.e., having one lag, then $C = I_q$, where q is the number of local features for each individual monitoring node (here, for our gas turbine application, $q = 5$). In general, for a VAR(p) model with p lags, per ensemble, per node, we have

$$x[k] = \left[x_1^f[k] \cdots x_q^f[k] \cdots x_1^f[k-p+1] \cdots x_q^f[k-p+1] \right]^T, \tag{4.5}$$

$$A = \begin{bmatrix} A_1 & A_2 & \cdots & A_{p-1} & A_p \\ I_q & 0_q & \cdots & 0_q & 0_q \\ 0_q & I_q & \cdots & 0_q & 0_q \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_q & 0_q & \cdots & I_q & 0_q \end{bmatrix}, \tag{4.6}$$

$$B = \begin{bmatrix} b & \underbrace{0_q \cdots 0_q}_{1,\ldots,p-1,\ p>1} \end{bmatrix}^T, \quad C = \begin{bmatrix} I_q & \underbrace{0_q \cdots 0_q}_{1,\ldots,p-1,\ p>1} \end{bmatrix}, \tag{4.7}$$

where $x_i^f, i = 1, \ldots, q$ are the local features for an individual monitoring node.

The initial value of R is set using noise characteristics of the raw measurements, linearly projected into the feature space as follows. Suppose y^r is the raw measured value of an individual monitoring node and the scalar v^r is the corresponding measurement noise, $y^r[k] = r[k] + v^r[k]$, where r is the true value of the signal and v^r is a zero-mean Gaussian white noise with variance σ. The feature vector y corresponding to this particular monitoring node is the projection of y^r in the feature space. Suppose that $\mathscr{F} : R \to R^q$ is the mapping from the raw signal measurement to its features. The raw data is projected into the feature space as

$$\left[x_1^f[k] \cdots x_q^f[k] \right]^T = Cx[k] = \mathscr{F}(r[k]). \tag{4.8}$$

Then we have

$$\begin{aligned} y[k] &= \mathscr{F}(y^r[k]) = \mathscr{F}(y[k] + v^r[k]) \\ &\simeq \mathscr{F}(r[k]) + \frac{\partial \mathscr{F}}{\partial r}_{|r=r[k]} v^r[k] \triangleq Cx[k] + J(r[k])v^r[k] \\ &\triangleq Cx[k] + v[k], \end{aligned} \tag{4.9}$$

where v is the derived measurement noise in the feature space and J is the Jacobian of $\mathscr{F}$ with respect to r. From (4.9), it is clear that the covariance of v is $\sigma J(r[k])^T J(r[k])$. Note that the scalar measurement noise of an individual monitoring node in the signal space is projected into a multivariate noise in the feature space. The linear approximation of noise maintains the noise zero-mean Gaussian white. This approximation is only used for the initial guess of the covariance, since after the initialization it is adaptively estimated.

As mentioned before, the number of such state-space models for each monitoring node equals the number of corresponding GMM clusters. The order of the state-space models remains the same within the ensembles of one particular node, but may differ from one node to another depending on the number of local features selected for each node.

4.3.4 Dynamic Ensemble Forecast Averaging

Within our proposed framework, different type of the predictors may be used to provide ensemble forecasts. This simplest predictor could be the model (4.4) itself, repeatedly executed using previous predictions through the prediction horizon. Although simple, this approach quickly leads to large prediction errors since there is no control or adjustment over the error covariance. Another simple approach is to use parametric prediction methods such as exponential smoothing. They provide certain level of parameter tuning capability but still suffer from proper error control. As such, although both approaches are applicable within our proposed framework, they are both limited to only very short prediction horizons.

In this chapter, an adaptive Kalman predictor (AKP) is applied to each ensemble model to provide ensemble forecasts. The process noise covariance of the Kalman predictor is readily available as Q as in (4.4). It is worth mentioning that for nonlinear models, an adaptive EKF or UKF can be used still in a similar fashion within the same framework. The covariance of the measurement noise of each AKP is estimated adaptively using the method proposed in Ding et al. (2007); Rutan (1991) as follows.

