Deep Learning Approaches for the Detection of Ramping Artifacts in Schedule Deviation Data in Belgium

1st Henrike von Hülsen*
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Karlsruhe, Germany
h.vonhuelsen1@gmail.com

2nd Ulrich Oberhofer*
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Karlsruhe, Germany
ulrich.oberhofer@kit.edu

3rd Oliver Lauwers
Elia Group
Brussels, Belgium
oliverlauwers@cepsis.be

4th Gust Verbruggen
Microsoft
Brussels, Belgium
gustverbruggen@gmail.com

5th Sebastian Pütz
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Karlsruhe, Germany
sebastian.puetz@kit.edu

6th Veit Hagenmeyer
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Karlsruhe, Germany
veit.hagenmeyer@kit.edu

7th Benjamin Schäfer
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Karlsruhe, Germany
benjamin.schaefer@kit.edu

***Abstract*—In order to contribute to the goals set by the Paris Agreement to combat climate change, economies are increasingly incorporating renewable energy sources into their energy mix. As the share of renewables in the energy mix increases, so does the level of volatility and flexibility. This effect requires stakeholders to implement advanced methods to maintain grid stability and, by this, to balance the supply and demand of electricity in real time. Decisions on different balancing measures are supported by near real-time forecasts published by the responsible transmission system operator (TSO). If such control data is corrupted by ramping artifacts, this could induce unnecessary costs or threaten stability. Using operational data from the Belgian TSO Elia, we present a solution to detect these artifacts. Specifically, we use the knowledge of the internal structure of the artifacts to generate a synthetic dataset as a foundation for training a supervised deep learning model. The proposed model architecture contains an 1d-CNN feature extractor combined with transformer encoder blocks. This architecture is analyzed and compared to other supervised deep learning models and a classical unsupervised model, the Isolation Forest, in two training scenarios, a classification and a regression task. Testing on an unseen synthetic test set, we manage to outperform all other evaluated models, achieving an F_β-Score of 0.99 in the classification task and 0.75 in the masking task.**

***Index Terms*—deep learning, artificial intelligence, ramping artifact detection, imbalance price forecasts**

I. Introduction

The growing share of renewable energy sources in the energy mix introduces increased variability and uncertainty in the power supply [1]. Transmission System Operators (TSOs) manage the electricity system and balance demand and supply in real-time, in order to provide a reliable and affordable power supply [2]. To perform precise transactions on the electricity markets, market participants heavily rely on detailed forecasts [3], which therefore directly influence the stability of the power grid and the electricity prices [4].

Through the imbalance price, part of the balancing is diverted to the market, by incentivizing market participants to balance the grid and ensure cost-effective operation [5], [3]. To facilitate reactions to the imbalance price resulting in counteracting system imbalance, the Belgian TSO Elia publishes a minutely forecast of the estimated imbalance price, which is updated every quarter hour [6]. Reactions to the imbalance prices have to be incorporated into forecasts of the overall system imbalance, leading to a complex interconnected system of forecasts. In the process of generating these forecasts on imbalance prices and price reactions, Elia discovered artifacts in their data, caused by ramping times of assets, that currently diminish the forecast quality and therefore directly affect the system imbalance. More generally, any energy time series displaying incorrect data, or artifacts, is a potential problem for the balancing and control of energy systems. Hence, detecting, classifying and removing these artifacts from data is an important challenge for current and future power systems that rely increasingly on data [7]. Moreover, a key aspect of this process is to preserve the underlying signal information, since losing information from the clean signal would lead to a worse prediction. Thus, falsely detecting an artifact where none exists is incidentally worse than missing a few artifacts in the prediction.

While anomaly detection plays a major role in time series

We gratefully acknowledge funding from the Helmholtz Association and the Networking Fund through Helmholtz AI and under grant no. VH-NG-1727. We thank Matthias Hertel for stimulating discussions.

*: H.v.H. and U.O. contributed equally.

analysis, the task of specific artifact detection has not received as much attention and is mostly investigated in medical data, such as Electroencephalography (EEG) [8], [9], Electrocardiogram (ECG) [10] or functional Magnetic Resonance Imaging (fMRI) [11] data. However, anomaly detection in time series data is essential in a variety of applications, including finance [12], healthcare [13], [14], [8] and energy [15], [16] sectors. In general, one of the major challenges in anomaly detection is the scarcity of labeled data, as manual labeling of data is time consuming and expensive [17], [18], [19]. There are several methods to deal with artifact detection with limited labeled data. In some simpler cases, artifacts can also be removed directly by filtering [14], [20], but these methods tend to introduce new artifacts or undesirably affect the underlying signal [20]. As the signal caused by reactions to the imbalance price is similar to the ramping artifacts, these methods also pose the problem of filtering out necessary information. Therefore, we work with more sophisticated approaches and use one stochastic approach as a baseline.

The majority of previous work on anomaly detection utilizes an unsupervised approach, exploiting the assumption that artifacts classify as anomalous behavior and only occur in a vast minority of cases [21]. Alternatives include manually labeling data or creating synthetic training datasets to apply supervised learning techniques [16]. In the case of ramping artifacts, the underlying structure is known and can be formally defined. Hence, we leverage this additional knowledge to apply supervised learning methods to a synthetically generated training dataset.

The remainder of the article first discusses how to model and detect artifacts (Sec. II) before showing how our Transformer approach outperforms baselines and alternatives (Sec. III) and finally discussing our results in IV, in which the article is concluded.
The code for the results of the present paper is available on [22].

