Improved Stiction Detection via Hybrid Residual Embedded Inception Module Networks

Muhammad Faisal Aftab * **Wahiba Bounoua** *
Muhammad Hamza Zafar * **Filippo Sanfilippo** *

* *Department of Engineering Sciences, University of Agder, 4879 Grimstad, Norway*

Abstract: Stiction in control valves, among other problems, presents a formidable challenge in industrial control loops, often resulting in suboptimal system performance. Given its significant impact, stiction detection has become a crucial aspect of controller performance monitoring. While machine learning-based methods for stiction detection have gained traction, this paper investigates the effectiveness of Inception Networks and Inception-Residual Networks as potential enhancements to the previously proposed CNN method. The results highlight that these adjustments improve accuracy, from 75.3% to 79.45%, using the same training dataset, effectively capturing variations overlooked by other methods. The application of real industrial data highlights the improvements offered by the proposed framework.

Keywords: Stiction, performance monitoring, inception-residual networks, convolutional neural networks.

1. INTRODUCTION

The performance of industrial control loops is pivotal for ensuring the optimal operation of any industrial enterprise. Any degradation in performance can lead to economic and environmental consequences. Thus, automated monitoring of control loop performance plays a vital role in sustaining the desired operational standards. Control loop performance monitoring (CPM) is instrumental in identifying and rectifying issues such as oscillations, non-linearities, external disturbances, and other common problems encountered in process control loops (Bounoua et al., 2022).

Stiction, derived from static friction, has been reported to be the prevalent cause of performance deterioration in industrial control loops. Several sources have documented the detrimental effects of stiction, highlighting the importance of implementing effective monitoring and mitigation strategies (Horch, 2007; Bacci di Capaci and Scali, 2018; Bounoua et al., 2022; Choudhury et al., 2004). Stiction occurs due to elevated static friction within the valve, causing it to stick in one position. As the control signal increases to overcome static friction, the valve suddenly jumps to a new position (slip phase), where it may stick again. This phenomenon is also exhibited during changes in the valve motion direction. Such a stick-slip behavior is typically represented by a distinct shape in the plot of controller output (OP) vs valve position (manipulated variable or MV) as shown in Figure 1. The detailed description can be seen in Bounoua et al. (2022); Horch (2007); Shoukat Choudhury et al. (2005).

Stiction detection has garnered considerable attention from the research community and both model-based (Srinivasan et al., 2005; Jelali, 2008; Karra and Karim, 2009) and data-driven approaches have been proposed.

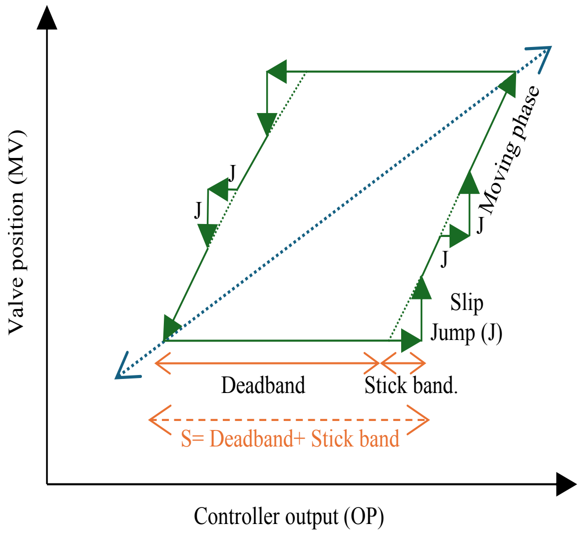

Fig. 1. Stiction phenomenon: Valve position (MV) vs controller output (OP)

Among these, Data-driven methods, in particular, have gained prominence due to their reliance solely on recorded data. These methods encompass a variety of techniques, including pattern recognition, time series analysis, decomposition techniques, and waveform shape analysis.

