Advance Sensor Signal Validation using Attention-based Deep Learning Models

Chau Tran [a*] and Mario Hoffmann [a]
[a]Institute for Energy Technology (IFE), Halden, Norway

Abstract: Condition-Based Maintenance (CBM) involves a continuous oversight and analysis of data from sensors to maintain the health and performance of equipment or systems. It aims to enhance operational efficiency, reduce downtime, and prevent unexpected failures; thus, playing a crucial role in critical industries, particularly in the context of nuclear power plants. To monitoring the power plant in detail, numerous sensors are installed leading to a concern on how to maintain these sensor's reliability over time. Therefore, this paper focuses on the validation of sensor measurements. In particular, the sensor signal validation is first identified as a nonlinear autoregressive exogenous problem, implying that the model generated value of a signal relies on a nonlinear mapping function involving its historical records and external factors. Then, various attention-based deep learning models are explored, namely, Dual-stage attention-based Recurrent Neural (DA-RNN) and Informer model. These models are assessed through a real case study collected from the boiling water reactor, encompassing 77 features. Evaluation criteria for the models include the Mean Square Error (MSE), providing insights into their effectiveness in signal validation for CBM. All models include an attention mechanism which identifies dominant input factors among spatial and/or temporal dimension. The study finds that these models yield varied prediction or reconstruction results between different signals. Hypothetically, the attention mechanism may unintentionally favor the dominant signal due to high correlations between certain signals.

Keywords: Condition-Based Maintenance, Signal Validation, Attention-based Deep Learning.

1. INTRODUCTION

Condition-based Maintenance (CBM) is a strategy employed across various industries to minimize failure occurrences and costs. CBM comprises four key components: condition monitoring, fault detection, diagnostics, and prognostics. During condition monitoring, real-time data is gathered from ambient and/or built-in sensors to provide an overview of the process or piece of equipment. This data is analyzed to detect faults when parameters exceed set thresholds. If a fault is detected, diagnostics identify its characteristics such as location and severity. Prognostics then use the data from the previous stages to predict potential failure events, aiding in the creation of a maintenance schedule. This study primarily focuses on the condition monitoring and fault detection stages.

Given its distinctive nature, the nuclear industry prioritizes safety intensely. To prevent accidents, both the systems and facilities are rigorously monitored and tested. One approach involves the extensive use of sensors. In addition to those embedded within the equipment, numerous sensors are installed in the surrounding environment. These sensors generate a vast amount of signal data crucial for condition monitoring. However, to avoid misleading information and enhance the reliability of the sensor-based approach, it is essential to monitor the status of the sensors, particularly to identify any malfunctions. By integrating sensor validation into the fault detection process, the Condition-Based Maintenance (CBM) becomes more reliable and trustworthy.

This study is a culmination of ongoing efforts that have been informed by a range of past practices, all aimed at enhancing the functionality of a toolbox known as PEANO [1]. Initially introduced to the public in 1998 as a real-time process signal validation system, PEANO combines fuzzy and possibilistic logic models for clustering with Artificial Neural Networks for signal validation. Over time, there have been several endeavours to improve the reliability and capability of this system. These efforts include the development of wavelet-based denoising filters [2] and the restructuring of the architecture into an ensemble of regression models to address large-scale models [3]. As technology has advanced, particularly with the rise of attention-based deep learning models following the introduction of Transformer [4], this study seeks to explore the application of this new

approach in terms of signal validation and fault detection, in alignment with the ongoing pursuit of refining PEANO's capabilities.

Two attention-based deep learning models have been selected for implementation and examination: the Dual-stage attention-based Recurrent Neural Network (DA-RNN) model [5] and the Informer model [6]. The Dual-stage Attention model incorporates two attention mechanisms to extract spatial and temporal properties within the input sequence. the Informer model, inspired by the Transformer, uses a self-attention mechanism to capture the relevance score of each item. It also introduces a uniform input representation by incorporating timestamps into the feature vectors. Both models have demonstrated promising results in handling long sequence inputs.

