Fault Detection in DC Microgrids using Recurrent Neural Networks

Ivan Grcić
University of Zagreb
Faculty of Electrical Engineering and Computing
Zagreb, Croatia
ivan.grcic@fer.hr

Hrvoje Pandžić
University of Zagreb
Faculty of Electrical Engineering and Computing
Zagreb, Croatia
hrvoje.pandzic@fer.hr

***Abstract*—Reliable and accurate fault detection plays a crucial role in the microgrid operation by enabling an increased operational flexibility. Successful classification of events in complex microgrid systems requires advanced models of sufficient speed and accuracy. Deep neural networks meet these requirements, as they have demonstrated their capabilities in a wide range of applications. In particular, Recurrent Neural Networks (RNNs) are used for sequence learning, making them suitable for online fault detection. In this work, the RNN is applied to the time-domain signal to detect faults in a photovoltaic-based DC microgrid. The classifier successfully discriminates all events and proves its performance using various metrics.**

***Index Terms*—fault detection, microgrid protection, deep learning, recurrent neural networks**

I. Introduction

The concept of a microgrid was introduced to denote a local group of distributed energy resources (DERs), storage systems, and loads that can operate as a single entity in a cooptimized manner. The DERs are usually directly connected to the low or medium voltage distribution system, where their benefits can be adequately utilized [1]. The power quality, resiliency and reliability of the main grid is increased when introducing controllable microgrids, while transmission losses are reduced [2]. Therefore, the load is less affected by the grid disturbances or failures. However, the presence of DERs has a negative impact on the power system protection. The traditional protection methods cannot handle with the conditions that occur in the microgrid environment, so new protection methods must be developed.

The integration of renewable DERs, especially small photovoltaics and wind turbines, is increasing. Their full production potential should be exploited for economic and environmental reasons. Control algorithms are used to adjust the sources to the maximum power point and provide an intelligent energy management. However, in addition to optimal microgrid control and energy management, system efficiency improvement should also be considered. Since energy storage systems, photovoltaics, and advanced loads, such as electric vehicles, all use DC voltage, DC microgrids could be an acceptable alternative to their AC counterparts. Line losses of a DC system are lower as compared to the AC system, which allows for higher transmission capacity and efficiency. In addition, the DC system is advantageous in terms of power quality and less sophisticated control is required [3].

To solve the mentioned protection problem, this paper proposes an advanced fault detection method for DC microgrid based on Recurrent Neural Network classifier. This classifier is widely used for time series pattern recognition, hence current and voltage time signals measured on the microgrid bus are given as inputs. The method is designed to distinguish the faults from the other disturbances and successfully identify a faulty microgrid instance. The system considered is a photovoltaic-based DC microgrid with integrated battery energy storage system.

II. Related Work and Contribution

Traditional protection methods have proved to be inefficient in the microgrid environment, as their main purpose is protection in transmission and distribution systems. The power flow of the aforementioned systems is violated by the inclusion of distributed energy resources (DERs), which has a significant impact on the traditional system protection. The majority of installed DERs generate electrical energy from renewable sources, mainly solar and wind. Since wind speed and solar irradiance are not constant, the production from these sources is intermittent, which can cause dynamic changes in the fault current magnitude. Other common problems include blinding of protection, unintentional islanding, and grid failure. To solve these problems, new methods had to be developed that are able to distinguish faults from other disturbances regardless on the system operating conditions. These methods can be divided into event estimation based, fuzzy based, field transform based and intelligent fault detection methods [4].

Fault detection methods must be able to correctly distinguish faults from other power disturbances in normal operating mode, such as load demand or generation changes. Since the microgrid environment can be a complex system with various distributed generators, energy storage systems and loads, classification can be a challenging task. This leads to a large number of features required for successful classification. Therefore, intelligent classifiers are used to process large amounts of data. The intelligent classifier based fault detection methods usually use machine learning models as classifiers and field transforms as feature providers. Short-time Fourier

978-1-7281-7660-4/21/$31.00 ©2021 IEEE

transform (STFT) proved to be a reliable feature provider for intelligent fault detection in DC microgrid in [5]. The amplitudes of the signal frequency components are used as input to an intelligent classifier to detect pole-to-pole fault. Discrete wavelet transform (DWT) coefficients of the current and voltage signal are provided to the Support Vector Machine classifier for the protection of a microgrid where it showed high accuracy in [6]. The DWT is also combined with the K-Nearest Neighbors classifier where the standard deviation of the DWT approximate coefficients of the voltage and current signal are features [7].