Pattern recognition and waveform shape analysis techniques have been explored because stiction-induced oscillations exhibit characteristic patterns in the waveforms of controller output, valve position, control error, and process measurement (process variable or PV) signals. Notable works in this area include identifying asymmetry in error signals (Singhal and Salsbury, 2005), fitting data to predefined shapes (Rengaswamy et al., 2001; Rossi and Scali, 2004), analyzing the shape of PV-OP plots (Yamashita, 2006; Scali and Ghelardoni, 2008; Kano et al., 2004; Yamashita, 2004).

Given the widespread adoption of machine learning tools across various domains, their applicability in detecting stiction has also been investigated. Recently, Akavalappil et al. (2023) introduced the 1-D convolutional neural network (CNN) for stiction detection using PV, and OP data. Henry et al. (2020, 2021) used a combination of AlexNet convolutional neural network (CNN) and principle component analysis (PCA) for stiction detection. Additionally, a simple artificial neural network (ANN) is used by Dambros et al. (2019b) for pattern recognition of PV-OP plots. Kamaruddin et al. (2020) proposed a butterfly shape-based detection (BSD) CNN, which derives a butterfly shape from PV/OP data, employing Stenman's stiction model.

In Bounoua et al. (2023), we utilized the Poincaré plot derived from OP data as an input for the stiction detection method employing a convolutional neural network (CNN). This approach achieved an accuracy of 75.3% while offering the advantage of utilizing only a single variable. In the current study, we extend this concept to explore the efficacy of the Inception Network and the Inception-Residual Network for stiction detection by leveraging the Poincaré plot. Our findings reveal that the Inception-based network, known for its computational efficiency, outperforms traditional CNNs. Furthermore, it's essential to acknowledge the challenges associated with obtaining comprehensive data from real-world plants. Analyzing the performance of these machine learning techniques on limited datasets is particularly relevant, as acquiring data that encompasses all possible scenarios from actual plants proves to be quite challenging.

The structure of the paper is as follows: Section 2 provides an overview of the adopted Inception Network, while Section 3 delves into the specifics of the employed framework. The results of the proposed framework are discussed in Section 4, followed by the concluding remarks in Section 5.

2. HYBRID RESIDUAL EMBEDDED INCEPTION NETWORK (RESINCEPNET)

Convolutional Neural Networks (CNNs) have been at the forefront of advancements in deep learning, particularly in image processing and computer vision tasks. The inception module and residual connections are two pivotal architectures that have significantly contributed to these advancements. The proposed technique combines these two architectures to enhance feature extraction and facilitate the training of deeper networks. The detailed architecture is shown in Figure 2.

2.1 Inception Module

The inception module employs parallel convolutional operations with different kernel sizes to capture information at various scales. The operations within the inception module can be represented mathematically as follows:

1x1 Convolution

$$C_{1x1}(X) = \sigma(W_{1x1} * X + b_{1x1}) \tag{1}$$

3x3 Convolution

$$C_{3x3}(X) = \sigma(W_{3x3} * X + b_{3x3}) \tag{2}$$

5x5 Convolution

$$C_{5x5}(X) = \sigma(W_{5x5} * X + b_{5x5}) \tag{3}$$

Where X is the input tensor, W represents the weights of

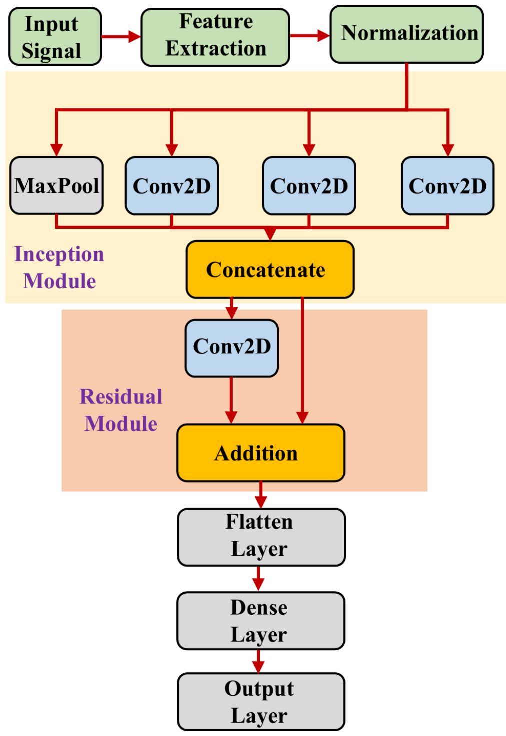

Fig. 2. Detailed Architecture of Proposed ResIncepNet Model

the convolutional filters, b is the bias, and σ denotes the ReLU activation function.

The outputs of these parallel operations, including max pooling, are concatenated along the channel dimension:

$$O_{\text{inception}} = \text{Concatenate}\Big(\Big[C_{1\times1}(X), C_{3\times3}(X), C_{5\times5}(X), \text{MaxPool}(X)\Big]\Big) \tag{4}$$

2.2 Residual Inception Block

The residual inception block introduces a shortcut connection that adds the input of the block to its output:

$$R(X) = \sigma(W_r * X + b_r) \tag{5}$$

Where $*$ is the convolution operator. The final output of the block is:

$$O_{residual} = O_{inception} + R(X) \tag{6}$$

This approach allows the network to learn identity mappings efficiently while benefiting from the inception module's feature extraction capabilities.

The residual connections in the Residual Inception Block were chosen to help the model learn better by making it easier for information to pass through layers without getting lost. This connection skips over some layers and directly adds the original input back to the output after those layers. By doing this, the gradient—the small updates that help the model learn—can flow back through the network more smoothly. This setup prevents the model from getting stuck in a situation where learning slows down as layers get deeper. In short, residual connections keep information flowing and help the model learn faster and more accurately.

2.3 Hyperparameters of the Proposed Model

The model's architecture is built around a convolutional structure with residual inception blocks designed for classification. It begins with an input image size of 100×100 pixels in grayscale (1 channel). The initial convolutional layer has 128 filters with a kernel size of 3×3 and uses the ReLU activation function, followed by MaxPooling to reduce spatial dimensions. The core of the model lies in two residual inception blocks, each configured with 16 filters across three types of convolution kernels: 1×1, 3×3, and 5×5, providing multi-scale feature extraction, with a residual connection to enhance gradient flow. These blocks use ReLU activations, and a dropout rate of 0.5 is applied after the second block to reduce overfitting. The network is then flattened and connected to a fully connected layer with 128 units, followed by a final softmax layer with 2 units for binary classification. The model is optimized using the Adam optimizer with a learning rate of 0.0001, a sparse categorical cross-entropy loss function (with logits), and accuracy as the evaluation metric, trained over 20 epochs. These hyperparameters (given in Table 1) are selected via grid search to balance model complexity, training stability, and accuracy.

Table 1. Hyperparameters and Settings of the Model

Hyperparameter	**Value**
Initial Convolution Activation	ReLU
Pooling Type	MaxPooling
Inception Module Filter Channels	16
Inception Module Kernels	(1,1), (3,3), (5,5)
Inception Activation	ReLU
Residual Connection Activation	ReLU
Convolution Filters	128
Convolution Kernel	(3, 3)
Dropout Rate	0.5
Fully Connected Layer Units	128
Output Layer Units	2
Output Activation	Softmax
Optimizer	Adam
Learning Rate	0.0001

3. FRAMEWORK

The framework employed in this study is adopted from Bounoua et al. (2023) to enable a fair comparison with the conventional CNNs performance. The detailed procedure is illustrated in Figure 3, while this section outlines the crucial steps involved in implementing the framework.

