Detection of Misconfigurations in Power Distribution Grids using Deep Learning

David Fellner*, Thomas I. Strasser*†, Wolfgang Kastner†

*AIT Austrian Institute of Technology, Vienna, Austria {david.fellner, thomas.strasser}@ait.ac.at
†TU Wien, Vienna, Austria {thomas.strasser, wolfgang.kastner}@tuwien.ac.at

***Abstract*—The electrical energy system is undergoing major changes due to the necessity for more sustainable energy generation and the following increased integration of novel grid connected devices, such as inverters. To operate reliably in novel circumstances, as created by the decentralization of generation, power systems usually need grid supportive functions provided by these devices. These include control mechanisms such as reactive power dispatch used for voltage control. In this work, an approach for the detection of misconfigured (wrongly parameterized control curve, etc.) grid devices using operational data is proposed. By generating and analysing operational data of power distribution grids, a Deep Learning approach is applied to the detection problem given. An end to end framework is used to synthesize and process the data as well as to apply the machine learning techniques on it. The results offer insights into applicability and possible ways to improve the proposed solution and how it could be employed by grid operators.**

***Index Terms*—Power distribution, deep learning, device malfunctions, operational data.**

I. Introduction

Today, especially power distribution system operators (DSO) have to cope with new challenges arising due to the transformation of the energy system. A major shift in paradigm is the increasing penetration of decentralized power generation [1], which leads to technical challenges in transmission and storage of power. Standing out is the impact of high photovoltaics (PV) proliferation; in case of generation outdoing demand locally, bidirectional power flows between voltage levels as well as voltage rises are the consequences [2]. If the voltage is lifted too much this can lead to voltage band violations, which consist of voltages above or below the admissible limits. Control mechanisms are employed to allow for a reasonable decentralized generation of renewable energy without creating said violations. For this purpose, voltage regulation is the preferred strategy [3], which is made possible by generation units implementing grid supporting functions. These target the frequency as well as the voltage amongst others. Apart from limiting the dispatch of active power, the most common way to influence the voltage is via the power factor and followingly the reactive power exchanged with the network, usually controlled by a local droop control [4].

To ensure that these grid supporting functions are actually delivered, DSOs need to monitor the operation of grid connected devices, for instance PV inverters, as to be sure that the network works in a stable manner. As the available information about grid components characteristics is often limited, a data driven approach is a favourable option [5] for a monitoring solution that is actually useful to DSOs. Such a solution can be crafted in a way as to only use operational data of the grid connected devices, in order to detect misconfigurations of the same. These deviations of configurations from the specifications – as defined by grid codes – can have two reasons: firstly, a different configuration than the normative one can be purposely implemented. Secondly, the configuration can change due to malfunctions or faults. Here misconfiguration stand for the latter meaning a deviation from a previous implementation of i.e. a control curve which is assumed to be initially correct. Figure 1 depicts how these terms are linked and what is needed to detect anomalies with respect to the type of anomaly. It becomes obvious that for the detection of involuntary misconfigurations only a detection of the execution of functionalities is necessary, which does not require knowledge about an implementation code or the fundamental specification. Therefore, only operational data is used for this purpose.

The main contribution of this work is the detailed description of an end to end framework that can be used to handle grid operational data and to detect misconfigurations. First, this framework is employed to either select or generate, clean and label data for further use. Then, various detection mechanisms can be applied on the data, which lastly are evaluated and compared against each other. In this work, Deep Learning (DL) approaches are under scrutiny. They are chosen because of voltage curves being highly non-linear and features can not be easily derived from them. Yet, our previous work indicates a detectable impact of misconfigurations on the voltage [6], this makes DL an interesting approach [7].

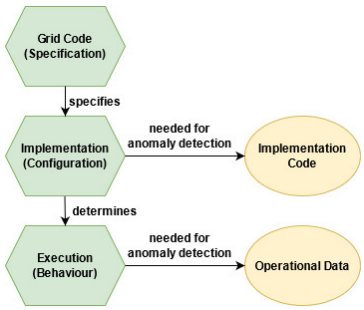

Fig. 1: Definitions of terms and requirements for the detection of wrong implementations (code needed) respective misconfigurations (data needed).

