Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Assessing Observability using Supervised Autoencoders with Application to Tennessee Eastman Process ⋆

Piyush Agarwal * **Melih Tamer** ** **Hector Budman** ***

* *University of Waterloo, Waterloo, ON, N2L3G1 Canada (e-mail: piyush.agarwal@uwaterloo.ca).*
** *Sanofi Pasteur, Toronto, ON, Canada (email: melih.tamer@sanofi.com)*
*** *University of Waterloo, Waterloo, ON, N2L3G1 Canada (e-mail: hbudman@uwaterloo.ca)*

Abstract: This work presents a novel approach to calculate classification observability using a supervised autoencoder (SAE) neural network (NN) for classification. This metric is based on a minimal distance between every two classes in the latent space defined by the hidden layers of the auto-encoder. Quantification of classification observability is required to address whether the available sensors in a process are sufficient to observe certain outputs (phenomenon) and which additional measurements are to be included in the dataset to improve classification accuracy. The efficacy of the proposed method is illustrated through case-studies for the Tennessee Eastman Benchmark Process.

Keywords: Autoencoder, semi-supervised learning, observability, classification, Tennessee Eastman Process, deep learning

1. INTRODUCTION

Identification of inputs that are highly informative and well correlated to an output of interest, e.g. productivity of a process, is crucial for efficient chemical process plant operation, manufacturing flexibility, improved system knowledge and operational robustness with respect to unknown disturbances. An input design space is defined as "the multidimensional combination and interaction of input variables and process parameters" Laky et al. (2019), that assures quality of product within specified operational constraints. Classification of the input variables' space into distinct regions that result in correspondingly distinct output classes is often challenging due to the proximity among input values that correspond to different classes combined with the presence of measurement noise. We focus on classification of different regions of an economic profit function for a chemical process with respect to process inputs using a Supervised Autoencoder Neural Networks (SAE-NNs). SAE-NNs are Autoencoders (AE) that predicts both reconstructed inputs as well as outputs. Previously, SAE-NNs or its variants have been used for image classification and other regression tasks in a semi-supervised setting i.e. making use of both labelled and unlabelled data (Epstein and Meir (2019); Seeger (2001)). To accomplish this task the following objective function is minimized with respect to the weights of the SAE-NN:

$$l_{SAE} = \sum_{s=1}^{N} L_r^s(\mathbf{y}_s, \hat{\mathbf{y}_s}) + \lambda_1 \sum_{i=1}^{N} L_p^s(\mathbf{x}_s, \hat{\mathbf{x}_s}) \quad (1)$$

The addition of the input reconstruction loss L_r^s (first term in Equation (1)) to the supervised learning related term L_p^s for a sample s (second term in Equation (1)) in the objective function has been a subject of debate as to why the input reconstruction helps in better classification Rigollet (2007). The focus of this work is to quantify a robust lower bound on classification observability (C_{obs}) of output classes from inputs using SAE-NNs models. The ability of classifying regions of the input space that result in corresponding classes of a process output, e.g. process productivity, depends on the degree of observability of the output from the measured process inputs. Quantifying observability of classification task can help answering several important industrial questions such as: are the available sensors sufficient to provide acceptable classification accuracy? which sensors are more informative for the classification task? It should be noticed that observability cannot be assessed by standard state observability methods since a state model is assumed to be unavailable. Hence, the novelty of the current proposed method is in assessing observability directly from input-output data that to our knowledge has not been thoroughly researched in the literature.

The development of an SAE-NN model to be used for classification involves several steps: i- feeding the inputs to the encoder, ii- feeding the outputs from the encoder to a fully connected layer and iii- feeding the outputs from the fully connected layer a to classification layer consisting of softmax functions. iv- jointly training the autoencoder