Although machine learning models proved as accurate classifiers, their insufficient capacity requires explicit feature selection, which can be a difficult task. Deep neural networks (NNs) overcome this problem by selecting features implicitly, i.e., the deep model designs the features independently [8]. The neural network as an intelligent classifier for detecting faults in microgrids has been shown to be accurate as it is able to provide information about fault type, phase and location in [9]. The neural network has also proved to be effective in the DC microgrid environment where the classifier input is the current on each side of the bus segment [10]. It is also combined with DWT as a feature provider in [11], where it successfully detects a variety of events such as pole-to-pole, pole-to-ground, and AC grid faults with high accuracy score. In [12] the fault location and in [13] both fault detection and fault location are determined using the NN. RNN is a type of Artificial NN that stores knowledge, i.e., it computes the output based not only on the input but also on the datapoints seen. RNN has been successfully implemented for the protection of a AC microgrid in [14]. Positive, negative and zero current sequence along with current magnitude are used as inputs to the classifier. In [15] RNN is used to determine fault location and resistance in a DC microgrid, but Decision Tree classifier was used for fault detection.

To further investigate the applicability of the Recurrent NNs (RNNs), DC microgrid protection and disturbance detection method is presented. The contributions of this research paper are:

- The development of an accurate fault detection method for microgrids.
- The use of RNN for fault detection in DC microgrids.
- The design of a practical fault detection method that uses a minimum number of inputs to detect faults and faulty microgrid instances.

The remainder of paper is structured as follows. Section III presents the RNN model and its optimization procedure, Section IV presents a simulation setup and RNN-based fault detection, and finally Sections V and VI present the results and conclude the paper.

III. Neural Networks for Sequential Data

Sequential data are typically modeled by a series of datapoints taken at successive, uniformly distributed points in time. The observation of a non-random signal datapoint $\mathbf{x}_i$ at time t_i depends on the previously observed values $\mathbf{x}_j$ at time t_j, where $j < i$. Since a labeled training dataset is assumed, each datapoint also contains the label y_t. The prediction of the label for a given datapoint should ideally take into account both the value of the current input and past observations, which is called *context*.

A. Feedforward Neural Networks

A standard feedforward neural network computes the output based on the current input using a composition of transformations. Composite transformations are parameterized with a set of parameters $\boldsymbol{\theta}$. A shallow feedforward neural network is defined with the following transformations:

$$\mathbf{h}^{(t)} = \sigma(\mathbf{U}\,\mathbf{x}^{(t)} + \mathbf{b}) \tag{1}$$

$$\mathbf{o}^{(t)} = \mathbf{V}\,\mathbf{h}^{(t)} + \mathbf{d}, \tag{2}$$

where $\mathbf{x}^{(t)}$ is the input vector, $\mathbf{h}^{(t)}$ is the output of the first transformation, $\mathbf{o}^{(t)}$ is the model output, and σ an arbitrary nonlinear function. The transformation is parameterized with the set $\boldsymbol{\theta} = (\mathbf{U}, \mathbf{V}, \mathbf{b}, \mathbf{d})$. The model predicts a probability of sample $\mathbf{x}^{(t)}$ belonging to each of K classes using softmax:

$$\hat{\mathbf{y}}^{(t)} = \frac{\exp \mathbf{o}^{(t)}}{\sum_{k=1}^{K} \exp \mathbf{o}_k^{(t)}}. \tag{3}$$

The predicted class is then chosen as the one with the highest probability: $\hat{y}^{(t)} = \operatorname{argmax}\, \hat{\mathbf{y}}^{(t)}$.

Feedforward NNs compute the output $\mathbf{o}^{(t)}$ and the prediction $\hat{y}^{(t)}$ solely on the current input. This model formulation might provide the correct prediction in some cases. However, the model ignores important assumptions regarding the interdependence of the observed datapoints.