$$\hat{v}[k] = y[k] - C^T\hat{x}[k|k-1], \tag{4.10}$$

$$R[k] = \begin{cases} \sigma J(r[k])^T J(r[k]) & k = 1, \ldots m \\ \frac{1}{m}\left[\sum_{j=1}^{m} \hat{v}[k-j]\hat{v}[k-j]\right] \ldots & \\ \qquad -C^T P^e[k|k-1]C & k > m, \end{cases} \tag{4.11}$$

where $\hat{v}$ is the predictor innovation sequence, m is the width of an empirically chosen rectangular smoothing window for the innovations sequence, and P^e is the prediction error covariance matrix. The smoothing operation improves the statistical significance of the estimator for $R[k]$, as it now depends on many residuals. Figure 4.7 shows the block diagram for dynamic ensemble forecast averaging, where N is the number of ensembles corresponding to a monitoring node and P is the forecasting horizon. It is worth mentioning that the ensemble modeling (GMM clustering and state-space system identification) is performed using *normal* dataset only as the models capture the normal operational behavior of the system, while the decision

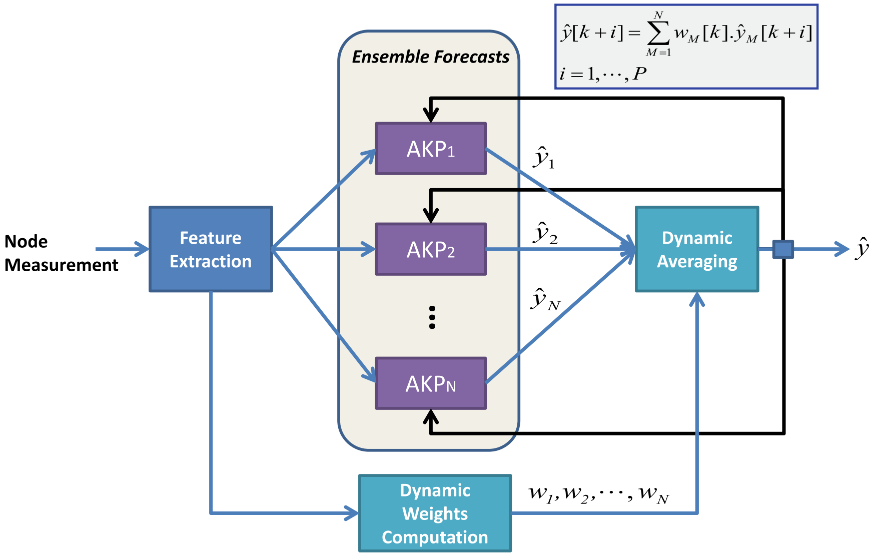

Fig. 4.7 Block diagram for dynamic ensemble forecast averaging. ©2018 IEEE. Reprinted, with permission, from (Abbaszadeh et al. 2018)

boundaries are computed using both *normal* and *abnormal* datasets. Furthermore, to emphasize the recent data, a forgetting factor is used in the covariance matrix update of each of the Kalman predictors.

The forecasting horizon of the multi-step forecasts can be determined using simulations, based on the prediction error and some threshold on the confidence interval. As the forecasting horizon extends, the confidence interval expands and eventually passes the threshold. Each AKP provides an ensemble forecast $\hat{y}_M$, $M = 1, \ldots, N$. The ensemble forecasts are dynamically averaged using weight $w_1, \ldots, w_N$. The weights are time varying and computed as normalized probabilities using the multivariate Gaussian probability density functions with mean and covariances computed during the GMM clustering. Suppose the real-time value of the feature vector is $x[k]$, and the mean and covariance of each Gaussian cluster are μ_i and Σ_i, respectively. Then we have

$$d_M[k] = \mathbf{Pr}\left\{x[k] \mid x[k] \sim \mathcal{N}(\mu_i, \Sigma_i)\right\}, \quad M = 1, \ldots, N,$$

$$w_M[k] = \frac{d_M[k]}{\sum_{M=1}^{N} d_M[k]}, \quad \sum_{M=1}^{N} w_M[k] = 1,$$

$$\hat{y}[k+i] = \sum_{M=1}^{N} w_M[k]\hat{y}_M[k+i], \quad i = 1, \ldots, P.$$

The ensemble averaged forecast $\hat{y}[k+i]$ is returned back to the AKPs as the next input, to provide the next-step forecast receding horizon fashion, up to the forecasting horizon.

Remark 4.2 Alternatively, the ensemble forecast of each AKP, $\hat{y}_M[k+i]$, could be fed back for multi-step forecasting, but feeding back $\hat{y}[k+i]$ to all AKPs is better, since it is a better prediction of system's true behavior.