In this work, these two models are implemented and tested on a real dataset collected from a nuclear Boiling Water Reactor located in a Scandinavian country. The dataset consists of 77 signals.

Section 2 and Section 3 provide detailed descriptions of the two attention-based deep learning models and their implementations on the Boiling Water Reactor dataset. Section 4 presents the conclusions and proposes future work.

2. ATTENTION-BASED DEEP LEARNING MODEL

The attention mechanism was introduced as an improvement to the conventional Encoder-Decoder model. In the standard model, only the last stage of the encoder is utilized as input for the decoder. However, as the length of the input sequence increases, the Encoder-Decoder model's performance declines, posing a challenge for the encoder to retain and deliver all pertinent information to the decoder. The attention method addresses this by prioritizing relevant information in the inputs through attention weights. This approach has been effectively employed in Sequence-to-Sequence tasks, especially in machine translation. Consequently, multiple efforts have aimed to leverage this mechanism in tackling long sequence time-series forecasting or prediction. Given that sensor signal data over a monitoring period can be construed as a long sequence time-series, this study explores promising attention-based neural networks that have demonstrated noteworthy performance: Dual-Stage Attention-Based RNN and Informer model.

2.1. Dual-Stage Attention-Based RNN (DA-RNN)

While attention-based encoder-decoder networks have demonstrated strong performance in tasks such as machine translation or image captioning, they have not yielded significant results with time-series data. This is primarily due to the complexity of time-series data, which typically contains multiple signal inputs, posing a challenge for attention-based models to discern the driving factors necessary for accurate predictions.

To address this challenge, DA-RNN model was introduced. This model incorporates a dual-stage attention mechanism inspired by studies on human behavior, which depict a two-phase selective mechanism [7]. In the first stage, the model extracts relevant input features' information at each time step by referencing the previous encoder hidden state. Subsequently, in the second stage, a temporal attention mechanism selects relevant information across all time steps. By employing these two attention mechanisms, the model not only identifies pertinent input features but also captures long-term temporal dependencies inherent in time-series data.

2.1.1 Problem Statement

To best fit the DA-RNN idea, the signal monitoring of this study is conceptualized as a Nonlinear Autoregressive Exogenous (NARX) problem. This signifies that the prediction values of a signal are determined by its historical data as well as other signal series. Then, the diagnosis of multiple sensor signals can be mathematically formed as:

- The current and past values of n driving signals, which don't include the target signal, are notated as $\mathbf{X} = (\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^n)^\top = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_T) \in \mathbb{R}^{n \times T}$, where T is the window size. In which,
    - $\boldsymbol{x}^k = (x_1^k, x_2^k, \dots, x_T^k) \in \mathbb{R}^T$: representing a signal k series during the T period.

- $\boldsymbol{x}_t = (x_t^1, x_t^2, \dots, x_t^n)^\top \in \mathbb{R}^n$: representing a vector of n exogenous input series at time step t.

- The previous values of the target signal y_T is notated as $y = (y_1, y_2, \dots, y_{T-1})$ with $y_t \in \mathbb{R}$.
- The current value of the target signal y_T is $\hat{y}_T = F(y, \mathbf{X}) = F(y_1, \dots, y_{T-1}, \boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_T)$, where $F(\cdot)$ is a nonlinear mapping function.