II. Methods

In the following, we work with an example dataset of a combined cycle gas turbine provided by Elia, which contains schedule deviation, schedule, generation and manual Frequency Restoration Reserve (mFRR) activation for each minute in one year. To ensure confidentiality, all data was normalized to a mean of 0 and standard deviation of 1. We therefore use arbitrary units on the y-axis and refer to time steps on the x-axis, which in this case are in minutes.

A. Modeling and generating ramping artifacts

Both, reactions to imbalance prices and ramping artifacts can be found in schedule deviation data, which is the difference between scheduled generation or consumption and actual generation or consumption plus control power requested by the TSO. A scheme showing the composition of the scheduled generation data is presented in [22].

Figure 1 right panel shows an idealized isolated artifact, which shows the following characteristic properties:

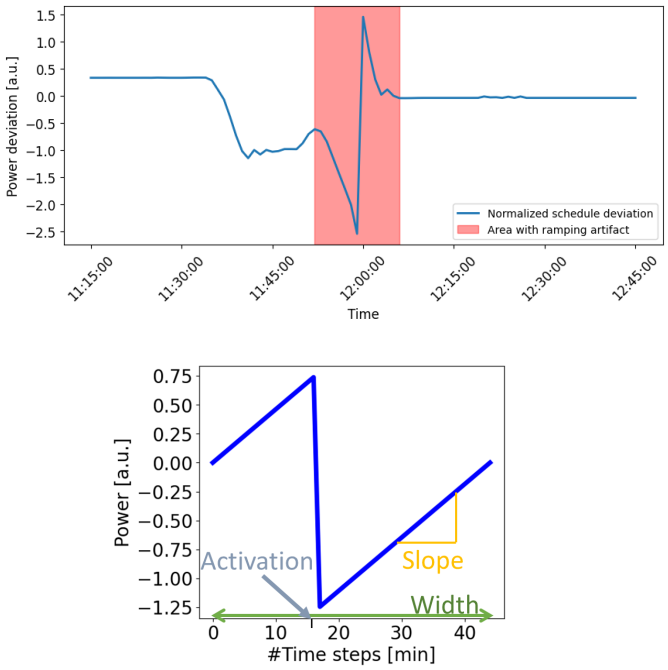

Fig. 1: Left: The schedule deviation (blue) of an example asset displays a ramping artifact (red shade). We note that the discontinuity characteristically takes place at a full quarter hour. Right: A generalized model of an isolated ramping artifact (blue) and its parameters are shown. Artifacts differ in width (green), slope (yellow), and position of the activation (gray), meaning the position of the discontinuity.

- A steep jump (discontinuity) at the point where the new schedule takes effect.
- The same slope before and after the discontinuity.
- Start and endpoint share the same y-value, the difference in x marks the width of the artifact.

In order to determine the possible variable spans, we have identified possible 123 artifacts in the schedule deviation of the provided industry dataset. Due to limited information and manual labeling, they can merely provide an approximate range of the variable assignment and do not represent a validated ground truth. We define the three parameters characterizing a ramping artifact as follows:

$$\text{width} = x_{\text{end}} - x_{\text{start}}, \quad \text{slope} = \frac{y_{\text{ref}} - y_{\text{start}}}{x_{\text{ref}} - x_{\text{start}}}, \quad x_{\text{relAct}} = \frac{x_{\text{act}} - x_{\text{start}}}{x_{\text{end}} - x_{\text{start}}}.$$

Here $(x_{\text{start}}, y_{\text{start}})$ marks the start point of the artifact, $(x_{\text{end}}, y_{\text{end}})$ the endpoint, $(x_{\text{act}}, y_{\text{act}})$ the point of the discontinuity and $(x_{\text{ref}}, y_{\text{ref}})$ one other reference point. In x_{relAct} the relative position of the activation is calculated.

Using these properties we exploit the known structure of the ramping artifact to generate synthetic ramping artifacts and introduce them to several open-source datasets, which originally do not contain artifacts of this shape. The algorithm for generating process synthetic artifacts can be found in [22].

For the synthetic training dataset we use a collection of different open source datasets. As we calculated all properties of

the artifacts in general time steps, we can use any open-source dataset independent from the original heritage of their data. The datasets vary in data type, source, number of time series, time series length and sampling interval. Four energy related datasets were provided by Monash University [23], [24], [25], [26], two other energy related datasets are part of the UCI ML repository [27], [28] and a last energy related dataset was taken from Kaggle [29]. Ten datasets were taken from the UCR Time Series Classification Database [30], two are ECG data [31] and another two are transformer temperature data [32]. The UCR Time Series Classification archive is renowned in the time series community and widely used for time series analysis tasks. At Elia, ramping artifacts were observed for different types of assets. Throughout different asset data, the structural information of the artifacts remain the same due to their physical properties, while the size and structure of the underlying time series data vary. The goal is therefore to detect the artifacts based on the structural information only, generalizing to a variety of data. In addition, since we train the proposed model on other data than the target data, we want to analyze how variety in the underlying data (modeled through utilizing a wide range of datasets) contributes to the model performance, when tested on an unseen dataset structure.