4.3.5 *Receding Horizon Anomaly Forecast*

The forecasted features, $\hat{y}$, are compared to a decision boundary for anomaly forecasting in each node. At each sampling time, a P-step ahead forecast of the features is computed using the dynamic ensemble averaging method. In the next sampling time, the horizon moves forward (recedes) by one time step, and a new forecast is computed through the new forecasting horizon.

$$
\begin{aligned}
k: &\ \Big[\hat{y}[k+1], \hat{y}[k+2], \ldots, \hat{\mathbf{y}}[\mathbf{k}+\mathbf{P}]\Big],\\
k+1: &\ \Big[\hat{y}[k+2], \hat{y}[k+3], \ldots, \hat{\mathbf{y}}[\mathbf{k}+\mathbf{P}+1]\Big],\\
k+2: &\ \Big[\hat{y}[k+3], \hat{y}[k+4], \ldots, \hat{\mathbf{y}}[\mathbf{k}+\mathbf{P}+2]\Big],\\
&\ \ldots
\end{aligned}
$$

At each sampling time, the last forecast in the horizon $\hat{y}[k+P]$ is compared to the decision boundary. This is similar to the Model Predictive Control (MPC), except that in MPC, at each sampling time, the first control action in the horizon is applied to the system.

Each decision boundary is computed by training an Extreme Learning Machine (ELM) as a binary classifier in a supervised training framework. ELM is a special type of feed-forward neural networks recently introduced (Huang et al. 2012). ELM was originally developed for the single hidden layer feed-forward neural networks (SLFNs) and was later extended to the generalized SLFNs where the hidden layer need not be neuron alike (Huang et al. 2013). Unlike in traditional feed-forward neural networks where training the network involves finding all connection weights and bias, in ELM, connections between input and hidden neurons are randomly generated and fixed, that is, they do not need to be trained. Thus training an ELM becomes finding connections between hidden and output neurons only, which is simply a linear least squares problem whose solution can be directly generated by the generalized inverse of the hidden layer output matrix (Huang et al. 2012).

Because of such special design of the network, ELM training becomes very fast. The structure of a one-output ELM networks is depicted in Fig. 4.8. Assume the

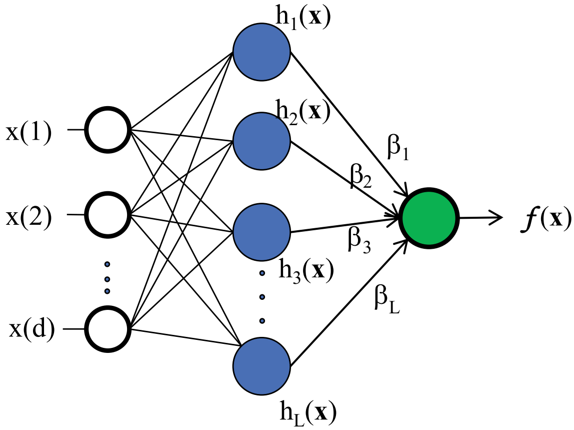

Fig. 4.8 An ELM network with one output. ©2018 IEEE. Reprinted, with permission, from (Abbaszadeh et al. 2018)

number of hidden neurons is L. Then the output function of ELM for generalized SLFNs is

$$f(x) = \sum_{j=1}^{L} \beta_j h_j(x) \triangleq \mathbf{h}(x)\boldsymbol{\beta}, \tag{4.12}$$

where $h_i(x) = G(\phi_i, b_i, x)$ is the output of jth hidden neuron with respect to the input x; $G(\phi, b, x)$ is a nonlinear piecewise continuous function satisfying ELM universal approximation capability theorems (Huang et al. 2006); β_j is the output weight vector between jth hidden neuron to the output node; and $h(x) = [h_1(x), \ldots, h_L(x)]$ is a random feature map, mapping the data from d-dimensional input space to the L-dimension random feature space (ELM feature space).

The objective function of ELM is an equality-constraint optimization problem, to minimize both the training errors and the output weights, which can be written as

$$\textit{Minimize: } \mathbf{L}_p = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}c\sum_{i=1}^{N_d} \xi_i^2 \tag{4.13}$$

$$\textit{s.t.: } \mathbf{h}(x_i)\boldsymbol{\beta} = l_i - \xi_i, \quad i = 1, \ldots, N_d, \tag{4.14}$$

where ξ_i is the training error with respect to the training sample x_i, l_i is the label of the ith sample, and N_d is the number of training samples (in the *normal* and *abnormal* datasets combined). The constant c controls the trade-off between the output weights and the training error.