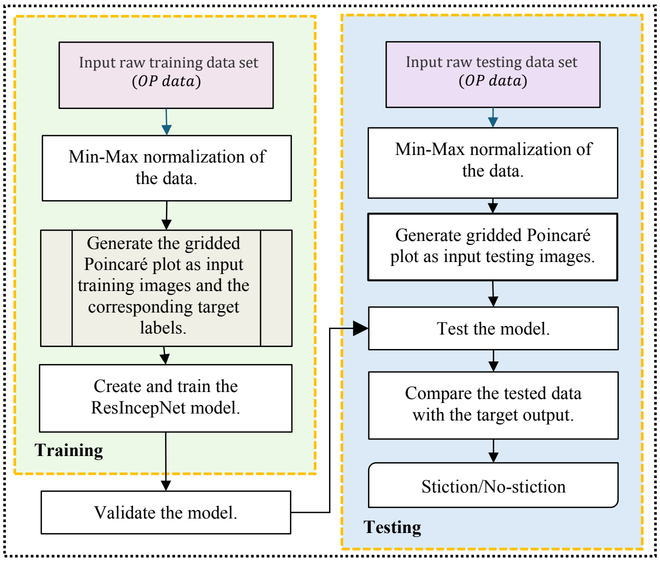

Fig. 3. Flowchart of the proposed ResIncepNet-based stiction detection framework

3.1 Data Scaling

Since the time series data (OP data) originates from different control loops with differing scales and measurements, it becomes imperative to scale this data appropriately to ensure the correct functioning of the neural network despite these variations. Thus, the time series data $\mathbf{x}$ is normalized to fall within the range of [0, 1] using the following relation (Bounoua et al., 2023):

$$\mathbf{x}_{\text{rescaled}} = \left[\frac{\mathbf{x} - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})} \right] \tag{7}$$

The normalized data, thus obtained, is used to generate a Poincaré plot as discussed in the subsequent section.

3.2 Poincaré plot

The Poincaré plot, with its roots in chaos theory, serves as a powerful tool for visualizing and analyzing the dynamics of time series data. By representing each sample as a function of the preceding one, it uncovers patterns and correlations that may not be captured by traditional analysis methods. This plot provides valuable insights into the underlying structure and behavior of complex systems across various domains, including physiology, neuroscience, and engineering.(Satti et al., 2019; Bounoua et al., 2023; Golińska, 2013).

The Poincaré plot is constructed by creating a scatter plot from the reconstructed phase space X, of a time series $\mathbf{x} = \{x_1, x_2, \ldots, x_N\} \in R^N$, with embedding dimension 2 and time lag $\tau = 1$ given by:

$$X = \begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \\ x_3 & x_4 \\ \vdots & \vdots \\ x_{N-1} & x_N \end{bmatrix} \tag{8}$$

The quantitative analysis of the Poincaré plot often involves fitting an ellipse to it and then calculating the standard deviations SD1 and SD2 along the minor and major axes, respectively.

The Poincaré plot is constructed from the scaled data in this study.

3.3 Gridded Poincaré Plot

The Poincaré plot, given in Section 3.2, is further refined into a gridded Poincaré plot by segmenting the graph into a 64x64 grid, achieved through evenly spaced horizontal and vertical grids (Yan et al., 2019). This segmentation allows more effective pattern and trend identification which may not be visible otherwise. The gridded Poincaré plot represented as an image, is then fed to the Hybrid Residual EMbedded Inception Network for stiction detection.

The preference for the gridded Poincaré plots is due to their ability to reveal distinctive patterns in data from stiction-affected control loops. To illustrate, Figures 4 and 5 depict both traditional and gridded Poincaré plots for OP data from two industrial loops—one with stiction and the other without. In the presence of stiction, an elliptical pattern is quite obvious, whereas the plot for the non-stiction scenario lacks such a distinctive shape. This segmentation enhances pattern recognition and facilitates clearer differentiation between stiction and non-stiction cases.