B. Detecting artifacts

To detect ramping artifacts we pursue two approaches:

1) Mask Approach: The mask approach solves the problem as a regression task and will predict a value for each point of each input time series between 0 and 1. The closer the value is to 1, the more confident the model is that this point belongs to an artifact. For this approach, for each input time series, the model will predict a mask of the same size.
2) Sliding Window Approach: This approach solves the problem as a classification task and will predict a single label for each window, 0 meaning no artifact present and 1 predicting an artifact in the center.

In this second case, the sliding window approach, we want to leverage the given additional information about the position of artifacts. Since we know that artifacts can only occur during schedule changes, we determine positions possibly containing artifacts by finding all positions at which a schedule change or a change in mFRR activation occurs. These positions always mark the position of the discontinuity of the artifact. Detecting artifacts in the target data can therefore be performed by solely evaluating these possible artifact positions. We mimic this behavior with a time series classification task, for which models are trained to produce a single output per input time series, predicting whether the window contains an artifact. For the generation of the synthetic dataset, we place the artifacts in the center of the sequence. Figure 2 shows two examples of generated training samples. The ground truth will be a mask in the mask approach and a single bit in the sliding window approach.

Because we do not have reliably labeled test data from the target dataset, we determine one dataset of the training datasets to exclude from training and use that dataset for testing purposes only. This mimics the anticipated workflow of training on synthetic data, in order to then apply the model on an unseen target dataset. As a test dataset we choose the dataset most similar to the provided industry dataset. To measure similarity, we compute chosen time series measures per time series in a dataset and average over all time series in one dataset. This yields one value per measure per dataset. Afterwards, we calculate the absolute distance between each value of each dataset to the respective value of the industry dataset. In this way, we are able to rank the dataset per measure from 1 (most similar to the industry data) to 21 (least similar to the industry data). The chosen measures include common measures of the data distribution and the increment distribution of the datasets. The dataset that is the most similar with respect to the industry data is then selected as test dataset while the 20 remaining sets are chosen as training sets.

For generating a labeled dataset, we use an iterative approach, which means that we generate the training time series step by step as the model is training. In each step of the training process, one of the 20 training datasets is chosen. Following that, one single time series is chosen randomly from that specific dataset and in this time series, a window of a previously decided fixed window length is chosen. In case the activity in the window is below a certain threshold, the window is discarded and a new window is chosen. Once a suitable window has been sampled, an artifact is introduced.

Utilizing the introduced synthetic artifacts and the open source datasets, we propose a model that combines a 1D Convolutional Neural Network (CNN) with transformer encoder blocks inspired by [19], see [22] for details. The hypothesis is that the 1D CNN block extracts local spatial features from the time series that are then fed into the transformer encoder, which can extract long-range dependencies in the time series to decide whether an artifact is present or not.

C. Loss functions and performance metrics

For the mask approach we choose the mean squared error loss, for the sliding window approach the binary cross entropy loss. The dataset for the mask approach is highly imbalanced, due to the fact that all positions not containing an artifact are labeled as negative. In addition, Elia's industry requirement includes that false positives have a higher negative impact than false negatives. Following the detection of the artifacts, the next step will be to remove the detected artifacts. For this it is not desirable to remove valid signal in places where a false artifact is detected, while the negative effects of missing some artifacts in the prediction are smaller. To compare and evaluate our models we therefore choose the F_β-score, which combines precision and recall into one metric, allowing to give more weight to one of them through adjusting the β parameter. Since we want to penalize a high number of false positives we choose $\beta = 0.5$, which gives more weight to precision.

In order to reach the best possible F_β-Score through the training process, we can choose a loss function that will contribute to improving the F_β-Score. Since the F_β-Score itself is not differentiable, it is unsuitable for gradient-based

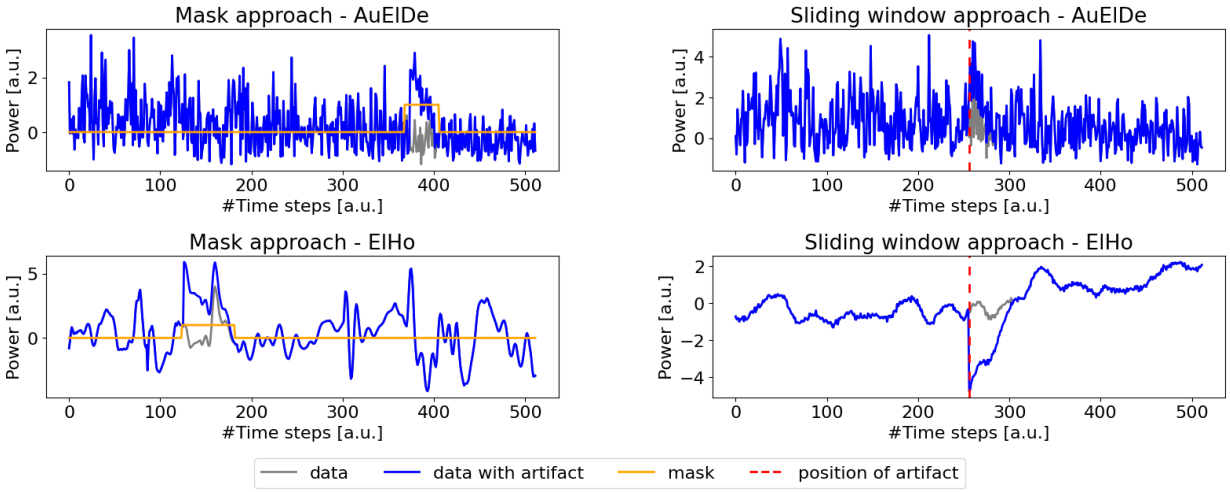

Fig. 2: Illustration of synthetic time series for the supervised learning approach: We plot the original data (gray) and compare it with an added generated artifact (blue). Depending on the randomly sampled parameters of the artifact, the size varies and can be relatively big and easily identifiable (top right) or rather small and more difficult to identify (bottom right). The top row uses the Australian electricity demand dataset (AuEIDe) [23], and the bottom row uses the electricity hourly dataset (ElHo) [24]. The left column examples are generated for the mask approach, with the ground truth in the form of a mask. The plots on the right side show examples generated for the sliding window approach, where the artifacts are always placed with the discontinuity in the center.