Based on the Karush–Kuhn–Tucker (KKT) condition, we can have the analytic solutions for the ELM output function f for non-kernel and kernel cases, respectively (see Huang (2014) for details). Since kernel two-class ELM learns a nonlinear hyperplane, it generally works better than non-kernel two-class ELM. Therefore, we have used a kernel ELM using a Radial Basis Function (RBF) kernel.

The distance d of any point (a sample) to the hyperplane constructed by the ELM can conveniently serve as an anomaly score, that is, the larger the distance, the more likely the sample is abnormal. Here f is an anomaly score function whose sign (compared to a threshold, normally, zero) determines the binary classification decision on the system status. We have trained the ELM such that *normal* samples generate negative scores.

4.3.6 Committed Horizon Anomaly Forecast

An extension to receding horizon prediction is committed horizon prediction (Chen et al. 2019). It considers a so-called commitment level $V < P$, and instead of committing to only one estimate obtains the final predicted value at time k by combining (e.g., via a weighted average) the estimates of the V receding horizon instances from time $k + 1$ to $k + V$. Therefore, with a P-step look ahead, the effective prediction horizon is $P - V$. In other words, at each time instance, there is a delay of V sampling times to get the forecast of P steps ahead. Committed horizon prediction tends to give better estimates because it accounts for both future and past information, and also provides an additional mechanism to adjust the trade-off between delay and prediction accuracy (Chen et al. 2019). However, it reduces the effective prediction horizon, and hence its capability to generate rapid early warnings for anomaly detection applications. Nevertheless, the committed horizon prediction approach may still be effectively used for short-term forecasting, especially if the sampling rate is much faster than the system dynamics.

4.4 Predictive Situation Awareness

In general, predictive situation awareness has three main elements (Endsley 1995):

1. *Perception*: monitoring the environment.
2. *Comprehension*: understanding the current situation.
3. *Projection*: predicting the evolution of the situation.

Figure 4.9 depicts the block diagram of situation awareness modules in this work. Here, the *perception* element consists of collecting and pre-processing data from the monitoring nodes including feature extraction and any dimensionality reduction. *Comprehension* is provided by the anomaly detection module supplying the current system status, and *Projection* is performed through anomaly forecasting.

As data is processed in stream or batch modes, anomaly detection provides an instance decision on the current system status, which is either *normal* or *abnormal* (attack or fault). Before an anomaly happens, the current system status is normal