2.1.2 DA-RNN Methodology

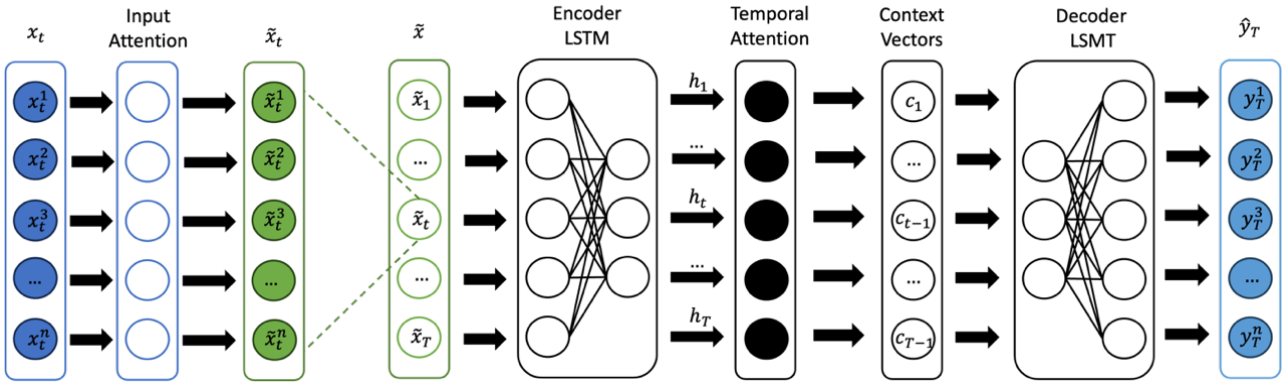

Figure 1: Dual-Stage Attention Architecture [5]

The DA-RNN process model can be depicted as shown in Figure 1, generated based on [5]. It comprises two primary blocks: the input attention mechanism, located on the left, responsible for selecting the relevant driving series, and the temporal attention mechanism, positioned in the middle, responsible for selecting relevant encoder hidden states across different time steps. The LSTM serves as the fundamental block for both the encoder and decoder components.

The attention mechanism for the spatial or temporal context of the DA-RNN model is computed using a common formula proposed by [8] to address the information limitation encountered when processing input data over extended periods. This formula encompasses three key computational components:

- Alignment scores $e_t^k = a(s_{t-1}, h_{t-1})$, with $\boldsymbol{s}_{t-1}$ and $\boldsymbol{h}_{t-1}$ being previous cell stage and hidden stage, respectively: These scores quantify the alignment between elements of the input series and the current output, indicating how well they correspond.
- Weights $\alpha_t^k = softmax(e_t^k)$: These weights are derived by applying the SoftMax function to the alignment scores, thereby providing a measure of importance for each element of the input series.
- Context vector $\boldsymbol{c}_t = \sum_{i=1}^{T} \alpha_t^i \boldsymbol{h}^i$, where $\boldsymbol{h}^i$ is the i^{th} encoder hidden state: This vector encapsulates the useful information extracted from the input series, incorporating the weighted contributions based on the alignment scores.

Applying the Bahdanau attention mechanism, DA-RNN computes the dual-stage attentions, the special (input) attention and the temporal attention, as below:

- Spatial attention: Given the input data of k^{th} signal $x^k = (x_1^k, x_2^k, \dots, x_T^k)^\top \in \mathbb{R}^T$, the previous end hidden state as $\boldsymbol{h}_{t-1} \in \mathbb{R}^p$, and the cell stage as $\boldsymbol{s}_{t-1}^e \in \mathbb{R}^p$ of the attention, the alignment scores (e_t^k), the weights (α_t^k), and the context vector or new input $(\tilde{\boldsymbol{x}}_t)$ are calculated as:
  - $e_t^k = \boldsymbol{v}_e^\top \tanh(\boldsymbol{W}_e[\boldsymbol{h}_{t-1}; \boldsymbol{s}_{t-1}^e] + \boldsymbol{U}_e \boldsymbol{x}^k))$, where $\boldsymbol{v}_e \in \mathbb{R}^T, \boldsymbol{W}_e \in \mathbb{R}^{T \times 2m}, \boldsymbol{U}_e \in \mathbb{R}^{T \times T}$ are learning parameters.
  - $\alpha_t^k = softmax(e_t^k)$
  - $\tilde{\boldsymbol{x}}_t = (\alpha_t^1 x_t^1, \alpha_t^2 x_t^2, \dots, \alpha_t^n x_t^n)^\top$