B. Recurrent Neural Networks

A recurrent neural network (RNN) incorporates the context of a given datapoint by preserving the internal state that contains knowledge about the previously seen datapoints. A recurrent model is defined by the following set of transformations:

$$\mathbf{h}^{(t)} = \sigma(\mathbf{U}\,\mathbf{x}^{(t)} + \mathbf{W}\,\mathbf{h}^{(t-1)} + \mathbf{b}) \tag{4}$$

$$\mathbf{o}^{(t)} = \mathbf{V}\,\mathbf{h}^{(t)} + \mathbf{d} \tag{5}$$

The parameter set $\boldsymbol{\theta}$ is extended by the parameter $\mathbf{W}$, which models the context of the current datapoint. Thus, the assumption of mutually dependent datapoints is included in the model design. The unfolding of the RNN structure in Figure 1 shows how the RNN can incorporate values of previous datapoints into the current prediction.

C. Iterative Gradient Optimization of Recurrent Neural Networks

Predicting whether the microgrid is in a faulty state can be viewed as a classification task. The standard classification task considers a set of labeled datapoints $D = \{\mathbf{x}^{(t)}, y^{(t)}\}_{t=1}^{N}$ and a model defined with a set of trainable parameters $\boldsymbol{\theta}$. The goal of the optimization task is to find the optimal set of parameters $\boldsymbol{\theta}$ for which the model correctly classifies the points of a given

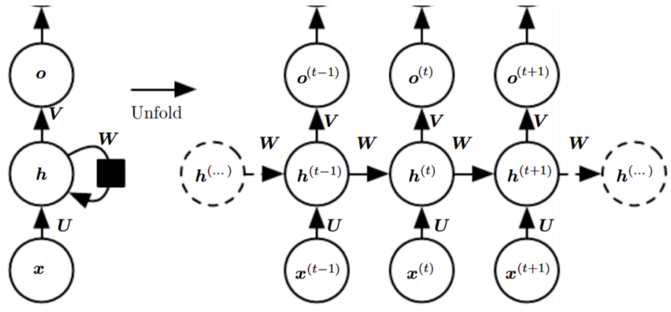

Fig. 1. Unfolded RNN architecture shows how previously observed datapoints are incorporated into the current prediction [8].

dataset. Optimizing the set of parameters $\boldsymbol{\theta}$ with respect to the desired objective requires the definition of the loss function L. The loss function should match with the defined objective in such a way that minimizing the loss function results in a step towards the optimization objective.

The loss function for multiclass classification, called cross-entropy loss is defined as follows:

$$L(\mathbf{x}^{(t)}, \mathbf{y}_{oh}^{(t)}, \boldsymbol{\theta}) = -\sum_{k=1}^{K} \mathbf{y}_{oh,k}^{(t)} \ln \hat{\mathbf{y}}_k^{(t)}. \tag{6}$$

$\mathbf{y}_{oh}^{(t)}$ represents the one-hot representation of the label $y^{(t)}$, while $\hat{\mathbf{y}}^{(t)}$ is the output of the model. In this case, the model is a recurrent neural network. The defined loss function L is derivable along its domain at any point. Therefore, the value of the function can be minimized if each model parameter $\boldsymbol{\theta}_j$ is shifted in the direction of its negative gradient. The gradient of each parameter can be easily calculated by the backpropagation algorithm based on the partial differentiation rule $\frac{\partial L}{\partial \boldsymbol{\theta}_j} = \frac{\partial L}{\partial h} \frac{\partial h}{\partial \boldsymbol{\theta}_j}$.

Finding the set of optimal parameters is done in an iterative manner. Finding such a set of parameters can be done using gradient-based algorithms, which use the current state of the parameters along with the computed gradients to compute a more advantageous set of parameters:

$$\boldsymbol{\theta}^{(i+1)} = \text{gradient\_algorithm\_step}(\boldsymbol{\theta}^{(i)}, \frac{\partial L}{\partial \boldsymbol{\theta}^{(i)}}) \tag{7}$$

The optimization procedure is independent of a particular gradient-based algorithm. In this work RMSProp optimization algorithm is used.

The given dataset can be of any length. Computing gradients after each sample is impractical because the training time of the model is radically increased and the gradient obtained using a single sample can be noisy. On the other hand, updating the model parameters for the entire dataset results in a suboptimal number of updates. Therefore, the parameters are updated using the gradient of data batches (minibatches). Typically, the size of a minibatch corresponds to 16, 32 or 64 datapoints. By using minibatch gradients, the gradient is denoised while the number of updates remains feasible.