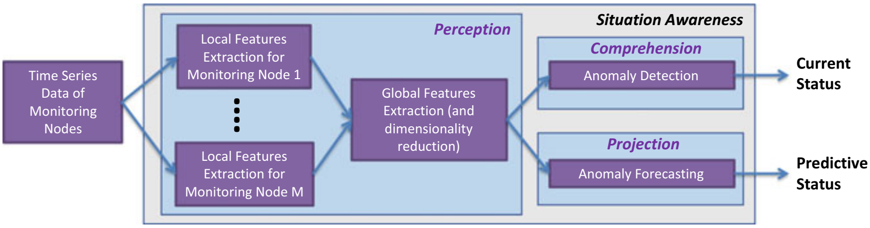

Fig. 4.9 Situation awareness block diagram

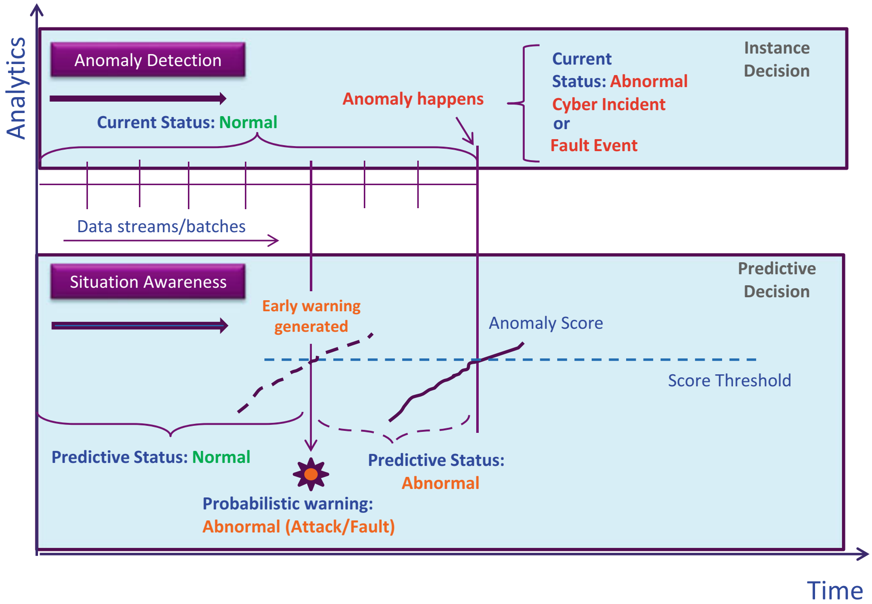

Fig. 4.10 Deterministic and probabilistic decisions for situation awareness

and it remains normal until an anomaly actually occurs. The anomaly detection algorithm detects an anomaly once it happens based on an anomaly score calculated at the current time instant passing a prescribed threshold (which could be fixed or adaptive itself). In addition, the situation awareness provides a predictive decision and generates early warnings. At each time instant, the forecasting algorithm projects the current status into future using stochastic dynamic forecasting described in the previous sections. The predictive status remains normal until the predicted anomaly score passes the threshold. Once an early warning is generated, future forecasting still continues, with a probabilistic decision on the predicted systems status based on anomaly score. The anomaly score increases between the time an early warning is generated and the time an anomaly actually happens, at which point the current status also reflects the anomaly. The concept is depicted in Fig. 4.10.

4.5 Simulation Results

To generate the early warning, the forecasted outputs of models (aka, future values of the features) are compared to the corresponding decision boundaries for anomaly detection. While comparing the feature vectors to the decision boundary, estimated time to cross the decision boundary will provide information for future anomaly. Figure 4.11 shows the early warning generation for a DWATT (gas turbine-generated power) sensor false data injection attack based on a short-term (10 s ahead) forecast. It is worth mentioning that this attack case was not included in the training dataset so this simulation represents an independent cross-validation of the algorithm. The attack is injected at $t = 129$. Without forecasting, the detection algorithm detects it at $t = 150$. With the 10-s ahead forecast, the forecasted features pass the local boundary at $t = 140$, at which point an early warning is generated. As seen, the forecasting is able to generate early warning 10 s ahead of the actual detection happening. With