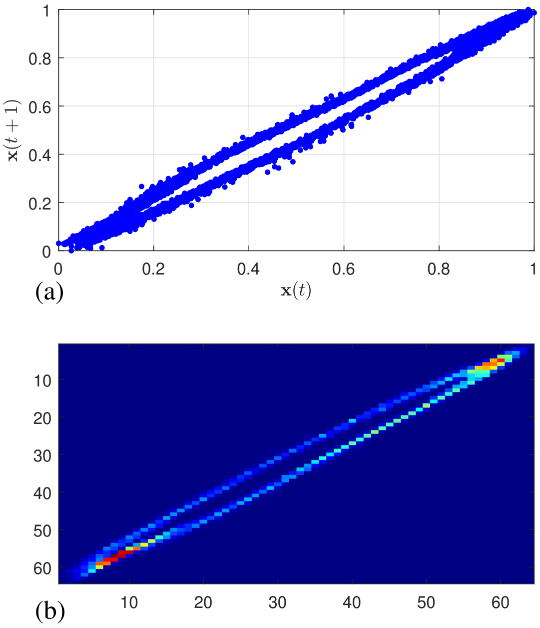

Fig. 4. The Poincaré plots generated from OP data for stiction case (a) Poincaré plot (b) gridded Poincaré plot (image)

4. RESULTS AND DISCUSSION

4.1 Training Data

Ideally, training any machine learning model for stiction detection should involve utilizing industrial data that encompasses a wide range of operating conditions. However, obtaining such comprehensive data is often challenging due to industrial stakeholders' various considerations (Dambros et al., 2019a; Bounoua et al., 2023).

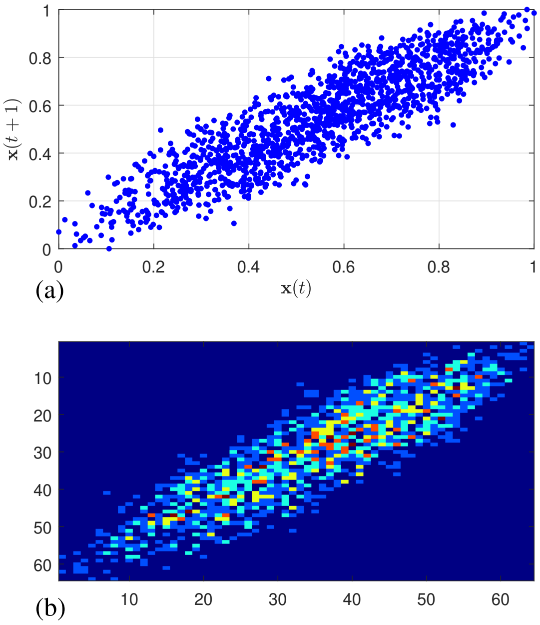

Fig. 5. The Poincaré plots generated from OP data for no stiction case (a) Poincaré plot (b) gridded Poincaré plot (image)

Thus, simulation data emerges as the only viable alternative. Consequently, a simulation model, implemented by Bounoua et al. (2023), is used. This model encompasses a single input and single output system, integrating stiction models sourced from the works of Shoukat Choudhury et al. (2005); Kano et al. (2004), to generate the required training data. The simulation data conditions mirror those utilized in Bounoua et al. (2023), ensuring a fair comparison between inception-based methods and conventional CNNs.

A total of 1000 datasets were generated, encompassing both stiction and non-stiction cases. To train the proposed model, the simulated data was divided into training, validation, and testing datasets with ratios of 60%, 20%, and 20%, respectively. The ResIncepNet Model underwent extensive training to fine-tune its performance optimally. Figures 6 and 7 illustrate the training and validation loss, as well as the training and validation accuracies, providing valuable insight into the model's learning dynamics over time. These figures demonstrate a clear trend of decreasing loss and increasing accuracy with the progression of epochs, indicating effective learning and convergence of the model. The close alignment between training and validation metrics indicates that the ResIncepNet model struck a desirable balance, minimizing overfitting while maximizing its generalization ability on unseen data. This preparatory phase was crucial in ensuring the robustness and reliability of ResIncepNet for stiction detection in industrial control systems.