⋆ This research is supported by Mitacs through Mitacs-Accelerate Program

and classifier. Due to the data projection (compression) operation achieved by the encoder, the outputs from the encoder are referred to as latent variables. The difficulty in observing the output classes from input data is due to the proximity/overlap among sets of input data, model structure error and noise. Regarding overlap, the support of the encoder functions corresponding to different output classes define regions in the latent variable space (output space of the encoder) that may strongly overlap with each other. This overlap may cause miss-classification of new samples, i.e. samples that were not used for training. In this study we perform numerical evaluation of the overlap between regions in the latent space that correspond to different classes and observability of the classes is quantified from the degree of overlap. The overlap is estimated for any two input data points $\mathbf{x}_i$ and $\mathbf{x}_j \in \mathbb{R}^{d_x}$ based on a distance d_{ij} between their projections in the latent variable space $\mathbf{z} \in \mathbb{R}^{d_z}$ where points $\mathbf{x}_i$ and $\mathbf{x}_j$ corresponds to different classes ($d_x > d_z$). If these distances are large enough as compared to certain threshold d_{ij} (robust observability distance measure) related to the noise in the input measurements, the classes are considered to be observable while if the distance is smaller than the threshold the system is considered unobservable as follows:

$$d = ||\mathbf{z}_i - \mathbf{z}_j||_2^2 = \begin{cases} \text{observable}, & \text{if } d > d_{ij}. \\ \text{non-observable}, & \text{otherwise.} \end{cases} \quad (2)$$

where points $\mathbf{z}_i$ and $\mathbf{z}_j$ are projections of $\mathbf{x}_i$ and $\mathbf{x}_j$ in the latent space that corresponds to different output classes.

Quantifying the observability in the latent variable space capitalizes on the lower dimensions of this space as compared to the original input space thus drastically simplifying the calculation.

It is also investigated that beyond its use for assessing classification observability, the degree of classification observability C_{obs} can be further enhanced by discarding inputs that are not informative for classification (Agarwal and Budman (2019); Agarwal et al. (2020)) and contribute to overlap between regions corresponding to different output classes and is not presented for brevity. The discarded inputs do not contribute to the classification task and instead they decrease the classification accuracy due to reducing distinction between classes (creating confusion) and overfitting. Eliminating sensors that do not contribute significantly to classification may help to reduce cost and to reduce miss-classification resulting from potentially faulty sensors/ irrelevant sensors.

Following the above, the three main contributions of this work: i) Assessment of the use of the reconstruction error for training the classification model; ii) Derivation of a robust observability distance measure (RODM) to evaluate the degree of classification observability C_{obs}; iii) Identification of input variables contributing to the overlap. The proposed contributions are illustrated through case-studies of the Tennessee Eastman Benchmark Process.

The paper is organized as follows: Section 2 provides a brief review on Autoencoder NNs. Section 3 provides the problem description and the TEP case-study. The two algorithms used for quantifying a robust lower bound on observability are presented in Section 4. Results and discussions on the case-study are shown in Section 5 followed by concluding remarks in Section 6.

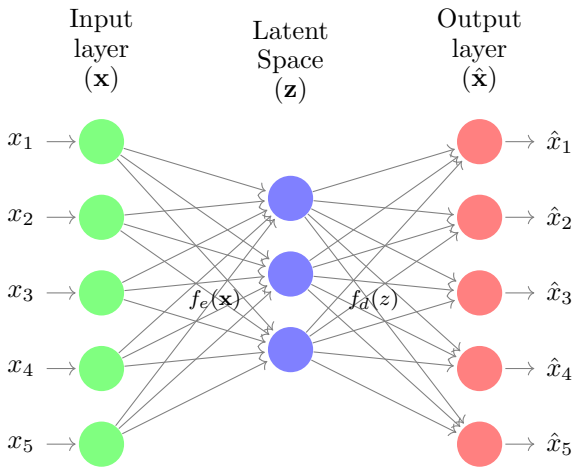

Fig. 1. Traditional single layer Autoencoder Neural Network (AE-NN)

2. PRELIMINARIES

This section briefly reviews the fundamentals of an Autoencoder (AE-NNs) and a Supervised Autoencoder Neural Networks (SAE-NNs) models.