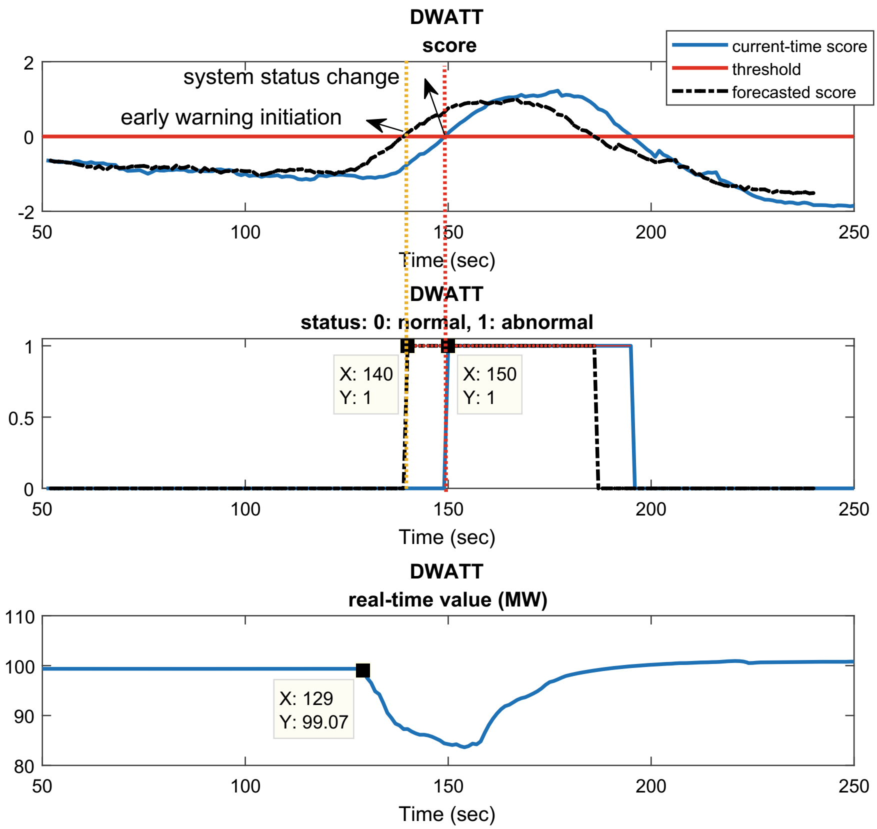

Fig. 4.11 Anomaly forecast and early warning generation for DWATT sensor. ©2018 IEEE. Reprinted, with permission, from (Abbaszadeh et al. 2018)

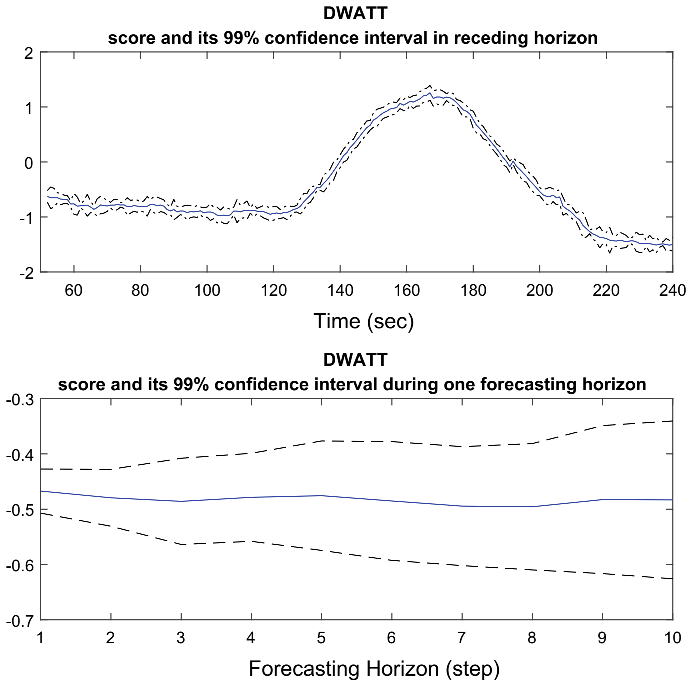

Fig. 4.12 Forecasted score for DWATT and confidence intervals for the whole simulation time and one forecasting horizon. ©2018 IEEE. Reprinted, with permission, from (Abbaszadeh et al. 2018

this technology, we are able to compensate for the delay in detection and generate early warning in the very early stage of an attack. Similarly, once the disturbance rejection control of the gas turbine brings the system back into the *normal* region, the forecasting algorithm is able to predict that before the actual system status goes back to *normal*. Note that here we are forecasting the features directly, and the anomaly score indirectly by passing the forecasted features through the decision boundary. Hence, the confidence intervals of ensemble feature forecasts are readily available from the AKPs, while those of the averaged forecasts and the anomaly score are computed using interval arithmetic (Bland and Altman 1996). The forecasted features are computed in a receding horizon with a forecasting horizon of 10 s (i.e., 10-step ahead forecasts are used for anomaly decision). In every sampling time, a 10-s forecast is computed along with its confidence interval. In the next sampling time, a new receding horizon forecast is computed, sliding the previous horizon by 1 s. Figure 4.12 shows the forecasted score for DWATT and confidence intervals for the whole simulation time and one forecasting horizon, respectively. The simulation is performed for 250 s (thus, 240 s of 10-step ahead receding horizon forecasts).

4.6 Conclusions

In this work, a framework for anomaly forecasting and early warning generation in industrial control systems is proposed based on a new feature-based dynamic ensemble forecasting method. The cyber-physical system anomalies addressed here could be either of cyber-incident or of naturally occurring faults/failures nature. The ensembles are selected via GMM clustering based on BIC criterion, each representing an operating point of the system. The cluster centroids are adjusted to the nearest physical points in the training dataset, and the associated covariance matrices are updated accordingly. Ensemble forecasts are provided by adaptive Kalman predictors applied to dynamic VAR models in the feature space, and fused through dynamic averaging, while the averaging weights are calculated using the Gaussian clusters mean and covariance matrices. The forecasts are multi-step and performed on different time scales in a receding horizon fashion. To predict the future status of the system, the forecasts are compared to decision boundaries computed using extreme learning machines. High-fidelity simulations on a GE gas turbine digital twin platform show the efficacy of our approach.