- Temporal attention: Given the previous decoder hidden state as $\boldsymbol{d}_{t-1} \in \mathbb{R}^p$ and the cell stage as $\boldsymbol{s}_{t-1}^d \in \mathbb{R}^p$ of the attention, the alignment scores (l_t^i), the weights (β_t^i), and the context vector ($\boldsymbol{c}_t$) are calculated as:
    - $l_t^i = \boldsymbol{v}_d^\top tanh(\boldsymbol{W}_d[\boldsymbol{d}_{t-1}; \boldsymbol{s}_{t-1}^d] + \boldsymbol{U}_d \boldsymbol{h}_i))$, where $\boldsymbol{v}_d \in \mathbb{R}^T, \boldsymbol{W}_d \in \mathbb{R}^{T \times 2m}, \boldsymbol{U}_d \in \mathbb{R}^{T \times T}$ are learning parameters.
    - $\beta_t^i = softmax(l_t^i)$
    - $\boldsymbol{c}_t = \sum_{i=1}^T \beta_t^i \boldsymbol{h}_i$

Then, the context vector is fed into the decoder block to generate a prediction of the target signal. The output is derived as:

$$\hat{y}_t = \boldsymbol{v}_y^\top (\boldsymbol{W}_y[\boldsymbol{d}_T; \boldsymbol{c}_T] + \boldsymbol{b}_w) + b_v,$$

where $[\boldsymbol{d}_T; \boldsymbol{c}_T] \in \mathbb{R}^{p \times m}$ is the concatenation of the decoder hidden state and the context vector.

2.2. Informer

Informer, a Transformer-based model, is another proposed model to tackle the unsatisfactory performance when dealing with long sequence time-series data. Transformer is a model that was introduced in 2017 by Google Brain [4]. It has an encoder-decoder architecture comprising of multiple layers of self-attention and feedforward neural network. Employing a self-attention mechanism on the input tokens enables the model to prioritize the most relevant aspects of the input data for the given task. Since its inception, it has demonstrated remarkable proficiency in capturing extensive dependencies compared to recurrent-based models, particularly in domains like natural language processing and image processing [9]. However, when dealing with long sequence time-series data, it exhibits several challenges in terms of efficiency, including quadratic time complexity and substantial memory usage. To mitigate these items, Informer introduces three key enhancements: (1) ProbSparse self-attention mechanism, (2) self-attention distillation, and (3) generative-style decoder.

2.2.1 Problem Statement

The signal diagnosis is defined as a Long Sequence Time-series Forecasting (LSTF) problem, which refers to the task of predicting future values or trends in a time series dataset over a long period of time. It is noticeable that LSTF's feature dimension is not limited to univariate case. Mathematically, the input (X^t) and output (Y^t) are formulated as:

- $X^t = \{x_1^t, x_2^t, \dots, x_{L_x}^t | x_i^t \in \mathbb{R}^{d_x}\}$ with L_x and d_x being the window size and the dimension of the input, respectively.
- $\boldsymbol{Y^t} = \{\boldsymbol{y_1^t}, \boldsymbol{y_2^t}, \dots, \boldsymbol{y_{L_y}^t} | \boldsymbol{y_i^t} \in \mathbb{R}^{d_y}\}$ with $\boldsymbol{L_y}$ and $\boldsymbol{d_y d_y}$ being the window size and the dimension of the output, respectively and $\boldsymbol{d_y} \geq \boldsymbol{1}$.

2.2.2 Informer Methodology

ProbSparse Self-attention:

The canonical self-attention mechanism, first described in [4], undertakes a series of linear transformations to map the input sequence into three distinct vectors, denoted as query ($\boldsymbol{Q}$), key ($\boldsymbol{K}$), and value ($\boldsymbol{V}$). Subsequently, the attention mechanism weights the values based on the similarity between the query and key vectors. Then, the summation of the weights and the original input sequence are forwarded to a feed-forward neural network to generate the output. enables the model to selectively attend to pertinent features and discern long-range dependencies within the data. This approach enables the model to selectively attend to the relevant information as well as capture the long-range dependencies. The canonical self-attention is calculated by using the scaled dot-product and formulated as: $attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax\left((\boldsymbol{Q}\boldsymbol{K}^\top)/\sqrt{d}\right)\boldsymbol{V}$, where $\boldsymbol{Q} \in \mathbb{R}^{L_Q \times d}$, $\boldsymbol{K} \in \mathbb{R}^{L_K \times d}$,

$\boldsymbol{V} \in \mathbb{R}^{L_V \times d}$, and d is the input dimension. The $\frac{1}{\sqrt{d}}$ is the scaling factor to avoid the vanishing gradients problem happening when the dot product magnitude grow large with the large input dimension d.