2.1 Autoencoder Neural Networks (AE-NNs)

A traditional AE-NN is a neural network model composed of two parts, encoder and decoder, as shown in Figure 1. An AE is trained in an unsupervised fashion to extract underlying patterns in the data and to facilitate non-linear dimensionality reduction. The encoder is trained so as to compress the input data onto a reduced latent space and the decoder uncompresses back the hidden layer outputs into the reconstructed inputs. Let consider the input to an AE is a vector $\mathbf{x} \in \mathbb{R}^{d_x}$, then the operation performed by the encoder for a single hidden layer between the input variables to the latent space $\mathbf{z} \in \mathbb{R}^{d_z}$ variables (latent variables) can be represented as follows:

$$\mathbf{z} = f_e(\mathbf{W}_e\mathbf{x} + \mathbf{b}_e) \quad (3)$$

where f_e is a chosen non-linear activation function for the encoder, $\mathbf{W}_e \in \mathbb{R}^{d_z \times d_x}$ is an encoder weight matrix and $\mathbf{b}_e \in \mathbb{R}^{d_z}$ is a bias vector. The decoder reconstructs back the inputs from the feature or latent space $\mathbf{z} \in \mathbb{R}^{d_z}$ as per the following operation follows:

$$\hat{\mathbf{x}} = f_d(\mathbf{W}_d\mathbf{z} + \mathbf{b}_d) \quad (4)$$

where f_d is a chosen activation function for the decoder, $\mathbf{W}_d \in \mathbb{R}^{d_x \times d_z}$ and $\mathbf{b}_d \in \mathbb{R}^{d_x}$ is a decoder weight matrix and a bias vector respectively. The 'tanh' function is used for both transforming the inputs into the latent variables and for reconstructing the inputs from the latent variables in this work. The AE-NN is trained based on the following minimization problem:

$$l_{SE}(\mathbf{x}, \mathbf{W}_d\mathbf{W}_e\mathbf{x}) = \frac{1}{2N}||\mathbf{x} - \hat{\mathbf{x}}||_2^2 = \frac{1}{2N}\sum_{s=1}^{N}||\mathbf{x}_s - \hat{\mathbf{x}}_s||_2^2 \quad (5)$$

where N is the number of samples.

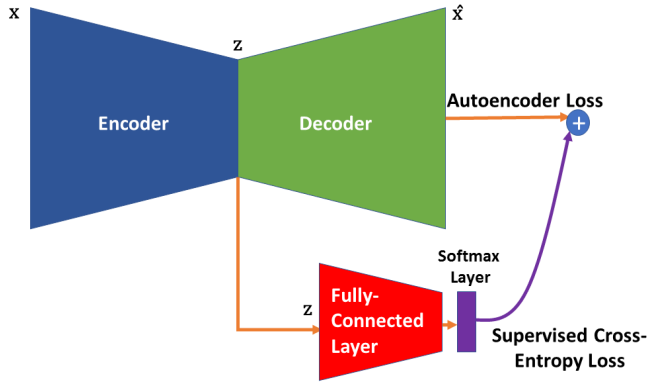

Fig. 2. Schematic of a single layer Supervised Autoencoder Neural Network (SAE-NN)

2.2 Supervised Autoencoder Classification Neural Networks (SAE-NNs)

The Supervised Autoencoder Neural Network (SAE-NN) model, shown in Figure 2, is trained based on the minimization of a combination of the reconstruction loss function and the supervised classification loss corresponding to the first and second terms in (Equation (9)) respectively. The reconstruction loss function in Equation (1 and 9) is ensuring that the calculated latent variables are able to reconstruct the input data with good accuracy. The goal is to learn a function that predicts the class labels in one-hot encoded form $\mathbf{y} \in \mathbb{R}^m$ from inputs $\mathbf{x} \in \mathbb{R}^{d_x}$. The encoder operation for a single hidden layer between the input variables to the latent variables $\mathbf{z} \in \mathbb{R}^{d_z}$ is represented as follows:

$$\mathbf{z} = f_e(\mathbf{W}_e\mathbf{x} + \mathbf{b}_e) \tag{6}$$

The latent variables are used both to predict the class labels and reconstruct inputs $\mathbf{x}$ as follows:

$$\hat{\mathbf{x}} = f_d(\mathbf{W}_d\mathbf{z} + \mathbf{b}_d) \tag{7}$$
$$\hat{\mathbf{y}} = f_c(\mathbf{W}_c\mathbf{z} + \mathbf{b}_c) \tag{8}$$

where f_c is a non-linear activation function for the output layer. $\mathbf{W}_c \in \mathbb{R}^{m\times d_z}$ and $\mathbf{b}_c \in \mathbb{R}^m$ are output weight matrix and bias vector respectively. For training the SAE, the following loss function is minimized:

$$\begin{aligned} l_{SAE} &= \lambda_1 \sum_{s=1}^{N} L_r^s(\mathbf{x}_s, \mathbf{W}_d\mathbf{W}_e\mathbf{x}_s) + \sum_{s=1}^{N} L_p^s(\mathbf{W}_c\mathbf{W}_e\mathbf{x}_s, \mathbf{y}_s) \\ &= \frac{\lambda_1}{N}\sum_{s=1}^{N} ||\mathbf{x}_s - \hat{\mathbf{x}}_s||_2^2 + \frac{1}{N}\sum_{s=1}^{N}\sum_{c=1}^{m} -y_{s,c}log(p_{s,c}) \\ &= \frac{1}{N}\left[\lambda_1 \sum_{i=1}^{N} ||\mathbf{x}_s - \hat{\mathbf{x}}_s||_2^2 + \sum_{s=1}^{N}\sum_{c=1}^{m} -y_{s,c}log(p_{s,c})\right] \end{aligned} \tag{9}$$

$$p_{s,c} = \frac{e^{(\hat{y_{s,c}})}}{\sum_{c=1}^{m} e^{(\hat{y_{s,c}})}} \tag{10}$$

where λ_1 is the weight for the reconstruction loss L_r, m is the number of classes, $y_{s,c}$ is a binary indicator (0 or 1), 1 if class label c is the correct classification for observation s, $\hat{y_{s,c}}$ is the non-normalized log probabilities and $p_{s,c}$ is the predicted probability for a sample s of class c.

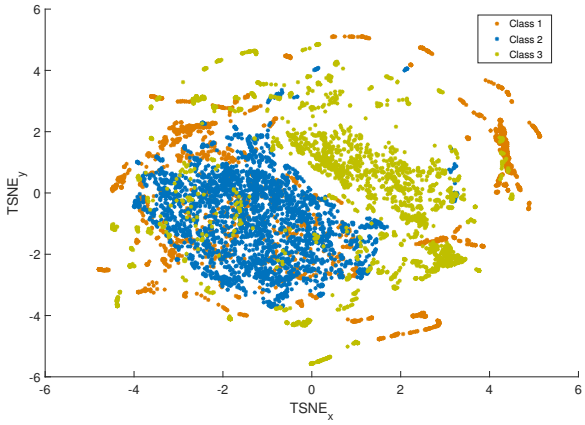

Fig. 3. Case 2: Projection of input space in 2 dimensions using TSNE for overlapping case

3. PROBLEM DESCRIPTION

The TEP involves several unit operations including a vapor-liquid separator, a reactor, stripper, a recycle compressor and a condenser. Four gaseous reactants (A, B, C and D) form two liquid products streams (G and H) and a by-product (F). Although several TEP simulators are available, in this work the one developed by Larsson et al. (2001) was used. The original controller settings were modified and different disturbances, i.e. referred to as faults in the TEP simulator, were introduced in order to generate different ranges of values of process profit since the goal in the current study is to classify the inputs according to their resulting process profit. The simulator involves 53 input variables of which 3 manipulated variables (Compressor Recycle Valve (XMV(5)), Stripper Steam Valve (XMV(9)) and Agitator Speed XMV(12)) were discarded initially (number of input variables = 50). Since the process profit for this case study is determined solely by the operating costs of the plant, this profit will be referred to as cost of productivity (COP).

COP ($/hr)	High Profit	Intermediate	Low Profit
Case 1	> 89.6	$89.6 - 142.6$	< 142.6
Case 2	> 108	$108 - 130$	< 130

Table 1. Profit-based defined classes for COP

Also, since the boundaries between classes corresponding to different ranges of COP values can be chosen arbitrarily, we examine two different cases that are defined in Table 1. These cases differ in the overlap between classes. This overlap is calculated from the training data based on simulated frequency of occurrences of COP values as shown in Figure 4. As shown in this figure Case 1 corresponds to very low overlap while case 2 results in significant overlap between classes. The overlap is illustrated by TSNE (t-distributed Stochastic Neighbor Embedding, Maaten and Hinton (2008)) projections of the input design space for the high overlap in Figure 3.