This work has the following content: In Section I a detailed discussion of monitoring needs and issues in power distribution grids is conducted. Section II describes the state-of-the-art related to malfunctions in power system as well as the usage of artificial intelligence for detecting them. In Section III, the functionality and implementation details of the detection framework are lined out and in Section IV a description and results of approaches explored using the framework are presented. Finally, Section V provides the conclusions and an outlook about potential further work.

II. Related Work

In the work of [8], electricity consumption data is modeled using a combination of polynomial regression and Gaussian distribution. This is done to detect anomalies in the electricity demand of several schools. This approach could be used for anomaly detection of grid connected devices, however, the models have to be fitted individually for each device making the application less suitable for broad usage.

In [9], consumption patterns of medium voltage transformers at substations are clustered using algorithms, such as k-means and fuzzy c-means. Abnormal consumption is then identified employing the local outlier factor (LOF) of hourly load data as a measure. Indicators such as irregular peak unusual consumption, broadest peak demand, sudden large gain and nearly zero demand unusual consumption are used as features here. Even if not applicable to this particular problem, this shows that there are features present that allow for general detection of anomalous behaviour from operational data.

This can be exploited by using DL. As elaborated in [10], Recurrent Neural Networks (RNN) can be used to classify time series data: an Elman network structure is applied to classify a time series. This includes a feed-forward part and a memory part which feeds network activation's from a previous time step as inputs to the network to influence predictions at the current time step. This is achieved through back propagation through time (BPTT); here the gradient of the cost function is propagated with regard to the parameters of the network, like weight matrices, for every time point of the sequence and each layer by unfolding the recurrent connections through time [11]. The parameters are updated using the gradient in a way that minimizes the cost function. The cost function is selected according to the task, such as classification or regression. Processing the input as a sequence adds a temporal dimension to the information gained and allows a more flexible window of information to be used in contrast to a feed-forward network. Here the most frequent classification result yielded by the output neurons is used as a classification result. This might be feasible for grammar checking but might need alteration for the problem addressed in the work here.

The RNN approach nevertheless has some deficiencies, most prominently its lacking ability to capture long-term dependencies in the sequential data, as lined out in [12]. In the Long Short-Term Memory (LSTM) RNNs recurrent hidden layer, so called 'memory blocks' are contained; they are made of memory cells that store the networks temporal state using self-connections and control the exchange of information through 'gates', which are multiplicative units. Namely, these are the input, output and forget gates, which respectively control the inflow or output of activation's to or from the cell or scales its internal states before using them recurrently, which can be interpreted as forgetting. This makes LSTM RNNs an interesting approach when working with longer time series.

Another approach to model long-term dependencies better are Gated Recurrent Unit (GRU) RNNs; they address the same vanishing gradient issues as the LSTM approach when back propagating the gradient of the cost function through time using a simpler structure. Only two gate types are employed by the GRU; an update gate that controls the inflow of information as well as a reset gate that decides over forgetting past information. Still exploding gradients remain an issue, which is however tackled by gradient clipping. This makes the GRU RNN have less parameters in comparison to an LSTM RNN and is therefore more light weight and has been observed to outperform the latter in several tasks. Because of these properties GRU RNNs could also be interesting for a distributed application in a detection mechanism and for frequent retraining if needed.

An alternative is posed by so-called Transformer architecture [13]. Here, attention mechanisms are used that enable capturing of global dependencies between input and output, regardless of the positions of the sample points in the time series or sequence. Here, no recurrent computation is used, allowing for better parallelization. Instead 'self-attention' is employed to reach a representation of a sequence through setting in relation the positions of the sequence. An encoder-decoder setup is used, where it performs a mapping of the input to an internal representation, which the decoder then processes to generate the output auto-regressively. The encoder and decoder both consist of feed-forward networks as well as multi-head self-attention mechanisms. This attention mechanism projects a query and key-value tuples on an output which is calculated using the weighted values. These weights are computed in turn with the query and the respective key. This yields an attention value for every query-key-value item and therefore a representation of the sequence. Multi-head attention now enables processing information from a higher dimensional query-key-value set at various positions in contrast to a single attention head, which is helpful. Additionally, positional encodings are simply added to the initial inputs to insert some hint about the positions of the points of the sequence for the feed-forward networks. This non recurrent approach could be an, also computationally, interesting option.