Although the scaled dot product attention mechanism enables simultaneous processing of the entire set of queries, the quadratic computational complexity and $\mathcal{O}(L_Q L_K)$ memory usage with respect to the sequence length makes self-attention inefficient for processing very long sequences. To improve the efficiency, Informer proposes an enhanced self-attention mechanism, named *ProbSparse* self-attention which utilizes the sparsity of self-attention probability distribution and an empirical approximation. In *ProbSparse* self-attention, the query vector ($\boldsymbol{Q}$) is alternated by a sparse matrix ($\bar{\boldsymbol{Q}}$) which contains only the top queries under the sparsity max-mean measurement $\bar{M}(\boldsymbol{q}, \boldsymbol{K})$. In the max-mean measurement, the Log-Sum-Exp $\left(\ln \sum_{j-1}^{L_K} \mathrm{e}^{\frac{q_i k_j^\top}{\sqrt{d}}}\right)$ component is replaced by a max-mean item $\left(\max_j \left\{\frac{q_i k_j^\top}{\sqrt{d}}\right\}\right)$ to avoid the numerical stability issue and the quadratic computation when calculating the dot product in the traditional sparse measurement. These formulas are:

$$\bar{M}(\boldsymbol{q}_i, \boldsymbol{K}) = \max_j \left\{\frac{\boldsymbol{q}_i \boldsymbol{k}_j^\top}{\sqrt{d}}\right\} - \left(\frac{1}{L_K}\right) \sum_j^{L_k} \frac{\boldsymbol{q}_i \boldsymbol{k}_j^\top}{\sqrt{d}}, \text{ with the } i\text{-th query } \boldsymbol{q}_i \in \mathbb{R}^d \text{ and } \boldsymbol{k}_j \in \mathbb{R}^d$$

$$ProbSparse\ attention(\bar{\boldsymbol{Q}}, \boldsymbol{K}, \boldsymbol{V}) = softmax\left(\frac{\bar{\boldsymbol{Q}} \boldsymbol{K}^\top}{\sqrt{d}}\right) \boldsymbol{V}$$

The max-operator in $\bar{M}(\boldsymbol{q}_i, \boldsymbol{K})$ is less sensitive to the zeros therefore it is more numerical stable than Log-Sum-Exp operator. When the length of queries and keys are equivalent, then the total complexity of time and space are reduced to $\mathcal{O}(L \ln L)$ with $L = L_K = L_Q$.

Self-attention Distilling:

To reduce the memory bottleneck when stacking multiple encoder or decoder blocks in dealing with long sequence inputs, Informer introduces the self-attention distilling mechanism which is inspired by the dilated convolution concept. The values forwarding from j-th layer to $(j+1)$-th block is distilled as follow:

$$\boldsymbol{X}_{j+1}^t = MaxPool\left(ELU\left(Conv1d\left(\left[\boldsymbol{X}_j^t\right]_B\right)\right)\right),$$

Where $[\cdot]_B$ presents an encoder/decoder block; $MaxPool(\cdot)$ is used for down-sampling $\mathbf{X}^{\mathrm{t}}$ by half by configuring stride as 2; $ELU(\cdot)$ is an activation function; and $Conv1d(\cdot)$ is a 1-D convolutional filters with kernel width being 3.

Moreover, two additional steps are added to make the distilling process more robustness, which are replication of the main stack with half of the input and reduction of the distilling layers one at a time. Before outputting, all the stack's feature maps are concatenated to make a feature map. This process extracts the dominating attention; consequently, the network size or total space complexity is reduced significantly to $\mathcal{O}\big((2-\epsilon) L \log L\big)$.