optimization methods [33]. There are approaches that introduce an adapted differentiable version of the F_β-Score as a loss function [33], however, we decide to implement a simpler loss function, penalizing a high number of false positives. For this reason, we introduce an additional parameter ϵ that adds a penalty for a high loss on negatively labeled samples. The new loss with the respective loss function of the mask or the sliding window approach is composed of the classic loss function plus the penalty term:

$$\text{loss}_\epsilon(l, y) = \text{loss}(l, y) + \epsilon \cdot \text{loss}(l_I, y_I),$$

where l describes the actual label and y the prediction of the according model. Additionally, the arrays l_I, y_I consist of exactly the samples i with label $l_i = 0$.

D. Model selection

In order to determine whether transformer encoder blocks add value to a model detecting ramping artifacts, we evaluate the performance of the 1D convolution block combined with the output layers only, and of the transformer blocks plus the output layers without feature extraction through a CNN. Since one requirement of the transformer architecture is that the embedding dimension must be divisible by the number of attention heads [19], we do need to choose some form of embedding even though we want to evaluate the sole performance of the encoder blocks as good as possible. For this we follow an approach called ConvTran [19] using only one 1D convolutional layer to perform a feature extraction as input embedding, choosing the embedding dimension as four times the number of attention heads. To evaluate the contribution of the transformer blocks further, we also evaluate our proposed model composed of a 1D CNN plus transformer encoder blocks against a model consisting of the 1D CNN plus a bigger dense block with approximately the same number of parameters as the transformer blocks have.

The proposed solutions are benchmarked against two baseline approaches. The first baseline approach is a very simple stochastic-based detector. It is known that the artifacts always contain a discontinuity, so at least one position with an especially high increment. Thus we define our baseline detector in terms of the sliding window approach to classify windows as containing an artifact, in case the absolute increment in the center of the artifact is higher than the mean increment plus k times the standard deviation, where we choose $k = 3$. For details of the implementation and the choice of k, see [22].

Furthermore, we compare our approaches to the fully convolutional network (FCN) by Wang et al. [34] claiming the FCN to be a simple but strong baseline for time series classification. They evaluated their architecture on the UCR TSC archive [30], which is also the source of 10 of our training datasets. The FCN corresponds to the first block of our architecture of 1D convolutions with batch normalization and a ReLU activation function after each layer, followed by a Global Average Pooling (GAP) and a softmax activation, which we exchange with a sigmoid activation, as we are only differentiating between two classes.

To understand the effect of the FCN configuration on the result, we also use an adapted version of the FCN with a different number of layers. Finally, as we will see in Section III and in Table I, we use different combinations of FCN and transformer blocks, therefore, we consider the following deep learning models: a FCN and an adapted FCN, a Transformer

network with both a GAP and a linear layer as output layer (Transformer GAP and Transformer Lin.), the FCN and the adapted FCN with a linear output layer (FCN Dense and adaFCN Dense), and the FCN and the adapted FCN combined with the Transformer network (FCN Trans. and adaFCN Trans.).

We are convinced that transforming the artifact detection problem into a supervised learning task by exploiting the structural information about the artifacts brings significant value. To assess this assumption, we also compute the performance of an Isolation Forest [35], as a basic unsupervised approach and alternative to our supervised approach.

III. Results

To test our algorithm on unseen data, we exclude the dataset from training that is most similar to the training data to approximate the typical i.i.d. assumption. In our collection of datasets this is the CiECGT dataset from the UCR TSC archive [30] which consists of ECG sensor data of the torso surface. All remaining datasets are split, using 80 % of the included time series for training and 20 % for creating a validation dataset. The validation dataset of a fixed size (2048 samples) is created once and stays the same during the training process. We choose the validation set to be composed of a fraction of the datasets, rather than selecting one specific dataset as the validation dataset because we do not want the model to adapt to a separately chosen single validation dataset. The classification threshold is determined by maximizing the F_β-Score on the validation set.

Since there are no epochs that define when to do the validation step, a parameter is set to define after how many steps the validation step will be performed. Per default, we set this parameter to 500 steps. The sliding window approach gives one prediction per sample on the synthetic balanced test dataset with 2048 samples, so the prediction ratios in Table I are based on 2048 predictions.
The deep learning models are trained on NVIDIA Tesla V100-GPUs. Table I shows the results of evaluating all models in the synthetic dataset chosen for testing. Since the mask approach predicts one value for each point in each time series (of length 512 time steps) of the test dataset (with 2048 samples), the overall number of predictions is very large ($2048 \times 512 > 10^6$). To increase readability and comparability between the two test sets, the prediction measures of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) are shown as ratios of the total number of predictions.