4.2 Industrial Case Study Results

After extensive training, the proposed model underwent testing on previously unseen industrial data for stiction detection. The Industrial Case Study, sourced from Jelali and Huang (2010), consists of 73 control loop datasets spanning various industrial processes.

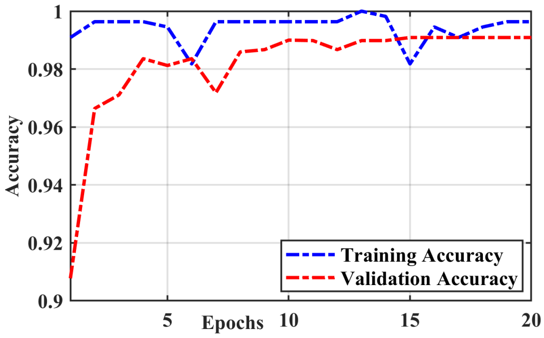

Fig. 6. Training and Validation Accuracy vs Epochs on Simulation Data for ResIncepNet

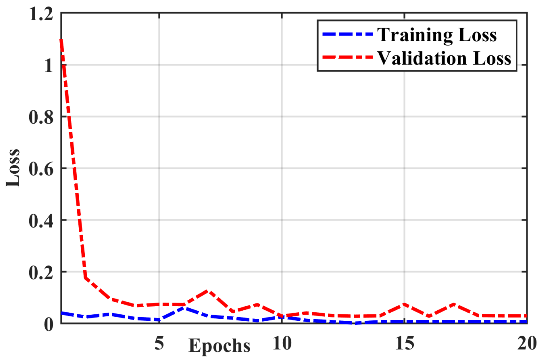

Fig. 7. Training and Validation Loss vs Epochs on Simulation Data for ResIncepNet

The results presented in Table 2 offer a comprehensive comparison of the three competing techniques for stiction detection in industrial control valves: ResInception Networks (ResIncepNet), Inception Networks (IncepNet), and the previously studied Convolutional Neural Network with specialized stiction convolution layers (SConvNet). The confusion matrix of the proposed model is also shown in Figure 8. The performance metrics used for comparison include accuracy, precision, recall, and the F1-score, which collectively provide a robust framework for assessing the effectiveness of each method in detecting the stiction.

Accuracy is the most intuitive performance metric, representing the overall correctness of the model across all classes. In this case, ResIncepNet leads with an accuracy of 79.45%, closely followed by IncepNet at 78.08%, and SConvNet at 75.30% (given in Table 2). The higher accuracy of ResIncepNet suggests its superior ability to correctly identify both the presence and absence of stiction in the control valves, indicating a more balanced performance across different operational scenarios.

Table 2. Evaluation Matrix Comparison of Competing Techniques for Stiction Detection

Method	Accuracy	Precision	Recall	F1-Score
ResIncepNet	79.45%	76.47%	78.79%	77.61%
IncepNet	78.08%	75.76%	75.76%	75.76%
SConvNet	75.30%	71.43%	75.76%	73.53%

Precision measures the model's accuracy in predicting positive instances (stiction detected) among all positive predictions made, while recall (sensitivity) assesses the model's ability to find all actual positive instances. ResIncepNet achieves the highest precision at 76.47% and the highest recall at 78.79%, indicating not only its reliability in confirming cases of stiction but also its competence in detecting the majority of actual stiction cases. These metrics suggest that ResIncepNet maintains a favorable balance between reducing false positives (misidentifying normal operation as stiction) and minimizing false negatives (overlooking actual cases of stiction).