The R-Transformer concept follows a similar idea as the aforementioned transformer approaches [14]. The main improvement proposed over the regular transformer consists in additional capturing of the sequential information in the data. This is done by positional encoding in the regular transformer, which only yields a scant impact. Furthermore, local structures are neglected because of the sheer number of other positions which allows only for a small signal at a local position, even if these structures might be of quite importance. To combat

these flaws, the R-transformer uses local RNNs sliding over the sequence applying windows of defined length to encode the sequential information in the data and capture local structures in the time series. Thus, latent representations are generated equally for each of the windows treated by the local RNN and not dependent on any of the other windows. Therefore, information about its local surrounding is ingrained in each data point's representation. Additionally, by sliding the RNN over the time series, the global sequentiality of the data is taken into account as well. The effect of the local RNNs can be compared to a one dimensional convolution operation, which has the advantage of being parallelizable, but also taking into account sequential information. The so gained and encoded local information of one position is then, like in the aforementioned transformer, directly connected to all other positions in the sequence through the multi-head attention mechanism. In a similar application to the one at hand (MNIST dataset with 784x1 sequences), the R-Transformer outperforms both the regular, a convolutional Transformer as well as simple recurrent approaches as LSTM and GRU, whereas an RNN performed significantly worse than all other approaches. This makes the R-Transformer an interesting approach.

Summarizing, the work on anomaly detection (see Table I) in the electrical grid domain shows that there are approaches that are not flexibly applicable to new devices or are only applicable at a transformer level. However, the domain of DL-based approaches offers methods that are, at least in theory, well suited for developing a solution to bridge this gap. Nevertheless, no applications to this specific problem can be found in literature and therefore, explorations and assessments of these have to be conducted.

III. Scenario for Monitoring and Detection

A. Employed Framework

To overcome the shortcomings of present approaches for detecting malfunctions an environment is introduced which is able to handle different detection scenarios, grid setups and data properties. This kind of a framework (see Figure 2) is used to either synthesize or clean, process and analyse data as well as apply DL methods on it. Either real world operational grid data or data of simulations using grids that are specifically designed for simulation purposes – like that form the SIMBENCH[15] project – are being used.

TABLE I: Non-functional requirements (NFR) fulfilled (X) or unfulfilled (–) by approaches in related publications cited.

	Reference						
NFR	[9]	[8]	[10]	[12]	[11]	[13]	[14]
Scalability	–	–	–	–	–	X	X
Adaptability	–	–	X	–	X	X	X
Integrability	X	X	X	X	X	X	X
Usability	X	X	X	X	X	X	X
Data Retention	X	–	X	X	X	X	X
Robustness	–	X	X	X	X	–	X
Quality	X	X	–	X	X	X	X

If operational data is to be synthesized, the grid data used is extracted from the respective files and prepared for further use in simulations. In this manner, the grid topology is checked and generation and load profiles as well as control curves are defined and handed over to the grid simulation software. Using simulations another plausibility and – if necessary – scaling of, for example, loads is conducted and a final grid model is yielded. It is then used for running simulations in which parameters like the time resolution of the data synthesized, the misconfiguration of interest and its the position as well as the control curve to be monitored can be varied. The simulation then delivers operational data of the grid including data of a malfunction, which is then labelled and saved. These results are finally used to pick relevant data, add noise to it and therefore create data sets. These data sets are used to asses the applicability of machine learning detection methods, especially DL approaches in this case.

B. Tackled Scenarios

What this looks like in practice, is illustrated by the schematic of a distribution grid with household loads and PV generation in Figure 3: one possible misconfiguration is shown. Here all PV inverters follow a certain control curve regarding to the power factor. As mentioned above, this is meant to help regulating the voltage in cases of high active power infeed through the variation of reactive power dispatch. One of the PV units inverts its control curve, it is therefore misconfigured and the voltage is not controlled as intended anymore, which is to be detected. For PV inverters, other possible misconfigurations involve a flat control curve, which equals no control, and different maximal or minimal power factors. This allows an assessment of how grave a misconfiguration has to be as to be detected by certain approaches. These other misconfigurations, but also misconfigurations in other devices such as battery energy storages, are supposed to be equally detectable using this approach; being grid supporting a change in behaviour should leave a similar impact on the operational data, as the voltage. The similarity of features should therefore make a detection possible.