Generative Style Decoder:

The Informer decoder shares the same structure as the vanilla Transformer. It consists of a stack of two multi-head attention layers However, it deviates from the conventional approach by incorporating the previous period's input values concatenated with the target sequence instead of solely relying on previously generated outputs. Formally, this concatenation is expressed as follows:

$$\mathbf{X}_{decoder}^{\mathrm{t}} = Concate(\mathbf{X}_{token}^{\mathrm{t}}, \mathbf{X}_0^{\mathrm{t}}),$$

where $\mathbf{X}_{token}^{\mathrm{t}} \in \mathbb{R}^{L_{token} \times d_{model}}$ represents a sequence of length L_{token} in the input and $\mathbf{X}_0^{\mathrm{t}} \in \mathbb{R}^{L_y \times d_{model}}$ denotes the target values, encompassing timestamp and contextual information.

The targets are padded into zeros. The decoder processes the inputs by only one feed forward procedure instead of using a step-by-step approach named "dynamic decoding". The adoption of a generative-style decoder serves to mitigate the computational slowdown associated with long-range predictions.

3. APPLICATION

3.1. Dataset

The dataset utilized in this experiment was sourced from a heat balance system of a 1200 MWth boiling water reactor (BWR) situated in Scandinavia. The measurements within the dataset pertain to a well-instrumented, albeit limited, segment of the process. The heat balance system, integral for verifying and monitoring the overall performance of the plant, undergoes regular calibration.

The dataset includes 77 measurement signals recorded at 10-minute intervals and covers operational states such as start-up, shut-down, and full power. Notably, most of the data corresponds to the full power state, resulting in an underrepresentation of transient situations, which could affect model performance for those process states. Additionally, the dataset does not include a coast-down period, a significant phase of the normal fuel cycle preceding a refuelling outage. Nevertheless, this omission is not critical for evaluating the performance of the model structures proposed in this study.

3.2. Implementation

In this study, each model undergoes two phases: training and testing, ensuring effective and reliable evaluation of the attention-based model's performance. Data is partitioned chronologically to maintain crucial temporal dependencies for effective learning. The training phase utilizes a dataset spanning 7.5 months of normal plant operation, from late May to mid-January, capturing diverse seasonal variations. Specifically, 80% of this timeframe is dedicated to training, while the remaining 20% is allocated for validation and testing. During the testing phase, three distinct test sets are utilized, each comprising data from periods before, during, and after the training period. These test sets contain data not included in the training phase, ensuring unbiased evaluation. Additional details are available in Figure 2.

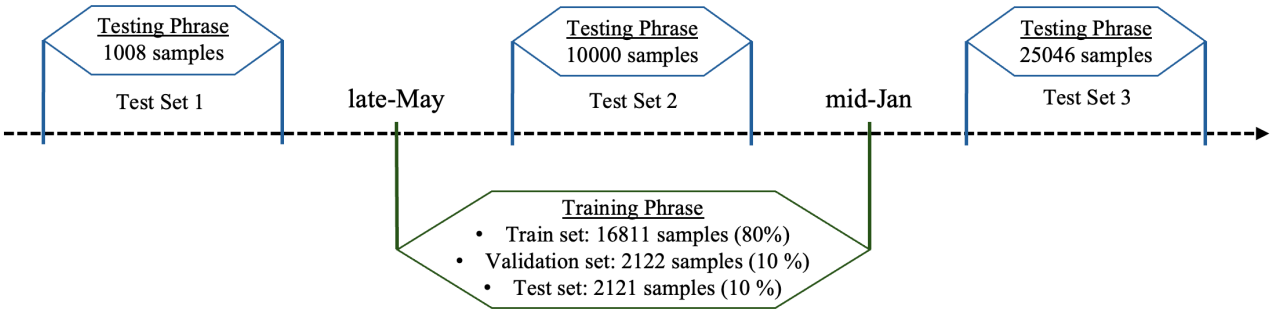

Figure 2: Overview of Dataset Allocation for Training and Testing Phrase

Following partitioning, each subset undergoes preprocessing steps to ensure data quality and compatibility with the models. The primary techniques include normalization using the standardization scaler and data cleaning to address missing values, applied uniformly across both models. Additionally, special care is taken in handling time-series features, ensuring extraction of meaningful patterns tailored for each model's requirements. For the DA-RNN model, data ordering is utilized to extract temporal characteristics. Meanwhile, for the Informer model, a time features vector is generated from timestamps, incorporating both local embedding (data ordering) and global embeddings (Minutes, Hours, Month, and Year) to enhance temporal understanding and feature representation.