Since the surrogate loss function is optimized instead of the non-derivable objective function, the optimization procedure leads to the overfitted solution. The overfitted solution does not generalize its prediction well to datapoints not seen during training. The overfitting is prevented by early stopping. This procedure involves splitting the given dataset D into two distinct datasets. The training dataset D_{train} contains about 90% of the total samples, while the validation dataset D_{val} contains the remaining datapoints. The training dataset is used in the optimization procedure for model learning. The validation dataset is used for performance evaluation of the model. Once the performance of the model on the validation dataset starts to decrease, the optimization procedure is stopped. Equivalently, an increase in cross-entropy loss on a validation dataset is an indication that the model has been overfitted.

IV. RNN-based Fault Detection

The proposed RNN-based fault detection method is described in this section. First, the observed DC microgrid with all simulated events is presented, followed by the description of the method and the architecture of the classifier.

A. Simulation Setup

The simulation setup is a photovoltaic-based DC microgrid with an integrated lithium-ion battery energy storage system (BESS) (Figure 2). The PV array is connected via a buck and BESS via a bidirectional buck-boost DC-DC converter to the DC bus, which is connected to the distribution grid via a voltage source converter (VSC). The PVs are in the maximum power point tracking (MPPT) control mode, with power output determined by temperature and irradiance. The BESS's converter is in the constant current control mode. The parameters of the microgrid can be found in Table I. Simulation is conducted in the Matlab Simulink environment.

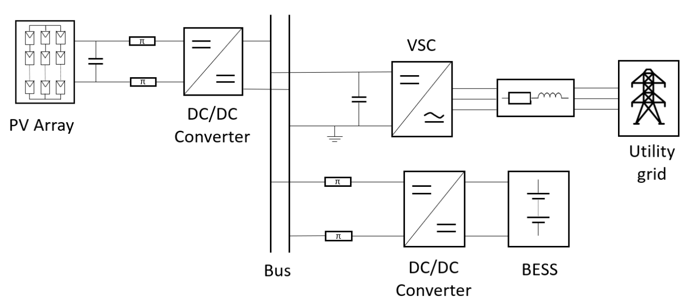

Fig. 2. Microgrid model.

B. Fault Detection

Since fault detection relies on the ability of a classifier to correctly categorize events, a possible confusion of a classifier is reduced if the training dataset contains as many events possible. This way, the classifier can more carefully choose features to ensure a firm boundary between the classes. The events occurring in the DC microgrid environment can be grouped in two main categories, namely load/generation changes and faults, which can be further divided into pole-to-pole and pole-to-ground faults. The observed microgrid contains a converter-interfaced PV source and an energy

TABLE I
MICROGRID PARAMETERS

PV array	44 kW
Line resistance	0.641 Ohm/km
Line inductance	0.34 mH/km
Line capacitance	0.1 μF/km
Line length	0.3 km
DC link capacitance	20 mF
Bus voltage	500 V
Battery pack nominal voltage	120 V
Battery pack nominal power	12 kW
Battery pack SOC	90 %

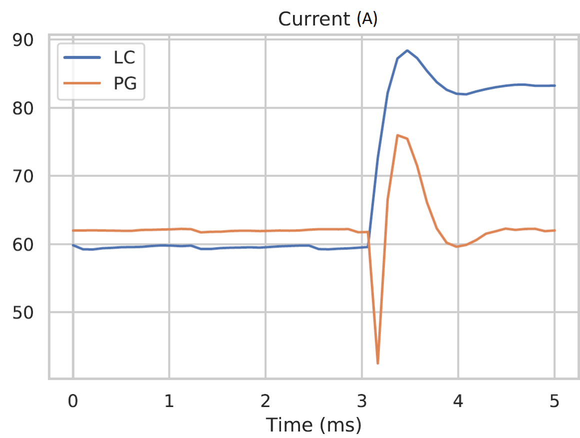

Fig. 3. DC bus current during battery load change and PG fault.

storage. Generation change of a PV array is fabricated by the irradiance change. Since the BESS is in the constant-current control mode, its load change is accomplished by changing the current reference. Faults are simulated at different locations with varying value of the fault resistance ranging from 0.1 to 20 Ω, for both the PP and the PG faults. Fault locations are at the DC bus, and at the PV array and BESS output terminals. In addition, faults are simulated for different operating conditions of the microgrid. Output power of the PV array is set to several points ranging from 0 to $P_{PV,max}$. The same is applied to BESS, with the difference that its power output can also be negative (acting as a load).