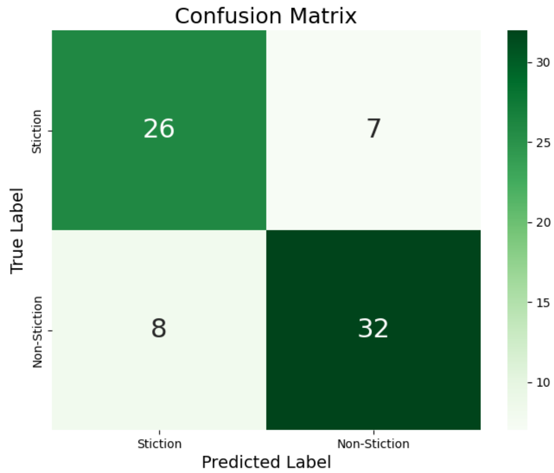

Fig. 8. Confusion Matrix of the Proposed ResIncepNet Model

The F1-score is a harmonic mean of precision and recall, providing a single metric to measure the balance between them. It is particularly useful when the cost of false positives and false negatives is similar or when the class distribution is imbalanced. ResIncepNet leads with an F1-score of 77.61%, followed by IncepNet at 75.76%, and SConvNet at 73.53%. The superior F1-score of ResIncepNet further confirms its effectiveness as a balanced and robust technique for stiction detection, excelling in both aspects of precision and recall.

The performance metrics highlight the advantages of leveraging advanced neural network architectures like ResIncepNet for stiction detection. The improved accuracy and balanced precision-recall performance of ResIncepNet can be attributed to its sophisticated architecture that combines the benefits of Inception modules with residual connections. This design enables it to extract and leverage complex features from control valve data, which are essential for accurately identifying stiction. Furthermore, the residual connections help mitigate the vanishing gradient problem, allowing deeper network architectures without compromising training effectiveness.

In comparison, IncepNet also demonstrates commendable performance, surpassing SConvNet in all metrics. This indicates the inherent advantage of the Inception architecture in capturing multi-scale information, which is crucial for detecting stiction patterns that may vary significantly in their temporal characteristics.

5. CONCLUSIONS

In this paper, an improved stiction detection scheme based on hybrid residual embedded inception module networks

(ResIncepNet) has been presented. The proposed scheme demonstrates a significant improvement in accuracy compared to the CNN-based approach used in the previous work. The incorporation of residual connections and inception modules leads to better feature extraction, enabling the model to capture variations that are missed by other methods. Moreover, the evaluation of the proposed scheme, conducted through various performance metrics such as accuracy, precision, recall, and the F1-score, emphasizes the superiority of the approach in stiction detection. Future work will focus on further refining the model architecture, exploring its applicability in different industrial settings, and investigating its potential for real-time implementation in industrial control systems.

REFERENCES

Akavalappil, V., Radhakrishnan, T.K., and Dave, S.K. (2023). A convolutional neural network (cnn)-based direct method to detect stiction in control valves. *The Canadian Journal of Chemical Engineering*.

Bacci di Capaci, R. and Scali, C. (2018). Review and comparison of techniques of analysis of valve stiction: From modeling to smart diagnosis. *Chemical Engineering Research and Design*, 130, 230–265.

Bounoua, W., Aftab, M.F., and Omlin, C.W.P. (2022). Controller Performance Monitoring: A Survey of Problems and a Review of Approaches from a Data-Driven Perspective with a Focus on Oscillations Detection and Diagnosis. *Industrial & Engineering Chemistry Research*, acs.iecr.2c02785.

Bounoua, W., Aftab, M.F., and Omlin, C.W.P. (2023). Stiction detection in industrial control valves using poincaré plot-based convolutional neural networks. *IFAC-PapersOnLine*, 56(2), 11687–11692.

Choudhury, M., Shah, S.L., and Thornhill, N.F. (2004). Detection and quantification of control valve stiction. *IFAC Proceedings Volumes*, 37(9), 865–870.

Dambros, J.W.V., Trierweiler, J.O., Farenzena, M., and Kloft, M. (2019a). Oscillation Detection in Process Industries by a Machine Learning-Based Approach. *Industrial & Engineering Chemistry Research*, 58(31), 14180–14192.