Both models have similar general hyperparameters. Theses configurations, presented in Table 1, include the number of epochs, loss function, learning rate, window size, and predicted size. The learning rate is reduced by a factor of 10, when a metric has stopped improving. The models' architecture setups are kept as the default.

Table 1: Models Hyperparameters

Hyperparameters	**Value**
Number of Epoch	100
Loss function	Mean Squared Error
Initial Learning Rate	0.001

Learning Rate Schedule	ReduceLROnPlateau
Window Size	20 samples
Predicted Size	1 sample

3.3. Result and Discussion

In the training phrase, the performance of DA-RNN and Informer model are suboptimal, as indicated by a relatively high MSE of 0.02 and 0.06, respectively. While the models capture some trends in the data, it struggles to accurately forecast future values, leading to errors.

During the testing phase with untrained data, the results exhibit similar trends to those observed during the training stage. The DA-RNN model consistently produces superior predictions, especially in Test set 1 and Test set 3. These models demonstrate enhanced performance on test sets that contain data points overlapping with the training set in terms of time properties (such as day and month), which explains the higher performance in Test set 1 and Test set 3. However, because Test set 2 data was excluded from the training set, the models were not trained on these specific time periods and seasonal conditions, resulting in lower performance on Test set 2.

Besides the overall analysis on the models' performances, individual signals' predictions on the three test sets are reviewed. It is noticeable that there are multiple signals having good results in both models with MSE being less than 10^{-3} or 10^{-4}. An example of the signal number 4 is presented in Figure 3.