Starting with the stochastic baseline approach, we see in Table I that it can point out some of the positive samples, however, it predicts more FP than TP and shows a low F_β-Score. We see that the adapted FCN model performs slightly better on all measures than the original FCN model presented in [34]. When determining the more suitable output layer for the transformer models without an elaborate input embedding, we see that the linear output layer performs significantly better, though it has to be noted that this version also trains on (few) more parameters. The original and adapted FCN architectures, which are flattened over the feature and time dimension and use a dense block as the output layer, do not learn much in the mask approach and perform the worst.

Although the adapted FCN outperforms the original FCN model, when adding transformer encoder blocks, the original FCN plus transformer encoder yields the best results in the mask approach when tested on the synthetic test dataset, especially regarding F_β-Score and the ratio of FP. Figure 3 shows the predictions before applying the classification threshold of the FCN transformer model, as well as the predictions of its isolated components: the original FCN model and the transformer block with a linear output layer. Within the figure, Figure 3a shows the sample with the least FP accumulated overall model predictions on this sample, Figure 3b the sample with the most FP. Analogously, Figure 3c and Figure 3d show the samples with the least and most FN predictions.

We can see from these examples that an artifact with comparably high values for slope and width as in Figure 3a is confidently detected by all three visualized models, while the two transformer models are more precise. Also the artifact with a high slope, but small width in Figure 3c is detected by all models. The artifact with small width and small slope in Figure 3b is not detected at all and Figure 3b and Figure 3d show that the models tend to predict a non-existing artifact in the window, even though they completely miss the actual artifact. This indicates that the share of windows without an artifact during training of the mask approach is currently too low, or the models have learned a bias towards predicting artifacts otherwise. This behavior is undesirable, especially taking into consideration the higher penalty for FP. Figure 3d shows another artifact that is missed by all models. The artifact presents a high value for width, and a smaller value for slope and is introduced in an area with high activity. We note that in this case, only the two transformer based models predict an artifact at a different position. The transformer model without FCN block as input embedding (labeled Trans. in the legend) predicts the false artifact confidently, while the FCN transformer model (labeled FCN Trans.) does not exceed its classification threshold of 0.687 (Table I). In this case, the transformer model benefits from the more extensive input embedding using a fully convolutional network.

In the sliding window approach, the stochastic baseline manages to detect artifacts quite well only producing 1 % FP. The construction of the synthetic dataset benefits the stochastic baseline approach since samples without an artifact in the center will just be random samples of the underlying dataset, which makes it unusual to show a disproportionately high increment in the center of the window.

The Isolation Forest does perform better than random guessing, but does not manage to outperform any other model in this section. Again, the adapted FCN outperforms the original FCN, and the transformer encoder with a linear output layer performs better than the transformer encoder with Global Average Pooling output, although less clearly than in the mask approach. As opposed to training on the mask approach, both versions of the FCN with a dense output block perform well

Model	Threshold	F_β-Score	Accuracy	Precision	Recall	TP	FP	FN	TN
				Mask Approach					
Stoch. Baseline	-	0.162	0.938	0.734	0.078	0.2 %	0.5 %	5.7 %	93.6 %
Isolation Forest	-	0.370	0.937	0.438	0.229	1.3 %	1.7 %	4.5 %	92.4 %
FCN	0.505	0.499	0.941	0.535	0.561	3.8 %	3.7 %	2.2 %	90.3 %
adapted FCN	0.515	0.577	0.967	0.578	0.630	4.4 %	1.7 %	1.6 %	92.3 %
transformer GAP	0.596	0.534	0.943	0.576	0.555	3.5 %	3.3 %	2.4 %	90.8 %
transformer Lin.	0.758	0.745	**0.985**	0.769	0.719	**4.8** %	0.3 %	**1.2** %	**93.8** %
FCN Dense	0.636	0.053	0.906	0.063	0.041	0.2 %	3.7 %	5.7 %	90.4 %
adaFCN Dense	0.626	0.059	0.914	0.074	0.039	0.2 %	2.9 %	5.7 %	91.1 %
FCN Trans.	0.687	**0.750**	**0.985**	**0.780**	**0.724**	4.7 %	**0.2** %	1.3 %	**93.8** %
adaFCN Trans.	0.545	0.577	0.967	0.578	0.630	3.7 %	1.6 %	2.3 %	92.5 %
				Sliding Window Approach					
Stoch. Baseline	-	0.974	0.967	0.980	0.952	47.1 %	1.0 %	2.3 %	49.6 %
Isolation Forest	-	0.541	0.483	0.472	0.491	20.9 %	23.4 %	21.7 %	33.9 %
FCN	0.525	0.700	0.712	0.681	0.783	38.7 %	22.9 %	10.7 %	32.5 %
adapted FCN	0.515	0.889	0.860	0.918	0.790	38.7 %	3.5 %	10.5 %	47.1 %
transformer GAP	0.333	0.990	0.980	0.997	0.961	47.5 %	0.1 %	1.9 %	40.4 %
transformer Lin.	0.505	0.991	0.980	0.999	0.959	47.4 %	0.0 %	2.0 %	50.5 %
FCN Dense	0.343	0.990	0.980	0.998	0.961	47.5 %	0.1 %	1.9 %	50.4 %
adaFCN Dense	0.373	0.981	0.967	0.992	0.942	46.5 %	3.9 %	2.9 %	50.2 %
FCN Trans.	0.252	**0.993**	**0.984**	**1.00**	0.968	**47.8** %	**0** %	**1.5** %	**50.6** %
adaFCN Trans.	0.343	0.982	0.977	0.987	**0.965**	47.7 %	0.6 %	1.7 %	49.9 %

TABLE I: Selected performance measures and the classification threshold of the chosen models are reported on a synthetic test set (not previously seen during training). All models are trained on the mask and the sliding window approach. The last four columns contain the ratio of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) compared to the total number of test samples. Bold values mark the best performance in the respective column separately for the mask and the sliding window approach.