A total of 8 datasets were generated, 1 normal operation and 7 each involving one known fault (IDV(1)-IDV(7)) for a total simulation time of 800 hours, i.e. a 100-

hour duration for each dataset. Each fault was activated at the start of the corresponding 800-hour time period and data samples were collected at a sampling rate of 100 samples/hour (total number of samples 8×10^4 per dataset). Out of which 3×10^4 samples were considered as training dataset and 1.5×10^4 samples as validation dataset and testing dataset. Each of these datasets resulted in various ranges of COP values, i.e. different classes (refer Figure 4 and Table 1) to be identified by the SAE-NN model.

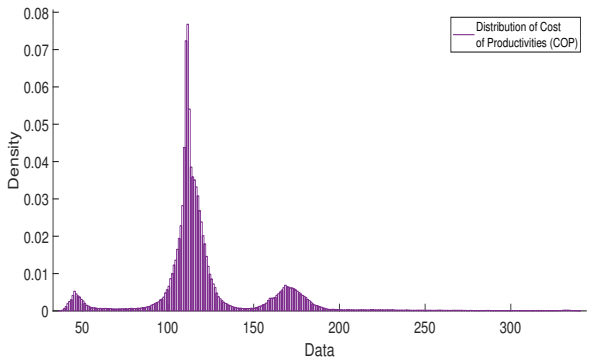

Fig. 4. Distribution of Cost Of Productivities (COP)

4. PROPOSED METHODOLOGY

The goal is to find an observability measure which is robust to measurement noise in the input data. The proposed algorithm is based on calculating a distance measure in the latent space using SAE with respect to noisy inputs using a boot-strapping approach. The two algorithms are as follows:

(1) Algorithm 1 (Robust Observability Distance Measure (RODM)) for the computation of the RODM ($d_{ij}, i \neq j$).
(2) Algorithm 2 (Evaluation of degree of observability C_{obs}) for the computation of a degree of observability C_{obs} which is defined as the percentage of overlap between points within a neighbourhood of distance ($d_{ij}, i \neq j$) from each point, where i and j are points in different class labels.

To compute the RODM d_{ij}, first a SAE-NN model is trained on the training data collected from the process. Subsequently a bootstrapping approach is used where the inputs are perturbed around different operating regions for R number of realizations of white-noise and the variances in the latent variables resulting from these input perturbations are evaluated (refer Equations (11) and (12). Finally, a distance measure is defined as the maximum of l_2 norm of the variance of perturbations in latent variables due to noisy inputs (refer Equation and (13)) across R realizations (see Algorithm 1).

To calculate the degree of classification observability C_{obs} (see Algorithm 2), we first evaluate pairwise Euclidean distance matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ where N is the number of samples in the validation dataset. Afterwards, inter-class samples that are closer than RODM (d_{ij}) are selected (referred as Total percentage overlap (%TOv) in the pseudo code Algorithm 2). Thereafter, the points which

Algorithm 1 Robust Observability Distance Measure (RODM)

1: Train an SAE classification NN $g(\mathbf{W}_c \mathbf{W}_e \mathbf{x})$ using an optimal weighting of the reconstruction and classification loss-functions, where this weighting is found by using a validation dataset.
2: Perturb the input variables, $\mathbf{x}_l$ ($l = 1, 2, \ldots, d_x$) (mean $\mu_{\mathbf{x}_l} = 0$; variance $\sigma^2_{\mathbf{x}_l}$), with input perturbations $\Delta\mathbf{x}_l$. Where $\Delta\mathbf{x}_l$ ($l = 1, 2, \ldots, d_x$) are independent normally distributed (i.i.d) random variables that has mean $\mu_{\Delta\mathbf{x}_l} = 0$ and variance $\sigma^2_{\Delta\mathbf{x}_l}$ for R uncorrelated realizations such that Signal-to-noise ratio (SNR) $\frac{\sigma^2_{x\,l}}{\sigma^2_{\Delta x_l}} = 10$ is maintained.
3: Compute the latent feature vectors $\mathbf{z}_k$ ($k = 1, 2, \ldots, d_z$) for R realizations of $\mathbf{x}_l + \Delta\mathbf{x}_l$ using the trained SAE model.
4: Estimate the variances of the latent variables resulting from the introduced perturbations to the inputs (noise) for R realizations in the latent space as follows:

$$V(\Delta\hat{\mathbf{z}_k}) = \mathbb{E}\Big[\Big(\Delta\mathbf{z}_k - \mathbb{E}\Big(\Delta\mathbf{z}_k\Big)\Big)\Big(\Delta\mathbf{z}_k - \mathbb{E}\Big(\Delta\mathbf{z}_k\Big)\Big)^T\Big] \tag{11}$$

$$\Delta\mathbf{z}_k = \Big(g(\mathbf{W}_e(\mathbf{x}_i + \Delta\mathbf{x}_i)) - g(\mathbf{W}_e\mathbf{x}_i)\Big) \tag{12}$$

where $k = 1, 2, \ldots, d_z$ & $\mathbb{E}$ is an expectation operator.
5: Robust Observability Distance Measure (RODM) is computed as the maximum of l_2 norm of the estimated variance of $\Delta\mathbf{z}$ for R realizations as:

$$d_{ij} = max\Big\{\sqrt{V(\hat{\Delta}\mathbf{z}_1) + V(\hat{\Delta}\mathbf{z}_2) + \ldots + V(\hat{\Delta}\mathbf{z}_k)}\Big\}_R \tag{13}$$

where $i \neq j$.

are correctly classified are discarded and the remaining samples are used to evaluate the Total Classification Percentage Overlap (%TCOv) as:

$$\%\text{TCOv} = \frac{\sum_{n(u)\neq l(u)} \text{unique(cov).length}}{ind.length}$$

Finally the degree of observability C_{obs} is calculated as

$$C_{obs} = \%\text{Training Accuracy} - \%\text{TCOv}$$

The obtained C_{obs} is evaluated for the worst-case possible using RODM and represents the lower bound on the degree of observability i.e. new samples are expected to exhibit equal or larger classification accuracy.

The evaluation of RODM (see Algorithm 1) not only helps in the evaluation of degree of classification observability C_{obs} but can also be used to define an extra term in the objective function with explicitly determining the inter-class distance as RODM in the latent space for achieving higher testing accuracy.

5. RESULTS AND DISCUSSION

The following section presents the results of the application of the algorithms of the previous section to the TEP case-study. The advantage of adding the reconstruction loss function L_r in Equation (9) is also assessed.

Table 2. Degree of classification observability (C_{obs}) for Case 1 and Case 2

	cov_{12}	cov_{21}	cov_{13}	cov_{31}	cov_{23}	cov_{32}	$d_{ij}(i \neq j)$	% TCOv	C_{obs}
Case 1	52	14	0	303	0	0	1.14	2.46%	95.28%
Case 2	4565	8	48	1931	0	1070	0.5168	43.84%	53.78%
Enhanced Classification Observability									
Case 1	54	1	10	240	0	0	1.5818	2.02%	96.16%
Case 2	4542	7	74	774	0	969	0.8654	42.40%	54.47%

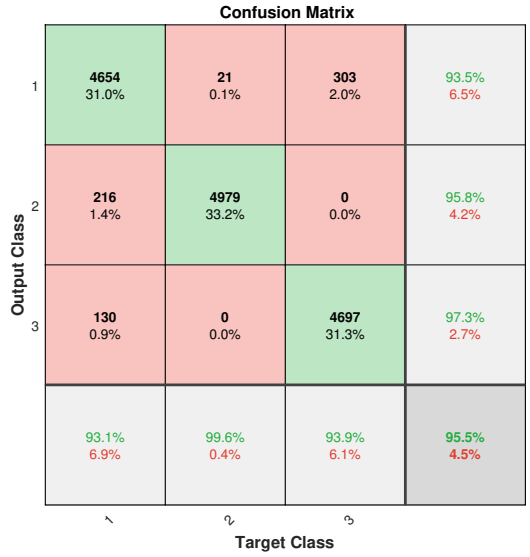

Fig. 5. Confusion Matrix for Case 1 (Validation Data-set)