Dambros, J.W., Farenzena, M., and Trierweiler, J.O. (2019b). Oscillation detection and diagnosis in process industries by pattern recognition technique. *IFAC-PapersOnLine*, 52(1), 299–304.

Golińska, A.K. (2013). Poincaré plots in analysis of selected biomedical signals. *Studies in logic, grammar and rhetoric*, 35(1), 117–127.

Henry, Y., Aldrich, C., and Zabiri, H. (2020). Detection and severity identification of control valve stiction in industrial loops using integrated partially retrained CNN-PCA frameworks. *Chemometrics and Intelligent Laboratory Systems*, 206, 104143.

Henry, Y., Aldrich, C., and Zabiri, H. (2021). Control valve stiction detection by use of alexnet and transfer learning. In *E3S Web of Conferences*, volume 287, 03012. EDP Sciences.

Horch, A. (2007). *Benchmarking Control Loops with Oscillations and Stiction*, 227–257. Springer London, London.

Jelali, M. (2008). Estimation of valve stiction in control loops using separable least-squares and global search algorithms. *Journal of Process Control*, 18(7-8), 632–642.

Jelali, M. and Huang, B. (eds.) (2010). *Detection and Diagnosis of Stiction in Control Loops*. Advances in Industrial Control. Springer London, London. doi: 10.1007/978-1-84882-775-2.

Kamaruddin, B., Zabiri, H., Amiruddin, A.M., Teh, W., Ramasamy, M., and Jeremiah, S. (2020). A simple model-free butterfly shape-based detection (bsd) method integrated with deep learning cnn for valve stiction detection and quantification. *Journal of Process Control*, 87, 1–16.

Kano, M., Maruta, H., Kugemoto, H., and Shimizu, K. (2004). Practical model and detection algorithm for valve stiction. *IFAC Proceedings Volumes*, 37(9), 859–864.

Karra, S. and Karim, M.N. (2009). Comprehensive methodology for detection and diagnosis of oscillatory control loops. *Control Engineering Practice*, 17(8), 939–956.

Rengaswamy, R., Hägglund, T., and Venkatasubramanian, V. (2001). A qualitative shape analysis formalism for monitoring control loop performance. *Engineering Applications of Artificial Intelligence*, 14(1), 23–33.

Rossi, M. and Scali, C. (2004). Automatic detection of stiction in actuators: A technique to reduce the number of uncertain cases. *IFAC Proceedings Volumes*, 37(9), 751–756.

Satti, R., Abid, N.U.H., Bottaro, M., De Rui, M., Garrido, M., Raoufy, M.R., Montagnese, S., and Mani, A.R. (2019). The Application of the Extended Poincaré Plot in the Analysis of Physiological Variabilities. *Frontiers in Physiology*, 10, 116.

Scali, C. and Ghelardoni, C. (2008). An improved qualitative shape analysis technique for automatic detection of valve stiction in flow control loops. *Control Engineering Practice*, 16(12), 1501–1508.

Shoukat Choudhury, M., Thornhill, N., and Shah, S. (2005). Modelling valve stiction. *Control Engineering Practice*, 13(5), 641–658.

Singhal, A. and Salsbury, T.I. (2005). A simple method for detecting valve stiction in oscillating control loops. *Journal of Process Control*, 15(4), 371–382.

Srinivasan, R., Rengaswamy, R., Narasimhan, S., and Miller, R. (2005). Control loop performance assessment. 2. hammerstein model approach for stiction diagnosis. *Industrial & engineering chemistry research*, 44(17), 6719–6728.

Yamashita, Y. (2004). Qualitative analysis for detection of stiction in control valves. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 391–397. Springer.

Yamashita, Y. (2006). An automatic method for detection of valve stiction in process control loops. *Control Engineering Practice*, 14(5), 503–510.

Yan, C., Li, P., Liu, C., Wang, X., Yin, C., and Yao, L. (2019). Novel gridded descriptors of poincaré plot for analyzing heartbeat interval time-series. *Computers in Biology and Medicine*, 109, 280–289.