The voltage at the coupling points of the loads and PV units is recorded, for example, with a sample rate of 15 minutes to mimic smart meter data. This data is then turned into a data set by creating samples of a certain sequence length, labelling the same in classes 0 (regular behaviour) and 1 (misconfiguration present) as well as choosing the ratio of classes, either balanced or unbalanced to an arbitrary degree, to fit the capabilities of the methods applied later. Finally, these labelled samples are fed into a data driven detection method to train on them and assess its performance in detecting a malfunction by recognizing the correct classes. The datasets compiled and used consist of either weekly or daily timeseries sampled in 15 min intervals (i.e., common for power system applications), which leaves us with either 96 or 672 datapoints per sample sequence. This allows for an assessment of the impact of sequence length on the performance of the applied

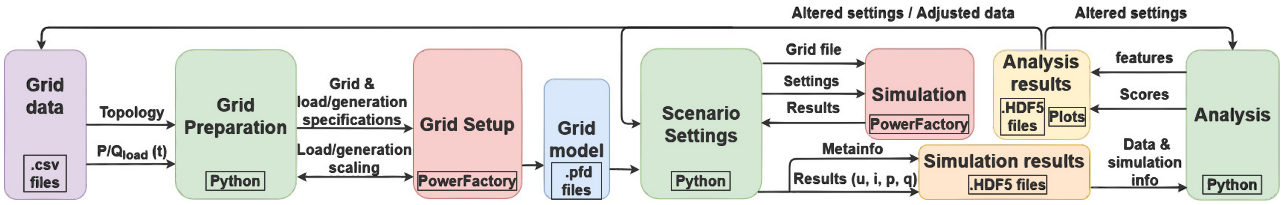

Fig. 2: Framework used for generation and handling of data of misconfigured devices in power grids as well as for assembling data sets using this data and applying and assessing methods and algorithms for misconfiguration detection.

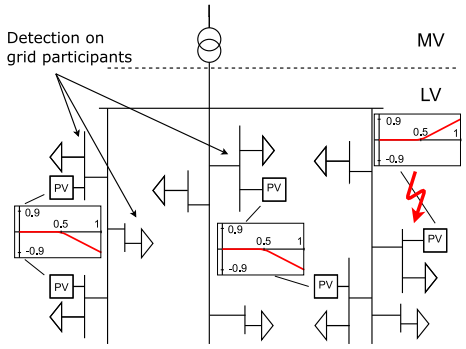

Fig. 3: Schematic grid used to generate data [16].

DL methods, which is supposed to stem from their respective handling of long-term dependencies in a sequence.

The data used for this work are created using 5 of the aforementioned SIMBENCH grids, which are either classified as rural or semi-rural since in such networks voltage issues are prevalent over current issues, making the misconfiguration relevant in these grids. Figure 4 shows in two weekly time-series samples the impact left by the misconfiguration on the operational data gained, namely the voltage. The variation in voltage for class 0 ('regular behaviour') is much smaller than for the malfunctioning class 1 ('malfunction/misconfiguration present'). This behaviour is what is expected here since the control is implemented as to keep the voltage within certain admissible limits. Therefore, this different impact of the misconfigured power factor control curve is to be detected. Various data sets with up to 200,000 samples of these kinds with balanced classes, to enable proper learning of features and classification using DL [17], were split into a train and test set and used for the adaption and assessment of the DL detection approaches described in the following.

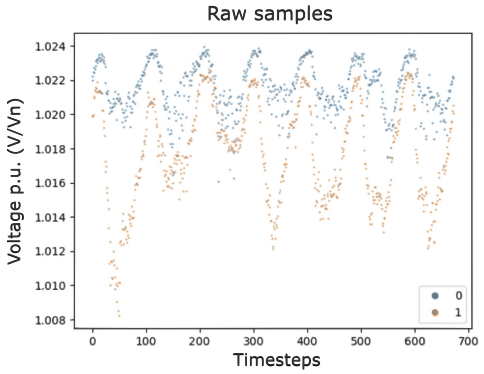

Fig. 4: Samples of both classes (0 (blue): regular; 1 (orange): misconfiguration present in grid connected device) used for Deep Learning.