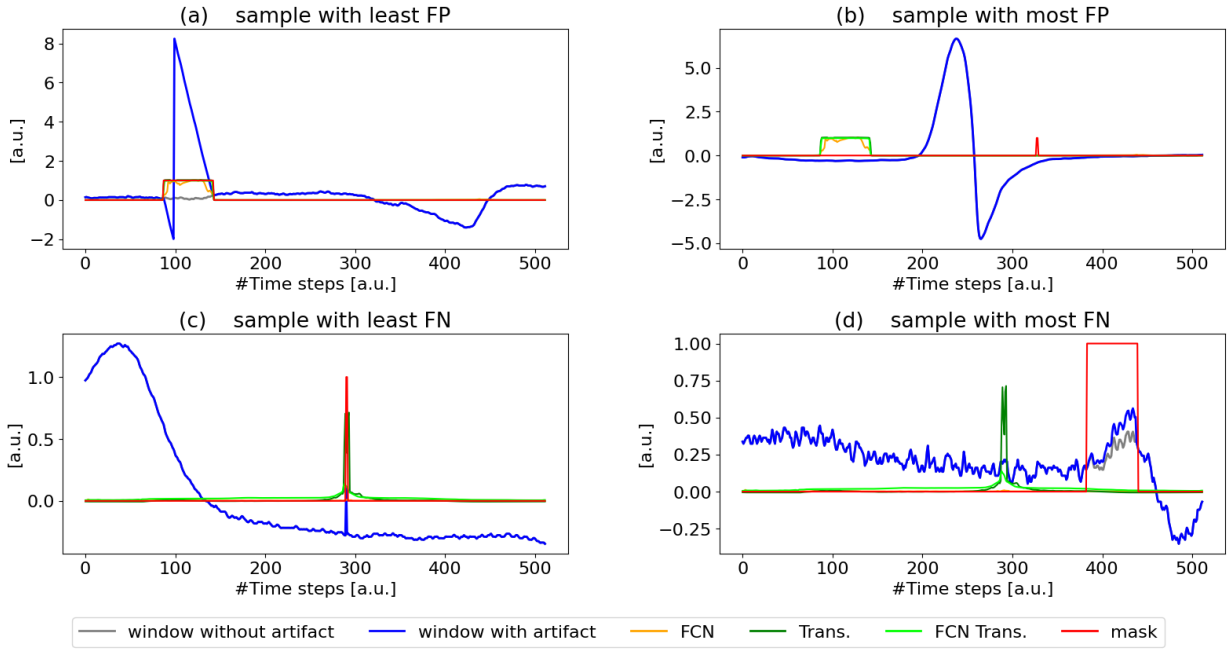

Fig. 3: Samples of the synthetically generated test dataset are plotted with the predictions by the FCN model (yellow), the transformer with minimal input embedding and linear output layer (Trans. in green) and the FCN transformer model (FCN Trans. in lime green) in the mask approach. The legend applies to all images. All graphs show the underlying time series window in gray and the time series with an added artifact in blue. The ground truth mask is shown in red. The top row shows the samples with the least and most FP, while the bottom row shows the samples with the least and most FN.

with an F_β-Score of 0.990 and 0.981 for original and adapted FCN, respectively. Similar to training on the mask approach, the original FCN with added transformer encoder blocks performs slightly better than the adapted version, even though