Acknowledgements This material is based on work supported by the US Department of Energy under Award Number DE-OE0000833.

References

M. Abbaszadeh, H.J. Marquez, Nonlinear observer design for one-sided Lipschitz systems, in *Proceedings of the 2010 American Control Conference* (IEEE, 2010), pp. 5284–5289

M. Abbaszadeh, H.J. Marquez, Robust state observation for sampled-data nonlinear systems with exact and euler approximate models, in *American Control Conference* (IEEE, 2007), pp. 1687–1692

M. Abbaszadeh, L. K. Mestha, W. Yan, Forecasting and early warning for adversarial targeting in industrial control systems, in *2018 IEEE Conference on Decision and Control (CDC)* (IEEE, 2018), pp. 7200–7205

C. Allegorico, V. Mantini, A data-driven approach for on-line gas turbine combustion monitoring using classification models, in *Second European Conference of the Prognostics and Health Management Society* (2014), pp. 92–100

J.M. Bland, D.G. Altman, Transformations, means, and confidence intervals. BMJ: British Med. J. **312**(7038), 1079 (1996)

V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey. ACM Comput. Surv. (CSUR) **41**(3), 15 (2009)

H. Chen, N. Paoletti, S.A. Smolka, S. Lin, Committed moving horizon estimation for meal detection and estimation in type 1 diabetes, in *American Control Conference (ACC)* (IEEE, 2019), pp. 4765–4772

L. Clifton, D.A. Clifton, Y. Zhang, P. Watkinson, L. Tarassenko, H. Yin, Probabilistic novelty detection with support vector machines. IEEE Trans. Reliab. **63**(2), 455–467 (2014)

P. Cortez, M. Rio, M. Rocha, P. Sousa, Multi-scale internet traffic forecasting using neural networks and time series methods. Expert Syst. **29**(2), 143–155 (2012)

A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B (methodological) 1–38 (1977)

W. Ding, J. Wang, C. Rizos, D. Kinlyside, Improving adaptive Kalman estimation in GPS/INS integration. J. Navig. **60**(3), 517–529 (2007)

J. Ehlers, A. van Hoorn, J. Waller, W. Hasselbring, Self-adaptive software system monitoring for performance anomaly localization, in *Proceedings of the 8th ACM International Conference on Autonomic Computing* (ACM, 2011), pp. 197–200

M.R. Endsley, Toward a theory of situation awareness in dynamic systems. Hum. Factors **37**(1), 32–64 (1995)

T. Gneiting, A.E. Raftery, Weather forecasting with ensemble methods. Science **310**(5746), 248–249 (2005)

S. Gupta, A. Ray, S. Sarkar, M. Yasar, Fault detection and isolation in aircraft gas turbine engines. Part 1: underlying concept. Proc. Inst. Mech. Eng. Part G: J. Aeros. Eng. **222**(3), 307–318 (2008)

D.I. Harvey, S.J. Leybourne, Testing for time series linearity. Econometr. J. **10**(1), 149–165 (2007)

W. Huang, N. Li, Z. Lin, G.-B. Huang, W. Zong, J. Zhou, Y. Duan, Liver tumor detection and segmentation using kernel-based extreme learning machine, in *35th annual international conference of the IEEE Engineering in medicine and biology society (EMBC)* (IEEE, 2013), pp. 3662–3665

G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification. IEEE Trans. Syst. Man Cybern. Part B (Cybernetics) **42**(2), 513–529 (2012)

G.-B. Huang, An insight into extreme learning machines: random neurons, random features and kernels. Cogn. Comput. **6**(3), 376–390 (2014)

G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications. Neurocomputing **70**(1), 489–501 (2006)

S. Johansen, *Likelihood-Based Inference in Cointegrated Vector Autoregressive Models* (Oxford University Press, 1995)

S.K. Khaitan, J.D. McCalley, Design techniques and applications of cyberphysical systems: a survey. IEEE Syst. J. **9**(2), 350–365 (2014)

T. Kokkonen, J. Hautamäki, J. Siltanen, T. Hämäläinen, Model for sharing the information of cyber security situation awareness between organizations, in *2016 23rd International Conference on Telecommunications (ICT)* (IEEE, 2016), pp. 1–5