IV. Applied Learning Methods & Achieved Results

A. Data Used

The data shown above has been slightly preprocessed; before its usage in the DL methods by subtracting its mean from every sample as to eliminate the influence of a grid feeder based voltage offset, as well as scaled to a range between -1 and 1. The scaler for this was fit on the training set, scaling all zero-meaned training sample between -1 and 1, and then later applied on the test set. Such samples were assembled to data sets of different sample sizes (1 day and 7 days respective 96 and 672 positions timeseries length) and sample numbers (1,000, 5,000, 10,000). Bigger sample sizes imply more data in this case, but longer timeseries might not be able to propagate back the gradients through time through the entire timeseries.

B. Method Implementation

As a loss criterion PyTorchs CrossEntropyLoss[1] is applied, which combines the LogSoftmax and negative log likelihood loss (NLLLoss). The input is expected to be the raw, untreated score of each of the two classes, as well as a class label.

Figure 5 depicts the most basic structure of the Elman network trained. There, a simple RNN with 2 layers with 6 features in the hidden states each as well as a fully connected layer with 6 neurons and 2 output neurons is presented. The output neurons obviously predict the classes 0 and 1. Each timestep is fed into the network, and the output of the final timestep, as it is the 'most informed' output, is used for calculating the loss and updating the weights as well as for making a classification. This approach was stuck to in all the recurrent approaches implemented.

The first goal was to train at least a weak learner, meaning that the output of the classifier should be more accurate than guessing. In the case of the malfunction detection task presented before, this was achieved at a sample number of 5,000 for the 1 day timeseries dataset as well as for the 7 days timeseries dataset. This was achieved only using data created using one grid as to be able to tell if there was even enough information in the data as to make a meaningful classification (i.e., for this task the F-score using the most data reached by the network was slightly over 0.5). Furthermore, a very small learning rate of 10^{-6} had to be chosen to reach sufficiently good results with a standard stochastic gradient (SGD) optimizer. The learning rate was controlled in a manner so as to increase the learning rate by a factor of 1.1 in case the loss between epochs diminishes, and decrease it in turn by a

[1] https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html

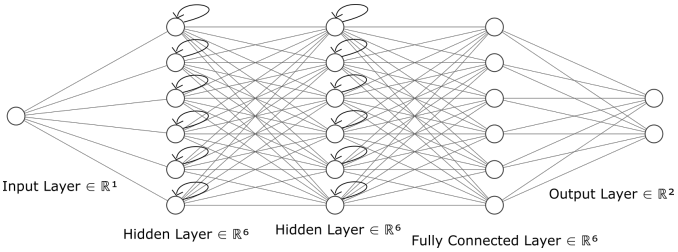

Fig. 5: Schematic depiction of the RNN trained and used.

factor of 0.9 at an increasing loss. Training was conducted for up to 100 epochs. A comparison with a linear model showed that the linear classifier did no better than guessing and therefore only reached a F-score of 0.33 on the balanced datasets. The RNN architecture put to trial here consisted of 5 RNN layers each consisting of 20 hidden units and a feed forward layer with 20 neurons as well. Training here and in the following experiments is always conducted for 20 epochs with a learning rate of 10^{-3} if not stated otherwise. The RNN approach was trained using SGD and Adam optimizer on the 1-day and 7-day samples datasets, with 200,000 samples from 5 grids and 20,000 samples from 1 grid.

As a first alternative to the simple RNN structure, an LSTM RNN was tried out. The architecture used also consisted of 5 LSTM layers and a feed forward layer with 20 hidden units respective neurons per layer, arranged in the same manner as for the simple RNN. An SGD optimizer was used for training.

To be able to compare the 'improved' simple recurrent approaches, for the GRU RNN the same architecture was chosen as for the LSTM RNN. As optimizers SGD and Adam were used when training on the same data as above.

The transformer as the only non-recurrent detection approach using an attention mechanism was used with an architecture of 5 feed forward layers with 20 neurons each. The attention mechanism constituted of a multi-head attention with one head at first. Here, an SGD optimizer was used.