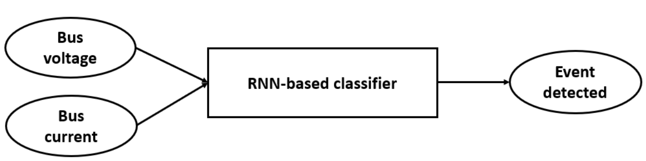

Fig. 4. Overview of proposed RNN-based fault detection.

Each microgrid instance is connected to the bus, directly or via converter. Therefore, all events that occur in the microgrid affect its current and voltage. This idea is used for fault detection, where the inputs of the classifier are the bus voltage and current signals. With only these two measurements, the fault detector is able to detect faults on all instances within a microgrid, regardless of the fault location or resistance. The bus current during the battery load change and the PG fault can be seen in Figure 3. From the figure, it can be seen how context can help in fault detection as the waveforms of both signals are similar after the event occurs. Figure 4 shows the overview of the proposed fault detection method.

Model architecture with the following properties is used. A linear transformation is applied to each input (current and voltage), increasing the dimensionality to 48, followed by the Rectified Linear Unit (ReLU) nonlinearity. Two recurrent layers have memory size of 96 and use hyperbolic tangent (tanh) nonlinearity. Finally, the model output is generated by ReLU followed by a linear transformation to a K-dimensional vector, where K is the number of classes. Softmax, a function that converts a real-valued vector into a vector whose elements sum to one, is applied to the output to generate probabilities per class for each sample. Overview of the model architecture is shown in Figure 5.

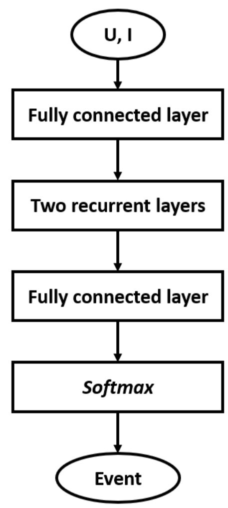

Fig. 5. RNN-based classifier model architecture.

The performance of the RNN classifier is validated using two metrics, accuracy and F1-score. Accuracy is the rate of correctly classified samples and the total number of samples. It is an intuitive metric, but an asymmetric dataset may lead to an exaggerated performance score. The F1-score is more conservative metric compared to accuracy. Its value is calculated as the harmonic mean of precision and recall. Precision and recall for binary classification problems are defined using the following definitions. Precision is the ratio between the number of correctly predicted positive samples and the total number of positively predicted samples. Recall is the ratio between a number of correctly predicted positive samples and all true positive samples. These metrics can be generalized for multiclass classification.

Since eight different events are simulated, each on a different microgrid instance, the classifier is trained to distinguish between all events. There are nine classes: normal operating condition, PP and PG faults at the bus, BESS and the PV,

