Fault Detection and Identification using Bayesian Recurrent Neural Networks

- even small faults in a complex industrial
  system can initiate a series of events that result in loss of efficiency
  and reliability

- The proposed approach tackles three key challenges typical of manufacturing systems:
- (1) nonlinearity,
- (2) nonGaussian distributed variables
- (3) high-degree of spatio-temporal correlations

RNNs

- RNN comprises an input layer, one or more hidden recurrent layers, and an output layer
- hidden recurrent layers capture the state with the response of its nodes being added to the inputs on the next time step.
- The state variable **s***t* is calculated based on the previous state **s***t*-1 and the current input **x***t*. The RNN output **y***t* is then calculated based on the current state
- the state and output layers have the general form
- training RNNs involves backpropagation through time (BPTT) to compute the gradient update of the model weights that minimizes the loss function

$$\mathbf{s}_t = f_s\left(\mathbf{x}_t, \mathbf{s}_{t-1} \middle| \theta_s\right)$$

$$\hat{\mathbf{y}}_t = \mathbf{s}_t \mathbf{W}_y + \mathbf{b}_y$$

Bayesian Recurrent Neural Networks

- instead of point estimates, BRNNs can effectively perform Bayesian inference which provides probabilistic distributions over the outputs .
- BRNNs view the model parameters $w = \{\mathbf{W}s;\mathbf{W}y;\mathbf{U}s;\mathbf{b}s;\mathbf{b}y\}$ as random variables from a prior distribution $p\ (w)$
- given a training dataset comprising **X** and **Y**, learning entails estimating the posterior distribution $p(w/\mathbf{X};\mathbf{Y})$
- distribution of a predicted output $\mathbf{y}*$

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}^*|\mathbf{x}^*, \omega)p(\omega|\mathbf{X}, \mathbf{Y})d\omega$$

Variational dropout

- exact inference of the posterior is not possible
- Variational inference is a technique used to approximate an intractable posterior distribution with a simpler parameterized distribution $q(w)$
- the integration above can be approximated simply by Monte Carlo (MC) integration using $q(w)$

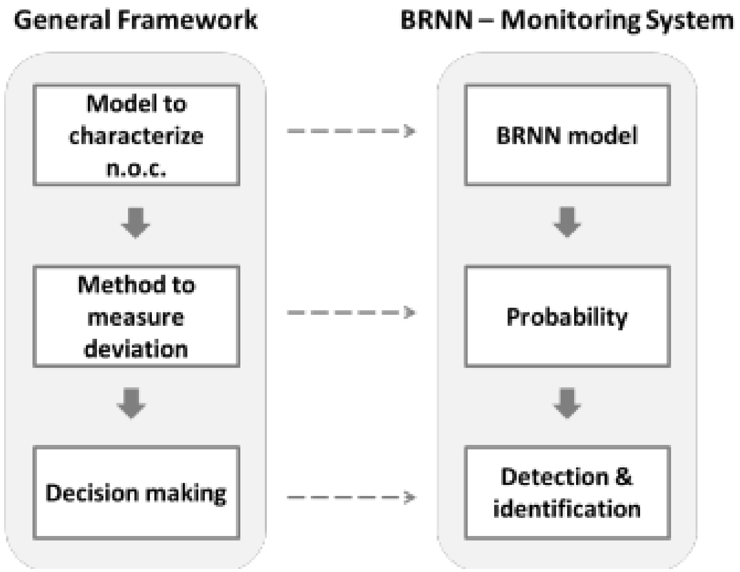
Variational dropout

- During both training and testing, the variational approximation involves sampling the model distribution with regard to the variational distribution over the weights, which is implemented by dropping out (i.e., forcing to zero) randomly chosen inputs, outputs, and hidden states
- the dropout mask used for each model realization be kept fixed between time steps

Methodology

- Historical data collected during NOC are used to build the model,
- one must also establish an approach to characterize the magnitude of the deviation from NOC based on the developed model
- given a new observation $\mathbf{x}*$, these are calculated to determine whether $\mathbf{x}*$ deviates substantially from the NOC

- As model they use BRNN with variational dropout



Fault detection

- The first step toward fault detection is to learn a model to characterizing the NOC
- this involves training to model the dynamics in time and correlations between sensors *and* the prediction uncertainty resulting from model mismatch and inherent system variability/noise.
- The BRNN model is trained directly on historical NOC data