Finally, the most sophisticated architecture used is the so called R-Transformer, following [14] which incorporates both attention mechanisms as well as recurrent and feed forward neural networks, as lined out in Figure 6. The multi-head attention approach allows to relate a part of a sequence to any other part of the sequence as it treats them all equally but encodes them positionally at the same time. This helps learning global dependencies, while neglecting local structures, which might also be of great interest during the course of a day. Therefore, each part of the sequence is processed beforehands by an RNN; a window of a certain number of points is slid over the sequence capturing local sequential information. In this architecture, this window had a size of 7 data points. Furthermore, the local RNNs were GRU RNNs of which 4 layers with 3 hidden units each were used. This was decided following a singular experiment conducted on the 7-day 200k dataset in which GRU reached an F-score of 0.51 after training for 47 epochs at a learning rate of 10^{-5}, outperforming RNN and LSTM. The multi-head-attention used had one head to be able to assess the impact the recurrence has in comparison to

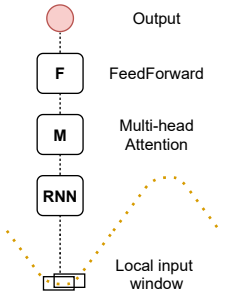

Fig. 6: Structure of the R-Transformer used.

the regular Transformer. In a first approach, only one block of stacking a local RNN, a multi-head-attention network and a feed forward layer was used. An SGD optimizer was used.

C. Results

As a result metric the expressive F-score was used, which combines and balances Precision, how many of the found misconfigurations are actually ones, and Recall, how many of the misconfiguarations present have been found. This allows quick understanding of how helpful a result is to a grid operator, since a DSO wants to balance between false alarms and finding all occurences.

The code used to produce the datasets and results can be found in the corresponding GitHub repository[2]. The results of the assessment for the small data set sourced from 1 grid as well as the big data set collected from 5 grids are summarized in Table II. In this context, a Weak Learner is performing better than the linear model which only guesses and therefore reaches an F-score of 0.33. The results achieved here are not good enough for actual usage, however they provide a good orientation for further refinement of methods.

V. Conclusions

A. Achievements

As the necessary integration of decentralized renewable energy generation proceeds, grid operators need novel ways to monitor the functionalities these generation units provide. They are crucial to the safe and reliable operation of power distribution grids. Thus, the framework described in this work allows for development of such monitoring capabilities by extracting and handling data as well as use them for the development and assessment of machine learning methods for this purpose. Several DL-based approaches have been described and compared in varying settings.

The RNN approach presented already demonstrates the applicability of DL for this task. This quite simple approach already yielded a weak learner for the 1 grid case, that can be extended to an ensemble method or be replaced by more sophisticated algorithms and network structures. Nevertheless, training had to be conducted with a very low learning rate and for a long time. When only trained for fewer epoch and with a higher learning learning rate, the RNN can not tackle the problem and does no better than the linear model.

[2]https://github.com/DavidFellner/Malfunctions-in-LV-grid-dataset

TABLE II: Overview of the results found using different sequence length, data set sizes and classifiers: the F-score balances Precision and Recall.

Model	RNN		LSTM RNN		GRU RNN		Transformer		R-Transformer	
Setup #grids & #samples	1 grid 20k	5 grids 200k	1 grid 20k	5 grids 200k	1 grid 20k	5 grids 200k	1 grid 20k	5 grids 200k	1 grid 20k	5 grids 200k
F-score 1 day-dataset (sequence length: 96)	0.33	0.33	0.34	0.33	0.47	0.33	0.33	0.33	0.49	0.47
F-score 7 day-dataset (sequence length: 672)	0.33	0.33	0.37	0.33	0.39	0.33	0.33	0.33	0.52	0.51
Weak Learner (better than linear model)	No	No	Yes	No	Yes	No	No	No	Yes	Yes

The LSTM and GRU RNN approaches both provide an improvement here, both yielding a weak learner for the 1-grid case. This shows that training can be done much faster with these approaches than with the simple RNN, probably because of the better back propagation of gradients through time. The GRU RNN performed significantly better than the LSTM RNN especially in the 1 day case, making it the more efficient structure. Therefore, GRU was chosen as the local RNN for the R-Transformer. Both approaches failed to provide a meaningful result on the dataset sourced from multiple grids.

The Transformer as sole fully non recurrent method showed that in the setting chosen feed-forward-only architectures do not yield satisfactory results as neither in the 1 grid nor in the 5 grid setup the linear model could be outperformed.