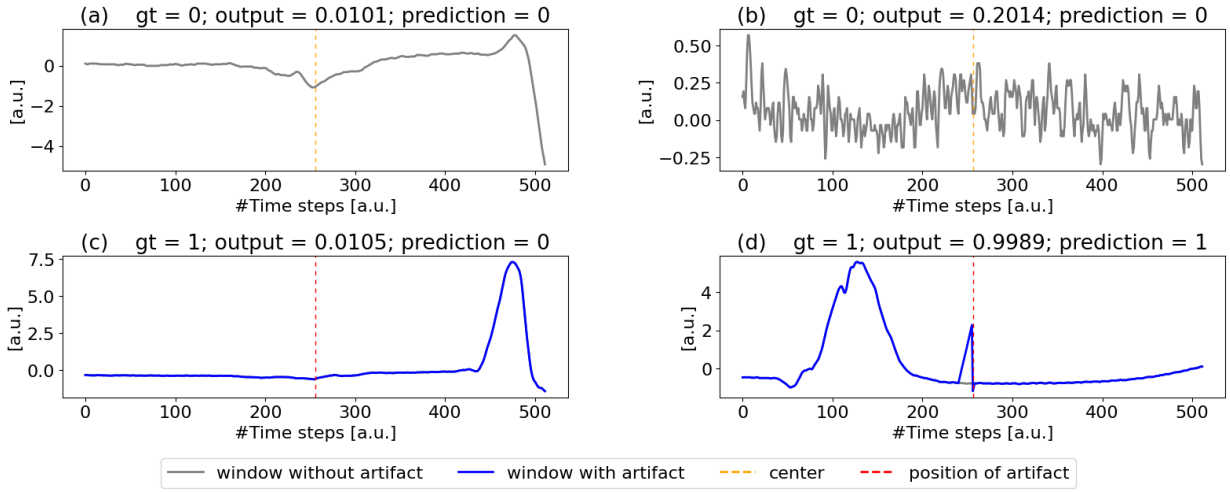

Fig. 4: Samples of the synthetically generated test dataset are classified by the FCN transformer model in the sliding window approach. All graphs show the underlying time series window in gray. The top row, (a) and (b), shows two negative samples without an added artifact. The center of the sequence, where a possible artifact would have been added is shown with a dashed orange vertical line. The bottom row, (c) and (d), shows positive samples with the time series with the added artifact shown in blue. The position of the artifact is shown with a dashed red vertical line in the center of the image. On the left side, in (a) and (c), the two samples with the lowest prediction score of the respective category are chosen, and on the right side, in (b) and (d), the samples with the highest prediction value.

the individual models without transformer blocks showed the opposite behavior. The best model, the FCN with transformer encoders, does not predict any FP.

Figure 4 shows four exemplary sequences of the synthetic test dataset, classified by the best sliding window model on the synthetic test data, the FCN plus transformer encoder model. The figure shows the sample without an artifact with the lowest prediction score in Figure 4a, the sample without an artifact with the highest prediction score in Figure 4b, the sample with an artifact with the lowest prediction score in Figure 4c and the sample with an artifact with the highest prediction score in Figure 4d. In Figure 4b we see that the highest prediction of a negative sample is 0.2, which shows that the prevention of FP is successful in the whole model, as no negatively labeled sample reaches a prediction score higher than 0.2. Meanwhile, it still yields 1.5 % FN (see Table I).

The artifact introduced in Fig. 4c is not detected and is in this graph invisibly small. We note that the FCN transformer approach can prevent a large number of FP on the synthetic test set, while predicting a still reasonable amount of FN, mainly struggling with small artifacts.

IV. Discussion and Conclusion

Artifacts (or other erroneous measurements) in power system time series pose a threat to system stability. Hence, we set out to identify and remove artifacts, utilizing empirical data from the Belgian TSO Elia.

Due to the lack of labeled real-life data, we exploited the known structure of the ramping artifacts to train supervised deep learning models on synthetically generated training data. In addition to a manually labeled industry dataset of a combined cycle gas turbine, we used a separate synthetic dataset for evaluation. We have introduced an architecture composed of a 1D convolutional feature extraction which was then passed as an input embedding to transformer encoder blocks. This architecture was compared to a stochastic baseline, a classically used unsupervised model, the Isolation Forest, and other supervised learning models. The supervised learning models included the separated components of the proposed methods, as well as the convolutional block with an added dense block of approximately the same count of trainable parameters.

Our results demonstrate that the sliding window approach outperforms the mask approach. We might explain this based on the difficulty of the respective task. Classifying a whole window is a simpler task compared to extracting the exact position of an artifact or possibly several artifacts in a sequence. In addition, the imbalance of the data set in the mask approach also has an influence on the performance. This approach is not limited to ramping artifacts but could also prove useful for artifact detection tasks in other (power system) time series. The main assumption is that domain knowledge of the underlying artifacts allows us to synthetically generate training samples for the supervised learning problem formulation. We have assumed all artifacts in the target datasets to occur with a time resolution of one minute. As most data that the TSOs collect and publish is on a minute resolution, it is likely that artifacts occurring in other data sources will also show a time interval of one minute.

Concluding, artifact detection with deep learning, specifically with convolutional and transformer approach is feasible and could contribute to a more secure and stable power grid.

With this promising approach established, there is much room for further research and improvement. First, we would like to include more training data, in particular from the realistic power system application. Second, the machine learning implementation could move beyond the vanilla transformer block [36], e.g. implementing a customized attention layer [37] or considering LSTM encoding [38]. In addition, the sliding window and mask approach could also be combined in the future to exploit the advantages of both models. Alternatively, restructuring the mask approach into a classification task and thus also using a different loss function, such as the cross-entropy loss, could improve the model performance. Finally, generalization to other data sets might be improved by including more domain knowledge, e.g. via physics-informed machine learning models [39].

References

[1] S. R. Sinsel, R. L. Riemke, and V. H. Hoffmann, "Challenges and solution technologies for the integration of variable renewable energy sources—a review," *Renewable Energy*, vol. 145, pp. 2271–2285, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148119309875

[2] J. Machowski, Z. Lubosny, J. W. Bialek, and J. R. Bumby, *Power system dynamics: stability and control*. John Wiley & Sons, 2020.

[3] "Keeping the balance," https://www.elia.be/en/electricity-market-and-system/system-services/keeping-the-balance, accessed: 13.05.2024.

[4] C. Koch and L. Hirth, "Short-term electricity trading for system balancing: An empirical analysis of the role of intraday trading in balancing germany's electricity system," *Renewable and Sustainable Energy Reviews*, vol. 113, p. 109275, 2019.

[5] "About elia," https://www.elia.be/en/company, accessed: 13.05.2024.

[6] "Elia open data platform," https://opendata.elia.be/pages/home/, accessed: 13.05.2024.