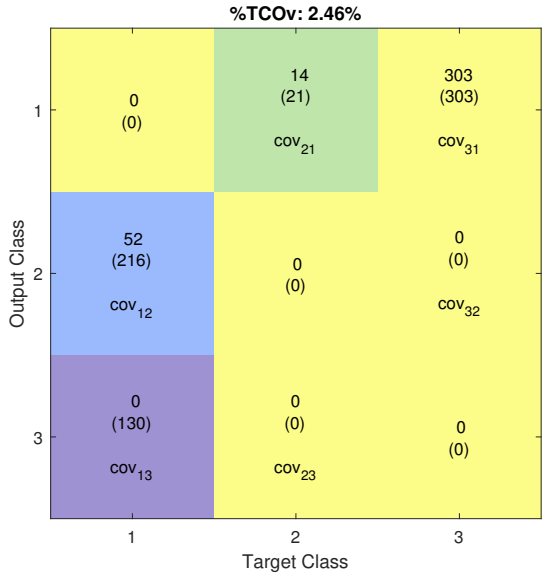

Fig. 6. Classification Overlap Matrix (COv) for Case 1 (Validation Data-set)

5.1 Effect of Reconstruction Error Loss function on classification accuracy

The first objective of the case study is to investigate whether the addition of the reconstruction loss L_r term to the supervised loss function L_p (see Equation (9)) for training the classification AE-NN model helps to improve classification. An SAE-NN with a single layer was trained with different weights λ_1 with a validation dataset. The hyper-parameters including the weight multiplying the reconstruction loss L_r term in Equation (9) and the dimension of the latent space $\mathbf{z}$ were chosen based on the highest classification accuracy achieved on the validation set for both cases 1 and 2. It can be observed in Table 3 that for different dimensions of the latent space $\mathbf{z}$, the validation classification accuracy and test classification accuracy with reconstruction loss function L_r was always higher than the NN architecture without the reconstruction loss function L_r.

5.2 Degree of Classification Observability (C_{obs}) for the TEP problem

The degree of classification observability C_{obs} is calculated according to Algorithms 1 and 2, presented in Section 4. First the RODM is calculated for both the cases i.e. Case 1 and Case 2 using $R = 1000$ realizations of input perturbations. It can be observed that RODM is larger i.e $d_{ij} = 1.14$ for case 1 as compared to case 2 i.e. $d_{ij} = 0.5168$ which indicates that case 1 has higher degree of observability than case 2. The confusion matrix for Case 1 and the classification overlap (COv) matrix evaluated using both Algorithm 1 and 2 are shown in Figures 5 and 6 respectively. The computation of COv matrix explains the root cause for the miss-classification of samples. The numbers shown in coloured boxes of COv matrix represents the number of samples that are miss-classified because of the proximity between each two different regions. The numbers shown below (in the brackets) shows the total number of samples miss-classified. The degree of classification observability C_{obs} for Case 1 is:

$$C_{obs} = 97.74\% - 2.46\% = 95.28\%$$

It can be seen that the C_{obs} (lower-bound) is smaller than the validation data-set accuracy 95.5% (shown in the right corner of Figure 5) and 97.5% for the test data-set accuracy (see Table 2). The results for both Case 1 and Case 2 are summarized in Table 2. The results corroborate that Case 1 is easily separable than Case 2 i.e. the degree of classification observability for Case 1 is much higher than Case 2.

6. CONCLUSION

This paper presents a novel method to compute a robust observability distance measure (RODM) and evaluate degree of classification observability C_{obs} for a classification problem based on noisy input data. The proposed method first computes a distance metric such that two clusters

Algorithm 2 Evaluation of degree of observability C_{obs}

Evaluate pairwise Euclidean distance matrix $\mathbf{D} \in \mathbb{R}^{N\times N}$
Determine the indices (ind_c, where $c = 1,2,\ldots,m$) of samples corresponding to m different classes.
$ind = \{ind_1, ind_2, \ldots, ind_c\}$; $t = 0$
for i in number of classes (m) **do**
  for j in number of classes (m) **do**
    if $i! = j$ **then**
      $t = t + 1$
      for q in ind_i.length (samples of class i) **do**
        for r in j_1.length, where $j_1 = ind_j$ **do**
          tov_{ij} = list() & $\text{cov}_{n(u)l(u)}$ = list()
          if $\mathbf{D}(ind_i(q), j_i(r)) < d_{ij}$ **then**
            tov_{ij}.add = $j_i(r)$
          end if
        end for
      end for
    end if
    return $n(t) = i$
    return $l(t) = j$
  end for
end for
for i in $m(m-1)$ i.e. #overlapping regions ($i \neq j$) **do**
  tov_{ij} = unique(tov_{ij})
end for
tov = {tov_{12},tov_{13},...,tov_{1m},tov_{21},tov_{23},...,$\text{tov}_{m(m-1)}$}
Total Percentage Overlap (% TOv) is determined by:

$$\%\text{TOv} = \frac{\sum_{i\neq j} \text{unique(tov).length}}{ind.length} \tag{14}$$