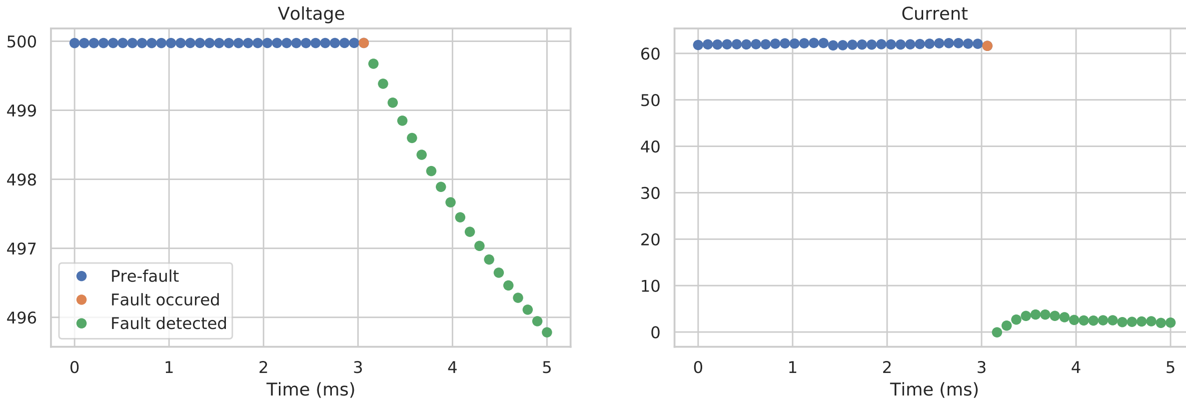

Fig. 6. Fault detection example with marked samples.

and load/generation changes of the BESS and PV. The inputs of the classifier are voltage and current signals, but it is also trained with each signal separately.

A total of 15120 sequences, each containing 50 samples, were used for training the proposed classifier. The remaining 1680 sequences are used for validation. Time step between the two consecutive samples is 100 μs. The number of current and voltage sequences are same.

V. Simulation Results

The results for classification of nine classes are listed in Table II. The method works best when both the voltage and the current are observed, achieving the accuracy of 99.81% and the F1 score 99.60%. When the number of input signals is reduced, the current signal proves to be a better choice for the detection task as it achieves a slightly higher score than the voltage-only classification. Since the classifier successfully distinguishes all events, it can be concluded that each event is characterized by certain features. The distinction arises from the fact that different types of converters will react differently to the fault due to differences in their topology [16]. The control strategy and controller parameters also affect the response of a converter to the disturbances and load or setpoint changes. However, it should be noted that deep neural network models select features implicitly, which means that the features are constructed in a complex way not known to the user. In addition, reducing the number of inputs results in less information provided to the classifier and a lower performance score.

TABLE II
Performance of RNN Classifier for 9 Classes

Input	Accuracy %	Precision %	Recall %	F1-score %
Voltage	94.82	84.59	84.31	84.10
Current	95.30	89.42	87.36	87.85
Voltage + Current	99.81	99.58	99.62	99.60

Reducing the number of classes from 9 to 4 by combining the load changes of PV and BESS into first, all PP faults into second, and all PG faults into a third class. This reduction does not have a large impact on accuracy (Table III). However, the F1-score increases, which means that the classification performance is improved by using fewer classes. This improvement is expected since the classifier can now use features shared by two classes. Performance remains better when both the current and the voltage are used as inputs, the same as in the case with nine classes. However, the ability to identify a faulty microgrid instance is lost as the number of classes is reduced.

TABLE III
Performance of RNN Classifier for 4 Classes

Input	Accuracy %	Precision %	Recall %	F1-score %
Voltage	94.26	89.93	89.06	89.31
Current	94.87	92.78	86.93	89.05
Voltage + Current	99.33	98.31	98.06	98.18

The number of classes can be further reduced by linking the classes PP and PG. This gives remarkable results as classification with voltage and current signals as inputs achieves 99.99% performance score (Table IV). However, combining faults into one class prevents the classifier to identify what type of fault has occurred.

TABLE IV
Performance of RNN Classifier for 3 Classes

Input	Accuracy %	Precision %	Recall %	F1-score %
Voltage	99.24	98.47	90.77	94.01
Current	99.84	99.74	98.06	98.87
Voltage + Current	99.99	99.99	99.99	99.99

The event detection time is analyzed for the 9-class case where both voltage and current are given as inputs to the classifier. Figure 6 shows voltage and current signals after the fault occurs at $t = 3$ ms. It can be seen that the second sample is successfully classified after the occurrence of the fault (marked in green in the figure). This sample reaches the classifier after 200 μs. Another 50 μs (Intel(R) Xeon(R) 2.3

GHz 2-core processor) are needed to process this input and make a decision to which class the sample belongs.

Comparison of the results shows that reducing the number of classes increases the performance score, but in case of current and voltage input, the increase is negligible, as it can be seen in the Figure 7. Therefore, it is unnecessary to trade accuracy for less information about the fault type and location. However, if identification of the fault is not important, using only three classes with only current input reduces both the complexity of the method and the number of measurements required. In addition, using both voltage and current as inputs increases classifier performance because two signals carry more information than one.