[7] Y. Zhang, X. Shi, H. Zhang, Y. Cao, and V. Terzija, "Review on deep learning applications in frequency analysis and control of modern power system," *International Journal of Electrical Power & Energy Systems*, vol. 136, p. 107744, 2022.

[8] D. Gorjan, K. Gramann, K. De Pauw, and U. Marusic, "Removal of movement-induced eeg artifacts: current state of the art and guidelines," *Journal of neural engineering*, vol. 19, no. 1, p. 011004, 2022.

[9] İ. Yıldız Potter, G. Zerveas, C. Eickhoff, and D. Duncan, "Unsupervised multivariate time-series transformers for seizure identification on eeg," *arXiv e-prints*, pp. arXiv–2301, 2023.

[10] S. Chauhan and L. Vig, "Anomaly detection in ecg time signals via deep long short-term memory networks," in *2015 IEEE international conference on data science and advanced analytics (DSAA)*. IEEE, 2015, pp. 1–7.

[11] R. H. Pruim, M. Mennes, D. van Rooij, A. Llera, J. K. Buitelaar, and C. F. Beckmann, "Ica-aroma: A robust ica-based strategy for removing motion artifacts from fmri data," *Neuroimage*, vol. 112, pp. 267–277, 2015.

[12] W. Hilal, S. A. Gadsden, and J. Yawney, "Financial fraud: a review of anomaly detection techniques and recent advances," *Expert systems With applications*, vol. 193, p. 116429, 2022.

[13] W. Mumtaz, S. Rasheed, and A. Irfan, "Review of challenges associated with the eeg artifact removal methods," *Biomedical Signal Processing and Control*, vol. 68, p. 102741, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1746809421003384

[14] K. T. Sweeney, T. E. Ward, and S. F. McLoone, "Artifact removal in physiological signals—practices and possibilities," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 3, pp. 488–500, 2012.

[15] M. Turowski, B. Heidrich, K. Phipps, K. Schmieder, O. Neumann, R. Mikut, and V. Hagenmeyer, "Enhancing anomaly detection methods for energy time series using latent space data representations," in *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, 2022, pp. 208–227.

[16] M. Turowski, M. Weber, O. Neumann, B. Heidrich, K. Phipps, H. K. Çakmak, R. Mikut, and V. Hagenmeyer, "Modeling and generating synthetic anomalies for energy and power time series," in *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, 2022, pp. 471–484.

[17] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: a comprehensive evaluation," *Proceedings of the VLDB Endowment*, vol. 15, no. 9, pp. 1779–1797, 2022.

[18] H. I. Fawaz, "Deep learning for time series classification," *arXiv preprint arXiv:2010.00567*, 2020.

[19] N. M. Foumani, C. W. Tan, G. I. Webb, and M. Salehi, "Improving position encoding of transformers for multivariate time series classification," *Data Mining and Knowledge Discovery*, vol. 38, no. 1, pp. 22–48, 2024.

[20] J. van Driel, C. N. Olivers, and J. J. Fahrenfort, "High-pass filtering artifacts in multivariate classification of neural time series data," *Journal of Neuroscience Methods*, vol. 352, p. 109080, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165027021000157

[21] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM computing surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[22] H. von Huelsen, "Artifactory," https://github.com/hvonhue/Artifactory, 2024.

[23] R. Godahewa, C. Bergmeir, G. Webb, R. Hyndman, and P. Montero-Manso, "Australian electricity demand dataset," Apr. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4659727

[24] ——, "Electricity hourly dataset," Apr. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4656140

[25] ——, "Solar dataset (10 minutes observations)," Apr. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4656144

[26] R. Godahewa, C. Bergmeir, G. Webb, M. Abolghasemi, R. Hyndman, and P. Montero-Manso, "Wind farms dataset (without missing values)," Mar. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4654858

[27] A. Trindade, "ElectricityLoadDiagrams20112014," UCI Machine Learning Repository, 2015, DOI: https://doi.org/10.24432/C58C86.

[28] C. W. Tan, C. Bergmeir, F. Petitjean, and G. I. Webb, "Household active power consumption dataset," Jun. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3902704

[29] "Smart meters in london," https://www.kaggle.com/datasets/jeanmidev/smart-meters-in-london, accessed: 20.05.2024.

[30] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, and Hexagon-ML, "The ucr time series classification archive," October 2018, https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

[31] "Ecg heartbeat categorization dataset," https://www.kaggle.com/datasets/shayanfazeli/heartbeat, accessed: 20.05.2024.

[32] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[33] N. Lee, H. Yang, and H. Yoo, "A surrogate loss function for optimization of f beta score in binary classification with imbalanced data," *arXiv preprint arXiv:2104.01459*, 2021.

[34] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 1578–1585.

[35] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[37] Y. Liu, G. Sun, Y. Qiu, L. Zhang, A. Chhatkuli, and L. Van Gool, "Transformer in convolutional neural networks," *arXiv preprint arXiv:2106.03180*, vol. 3, 2021.

[38] S. Pütz, H. El Ashhab, M. Hertel, R. Mikut, M. Götz, V. Hagenmeyer, and B. Schäfer, "Feasibility of forecasting highly resolved power grid frequency utilizing temporal fusion transformers," in *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*, 2024, pp. 447–453.

[39] J. Kruse, E. Cramer, B. Schäfer, and D. Witthaut, "Physics-informed machine learning for power grid frequency modeling," *PRX energy*, vol. 2, no. 4, p. 043003, 2023.