Total Classification Percentage Overlap (% TCOv) is determined by taking SAE-NN output probabilities $p_{i,c}$, where $c = (1, 2, \ldots, m)$ into account.
for u in $m(m-1)$ i.e. #overlapping regions ($i \neq j$) **do**
  for v in the length of $\text{tov}_{n(u)l(u)}$ **do**
    if $p_{v,l(u)} > p_{v,l\setminus l(u)}$ **then**
      $\text{cov}_{n(u)l(u)}$.add = tov(u, v)
    end if
  end for
end for
cov = {cov_{12},cov_{13},...,cov_{1m},cov_{21},cov_{23},...,$\text{cov}_{m(m-1)}$}
Total Classification Percentage Overlap (% TCOv) is determined by:

$$\%\text{TCOv} = \frac{\sum_{n(u)\neq l(u)} \text{unique(cov).length}}{ind.length} \tag{15}$$

$$C_{obs} = 100\% - (\%\text{TCOv} + (100\% - \text{Training \%Accuracy}))$$

$$C_{obs} = \%\text{Training Accuracy} - \%\text{TCOv} \tag{16}$$

of points belonging to different classes should be at least distance d_{ij} ($i \neq j$) apart in the worst case-scenario where i and j are points corresponding to different labels in representation space for a good classification. The merit of the method is that it can be used to assess the observability of output classes from available input data that is corrupted by noise. Furthermore, it is shown that the observability and classification accuracy can be enhanced by discarding variables that are not relevant for the classification task and contribute to overlap between different regions corresponding to output classes. It is argued that the proposed observability can be used in the future for selecting sensors to increase the observability of the classes or for providing a threshold for relevances of inputs and neural network interconnections where this threshold will serve for pruning the network thus avoiding over-fitting of noisy data.

Table 3. Classification Accuracy for both cases ($\mathbf{z} \in \mathbb{R}^{d_z}$, $d_z = 7$)

	L_r Weights	Validation Accuracy	Training Accuracy	Test Accuracy
Case 1	0.5	95.53%	97.74%	97.5%
Case 1	0	94.81%	97.12%	-
Case 2	0.5	55.94%	97.62%	55.8%
Case 2	0	55.59%	92.05%	-

REFERENCES

Agarwal, P. and Budman, H. (2019). Classification of profit-based operating regions for the tennessee eastman process using deep learning methods. *IFAC-PapersOnLine*, 52(1), 556–561.

Agarwal, P., Tamer, M., Sahraei, M.H., and Budman, H. (2020). Deep learning for classification of profit-based operating regions in industrial processes. *Industrial & Engineering Chemistry Research*, 59(6), 2378–2395. doi: 10.1021/acs.iecr.9b04737. URL `https://doi.org/10.1021/acs.iecr.9b04737`.

Epstein, B. and Meir, R. (2019). Generalization bounds for unsupervised and semi-supervised learning with autoencoders. *arXiv preprint arXiv:1902.01449*.

Laky, D., Xu, S., Rodriguez, J.S., Vaidyaraman, S., García Muñoz, S., and Laird, C. (2019). An optimization-based framework to define the probabilistic design space of pharmaceutical processes with model uncertainty. *Processes*, 7(2), 96.

Larsson, T., Hestetun, K., Hovland, E., and Skogestad, S. (2001). Self-optimizing control of a large-scale plant: The tennessee eastman process. *Industrial & engineering chemistry research*, 40(22), 4889–4901.

Maaten, L.v.d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov), 2579–2605.

Rigollet, P. (2007). Generalization error bounds in semi-supervised classification under the cluster assumption. *Journal of Machine Learning Research*, 8(Jul), 1369–1392.

Seeger, M. (2001). Learning with labeled and unlabeled data (technical report). *Edinburgh University*.