G. Koop, D. Korobilis, Forecasting inflation using dynamic model averaging. Int. Econ. Rev. **53**(3), 867–886 (2012)

R. Lamedica, A. Prudenzi, M. Sforna, M. Caciotta, V.O. Cencellli, A neural network based technique for short-term forecasting of anomalous load periods. IEEE Trans. Power Syst. **11**(4), 1749–1756 (1996)

T.H. McCormick, A.E. Raftery, D. Madigan, R.S. Burd, Dynamic logistic regression and dynamic model averaging for binary classification. Biometrics **68**(1), 23–30 (2012)

L.K. Mestha, O.M. Anubi, M. Abbaszadeh, Cyber-attack detection and accommodation algorithm for energy delivery systems, in *IEEE Conference on Control Technology and Applications (CCTA)* (2017), pp. 1326–1331

S. Nateghi, Y. Shtessel, J.-P. Barbot, C. Edwards, Cyber attack reconstruction of nonlinear systems via higher-order sliding-mode observer and sparse recovery algorithm, in *2018 IEEE Conference on Decision and Control (CDC)* (IEEE, 2018), pp. 5963–5968

S. Nateghi, Y. Shtessel, J.-P. Barbot, G. Zheng, L. Yu, Cyber-attack reconstruction via sliding mode differentiation and sparse recovery algorithm: Electrical power networks application, in *15th International Workshop on Variable Structure Systems (VSS)* (IEEE, 2018), pp. 285–290

G. Park, C. Lee, H. Shim, Y. Eun, K.H. Johansson, Stealthy adversaries against uncertain cyber-physical systems: threat of robust zero-dynamics attack. IEEE Trans. Autom. Control **64**(12), 4907–4919 (2019)

M.A. Pimentel, D.A. Clifton, L. Clifton, L. Tarassenko, A review of novelty detection. Signal Process. **99**, 215–249 (2014)

A.E. Raftery, M. Kárnỳ, P. Ettler, Online prediction under model uncertainty via dynamic model averaging: application to a cold rolling mill. Technometrics **52**(1), 52–66 (2010)

G. Rigatos, N. Zervos, P. Siano, P. Wira, M. Abbaszadeh, Flatness-based control for steam-turbine power generation units using a disturbance observer. IET Electr. Power Appl. (2021)

S.C. Rutan, Adaptive Kalman filtering. Anal. Chem. **63**(22), 1103A-1109A (1991)

F. Skopik, M. Wurzenberger, G. Settanni, R. Fiedler, Establishing national cyber situational awareness through incident information clustering, in *2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)* (IEEE, 2015), pp. 1–8

S. Sridhar, M. Govindarasu, Model-based attack detection and mitigation for automatic generation control. IEEE Trans. Smart Grid **5**(2), 580–591 (2014)

Y.-Y. Wang, H. Zhang, C.-H. Qiu, S.-R. Xia, A novel feature selection method based on extreme learning machine and fractional-order darwinian PSO. Comput. Intell. Neurosci. **2018**, Article ID 5078268, 8 pages (2018)

F. Xue, W. Yan, Parametric model-based anomaly detection for locomotive subsystems, in *International Joint Conference on Neural Networks, IJCNN 2007* (IEEE, 2007), pp. 3074–3079

W. Yan, L. Yu, On accurate and reliable anomaly detection for gas turbine combustors: a deep learning approach, in *Proceedings of the Annual Conference of the Prognostics and Health Management Society* (2015)

W. Yan, L.K. Mestha, M. Abbaszadeh, Attack detection for securing cyber physical systems. IEEE Int. Things J. **6**(5), 8471–8481 (2019)

Y. Yin, Y. Zhao, B. Zhang, C. Li, S. Guo, Enhancing elm by markov boundary based feature selection. Neurocomputing **261**, 57–69 (2017). Advances in Extreme Learning Machines (ELM 2015)

A. Zaher, S. McArthur, D. Infield, Y. Patel, Online wind turbine fault detection through automated scada data analysis. Wind Energy **12**(6), 574–593 (2009)

A. Zimek, E. Schubert, H.-P. Kriegel, A survey on unsupervised outlier detection in high-dimensional numerical data. Stat. Anal. Data Mining: ASA Data Sci. J. **5**(5), 363–387 (2012)