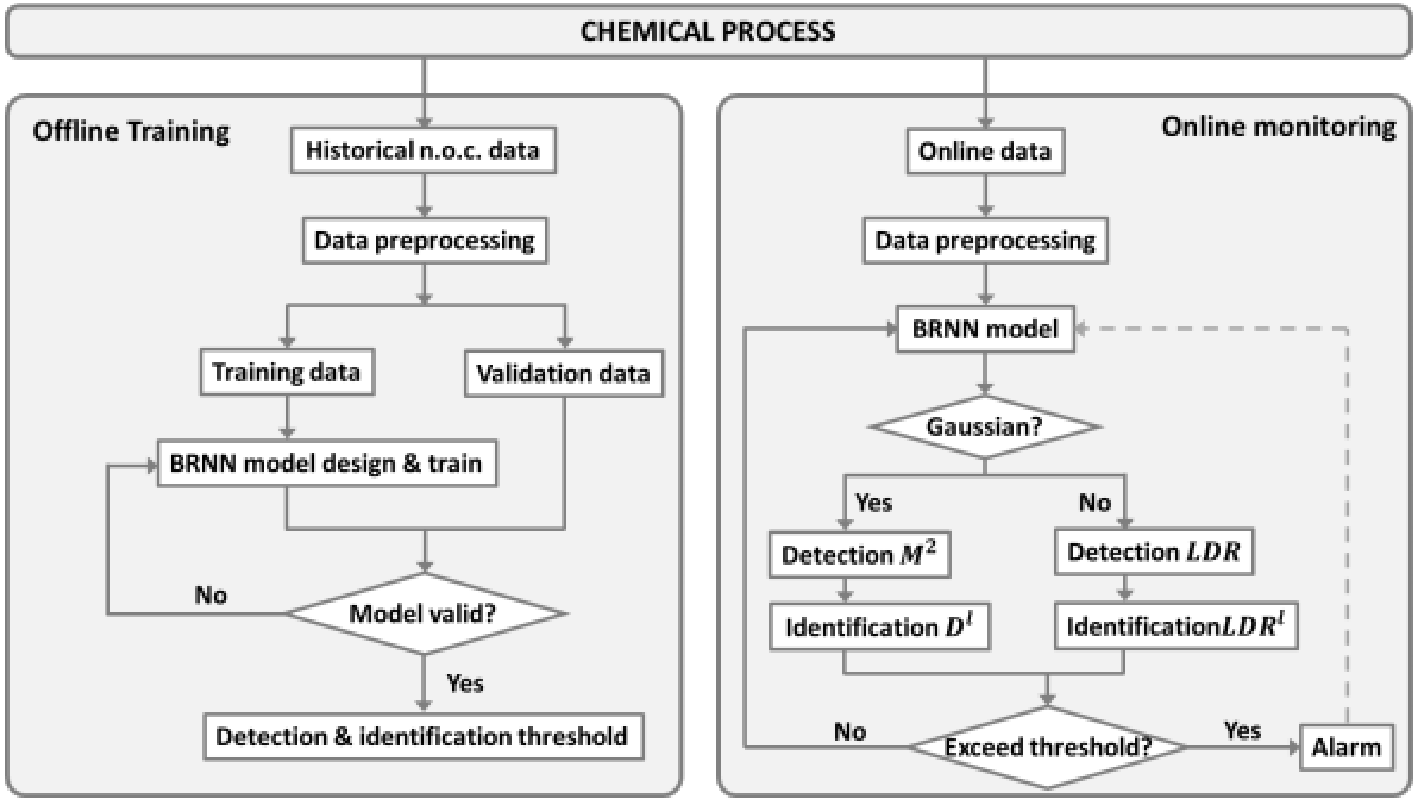
Fault detection

- After training, the model output **x***t*+1 from the BRNN is sampled from the posterior predictive distribution for next observation **x***t*+1 via variational dropout model realizations.
- stochastic forward pass is repeated *N* times, each with a different dropout mask, and the predictive distribution of the output for $t + 1$ is approximated based on the MC samples of the BRNN model
- Then, when the true observation **x***t*+1 is available, it is compared to the predictive distribution
- the true observation is fed into the BRNN model and the procedure is repeated for the next time step

Fault identification

- BRNN fault identification is obtained by applying the fault detection approach but independently for each variable.
- To determine which variables deviate abnormally, each observation variable is compared to its corresponding predicted *marginal* posterior distribution estimated from the BRNN samples.
- This variable-wise comparison allows us to identify variables with values in low probability areas and thus more likely to be relevant for diagnosing the fault.
- Marginal distribution for each variable also evolve over time with dynamics that depend on other variables and past observations

Scheme


CHEMICAL PROCESS
Offline Training
Historical n.o.c. data
Data preprocessing
Training data
Validation data
BRNN model design & train
No
Model valid?
Yes
Detection & identification threshold
Online monitoring
Online data
Data preprocessing
BRNN model
Gaussian?
Yes
No
Detection M^2
Detection LDR
Identification D^I
Identification LDR^I
No
Exceed threshold?
Yes
Alarm