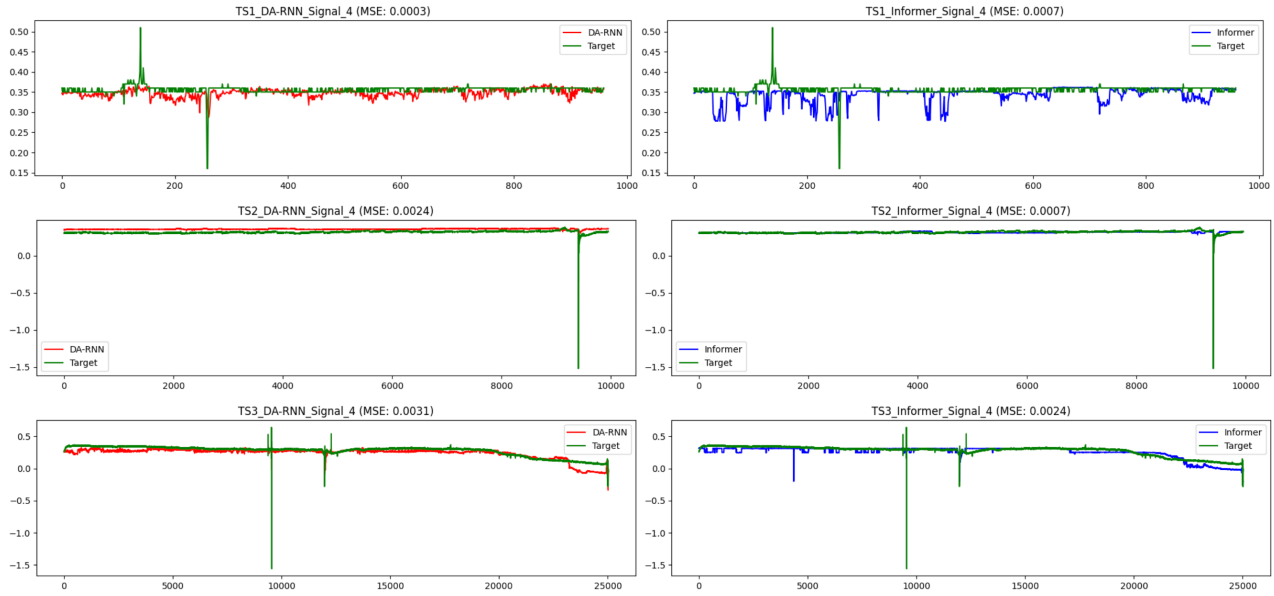

Figure 3: The predictions of signal number 4 in three test sets with both model architectures.

The dataset in both phases contains multiple redundant signals due to the nature of the nuclear power plant, which employs various sensors performing identical functions. These sensors generally yield better prediction results. The attention mechanism prioritizes specific features according to their relevance in the current context. Consequently, if an excessive number of signals exhibit the same values and trends, it can overwhelm the weight matrix.

4. CONCLUSION

The research investigates the feasibility of employing attention-based deep learning models, specifically the DA-RNN and Informer models, for CBM in nuclear power plants. A dataset obtained from a boiling water reactor is utilized to evaluate the models' performance, focusing on their capacity to validate sensor signals effectively.

The analysis reveals diverse prediction or reconstruction outcomes across different signals, potentially attributed to the inadvertent emphasis on dominant signals by the attention mechanism or strong correlations

between certain signals. This observation is underscored by the relatively high Mean Square Errors (MSEs) of 0.02 and 0.06 recorded during the training phase. Nonetheless, several individual signal predictions demonstrate satisfactory results in both models, with MSE values falling below 10^{-3} or 10^{-4}.

In summary, while attention-based deep learning models hold potential for sensor signal validation in CBM for nuclear power plants, challenges such as managing high signal correlations and addressing the influence of redundant signals on the attention mechanism need to be addressed. Further research is essential to refine these models and enhance their predictive accuracy, ensuring robust, reliable and safe CBM implementations in nuclear power plant operations.

Acknowledgements

The work presented in this paper was performed under the OECD Nuclear Energy Agency (NEA) Halden Human Technology Organization (HTO) project.

References

[1] P. F. Fantoni, S. Figedy, and A. Racz, “PEANO, a toolbox for real-time process signal validation and estimation,” Feb. 1998, Accessed: Jun. 15, 2024. [Online]. Available: https://www.osti.gov/etdeweb/biblio/20457892

[2] P. F. Fantoni, M. Hoffmann, S. Lipcsei, and D. Roverso, “PEANO: advancements in 1998-99,” Apr. 1999, Accessed: Jun. 15, 2024. [Online]. Available: https://www.osti.gov/etdeweb/biblio/20913327

[3] G. Gola, D. Roverso, M. Hoffmann, P. Baraldi, and E. Zio, “Large scale nuclear sensor monitoring and diagnostics by means of an ensemble of regression models based on Evolving Clustering Methods,” *PSAM10 10th Int. Probabilistic Saf. Assess. Manag. Seattle USA*, pp. 1–10, Jun. 2010.

[4] A. Vaswani *et al.*, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: Jun. 15, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[5] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, “A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction.” arXiv, Aug. 14, 2017. doi: 10.48550/arXiv.1704.02971.

[6] H. Zhou *et al.*, “Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting,” *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, Art. no. 12, May 2021, doi: 10.1609/aaai.v35i12.17325.

[7] R. Hübner, M. Steinhauser, and C. Lehle, “A dual-stage two-phase model of selective attention,” *Psychol. Rev.*, vol. 117, no. 3, pp. 759–784, 2010, doi: 10.1037/a0019471.

[8] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate.” arXiv, May 19, 2016. doi: 10.48550/arXiv.1409.0473.

[9] T. Lin, Y. Wang, X. Liu, and X. Qiu, “A Survey of Transformers.” arXiv, Jun. 15, 2021. doi: 10.48550/arXiv.2106.04554.