The R-Transformer posed the most complex approach under scrutiny, which also yielded the best results for the 1 grid 20k samples dataset, remarkably showing better performance on the 7 day data. This marks the impact the attention mechanism has as it improves the handling of longer sequences in comparison to the other recurrent approaches. Comparing to the feed-forward Transformer the advantage of using the local GRU RNN become obvious as the R-Transformer manages to provide meaningful classification. Especially on the 200k samples dataset from 5 grids the combination of these two features shows its strength as the R-Transformer is the only architecture that manages to gain traction in this setup and yield a weak learner. The performance is slightly higher for the short sequence length though, probably due to the simple network architecture used and a resulting lack in capacity.

The study conducted shows how the framework can be utilized to explore methods, which lead in this case to the finding that the R-Transformer outperformed its competitors, which however still provided mostly functional solutions.

B. Outlook

The presented work is a foundation for a future decision support tool for power grid operators which helps them to implement central monitoring of low voltage grids using DL detection approaches. Further work includes extensive architecture exploration in order to find the best fitting approach and an optimal model thereof for the task at hand. When such a model is found a field trial in real world grids for validation and further refinement of the method can be conducted. Furthermore, the range of use cases is to be expanded by training models on data of malfunctioning devices such as battery energy storage, electric vehicle charging stations or heat pumps. This would then lead to an implementation in said decision support tool and therefore integration into a grid operators toolbox of further monitoring capabilities.

ACKNOWLEDGMENT

This work received funding from the Austrian Research Promotion Agency (FFG) under the "Research Partnerships – Industrial PhD Program" in DeMaDs (FFG No. 879017).

REFERENCES

[1] E. Brown, J. Cloke, and J. Harrison, "Governance, decentralisation and energy: a critical review of the key issues." Loughborough University, 2015.

[2] J. Von Appen, M. Braun, T. Stetz *et al.*, "Time in the sun: The challenge of high pv penetration in the german electric grid," *IEEE Power and Energy Magazine*, vol. 11, no. 2, pp. 55–64, 2013.

[3] N. Mahmud and A. Zahedi, "Review of control strategies for voltage regulation of the smart distribution network with high penetration of renewable distributed generation," *Renewable and Sustainable Energy Reviews*, vol. 64, pp. 582–595, 2016.

[4] P. P. Vergara, T. T. Mai, A. Burstein, and P. H. Nguyen, "Feasibility and performance assessment of commercial pv inverters operating with droop control for providing voltage support services," in *2019 IEEE PES Innov. Smart Grid Techn. Europe (ISGT-Europe)*, 2019, pp. 1–5.

[5] W. Luan, J. Peng, M. Maras, J. Lo, and B. Harapnuk, "Smart meter data analytics for distribution network connectivity verification," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1964–1971, 2015.

[6] D. Fellner, H. Brunner, T. Strasser, and W. Kastner, "Towards data-driven malfunctioning detection in public and industrial power grids," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2020, pp. 1–4.

[7] N. Mehdiyev, J. Lahann, A. Emrich *et al.*, "Time series classification using deep learning for process planning: A case from the process industry," *Procedia Computer Science*, vol. 114, pp. 242–249, 2017.

[8] W. Cui and H. Wang, "A new anomaly detection system for school electricity consumption data," *Information*, vol. 8, p. 151, Nov. 2017.

[9] D. D. Sharma, S. Singh, J. Lin *et al.*, "Identification and characterization of irregular consumptions of load data," *Journal of Modern Power Systems and Clean Energy*, vol. 5, no. 3, pp. 465–477, 2017.

[10] M. Hüsken and P. Stagge, "Recurrent neural networks for time series classification," *Neurocomputing*, vol. 50, pp. 223–235, 2003.

[11] S. Kanai, Y. Fujiwara, and S. Iwamura, "Preventing gradient explosions in gated recurrent units," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[12] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," *INTERSPEECH*, pp. 338–342, Jan. 2014.

[13] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[14] Z. Wang, Y. Ma, Z. Liu, and J. Tang, "R-transformer: Recurrent neural network enhanced transformer," *arXiv preprint arXiv:1907.05572*, 2019.

[15] S. Meinecke and et al., "Simbench—a benchmark dataset of electric power systems to compare innovative solutions based on power flow analysis." *Energies*, vol. 13.12:3290, 2020.

[16] D. Fellner, "Data driven detection of malfunctions in power systems," in *Proceedings 9th DACH+ Conference on Energy Informatics*, 2020.

[17] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, 2016.