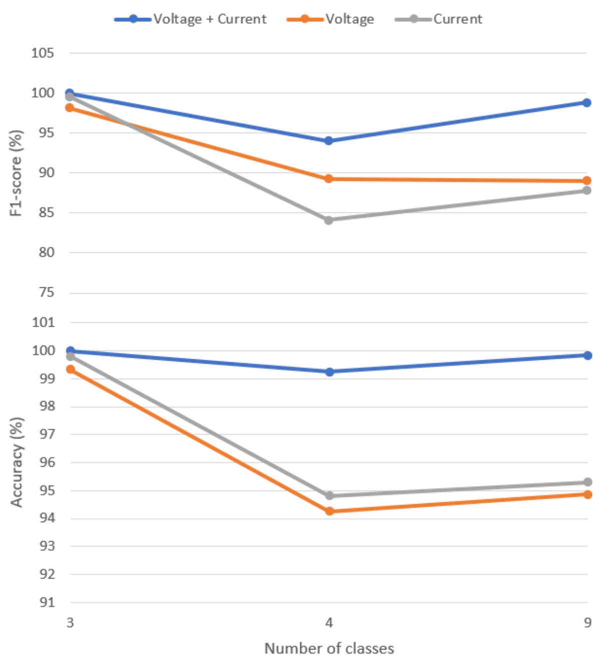

Fig. 7. Graphical representation of Accuracy and F1-score.

VI. Conclusion and Discussion

In this paper, the RNN based method is proposed for fault detection. The method is validated in the DC microgrid environment, where it succeeded in distinguishing faults from other events occurring under different operating conditions. The inputs to the classifier are voltage and current signals measured on the microgrid bus. The performance of the classifier was validated using the accuracy and F1-score metrics, where it achieved a high performance score.

Despite the high score obtained, the method has certain drawbacks that are characteristic for machine learning models. The quality of the input data plays a crucial role in training the classifier. The quality increases with the number of possible different events included. This often requires a large number of simulations and an accurate system model when real world data is sparse or non-existent. Moreover, the detection time directly depends on the computation time, which is limited by the processor power.

Acknowledgement

This work was supported by the Croatian Science Foundation and the European Union through the European Social Fund under the project Flexibility of Converter-based Microgrids–FLEXIBASE (PZS-2019-02-7747).

References

[1] T. Funabashi, *Integration of Distributed Energy Resources in Power Systems: Implementation, Operation and Control*. Elsevier Inc., 3 2016.

[2] V. Sarfi and H. Livani, "An economic-reliability security-constrained optimal dispatch for microgrids," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6777–6786, 11 2018.

[3] C. N. Papadimitriou, E. I. Zountouridou, and N. D. Hatziargyriou, "Review of hierarchical control in DC microgrids," pp. 159–167, 5 2015.

[4] A. Yılmaz and G. Bayrak, "A real-time UWT-based intelligent fault detection method for PV-based microgrids," *Electric Power Systems Research*, vol. 177, p. 105984, 12 2019.

[5] I. Grcić, H. Pandžić, and D. Novosel, "Fault Detection in DC Microgrids Using Short-Time Fourier Transform," *Energies*, vol. 14, no. 2, p. 277, 1 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/2/277

[6] M. Manohar, E. Koley, and S. Ghosh, "A reliable fault detection and classification scheme based on wavelet transform and ensemble of SVM for microgrid protection," in *Proceedings of the 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology, iCATccT 2017*. Institute of Electrical and Electronics Engineers Inc., 6 2018, pp. 24–28.

[7] M. Manohar, E. Koley, Y. Kumar, and S. Ghosh, "Discrete Wavelet Transform and kNN-Based Fault Detector and Classifier for PV Integrated Microgrid," in *Lecture Notes in Networks and Systems*. Springer, 2018, vol. 38, pp. 19–28.

[8] Ian Goodfellow and Yoshua Bengio and Aaron Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[9] J. J. Yu, Y. Hou, A. Y. Lam, and V. O. Li, "Intelligent fault detection scheme for microgrids with wavelet-based deep neural networks," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 1694–1703, 3 2019.

[10] Q. Yang, J. Li, S. Le Blond, and C. Wang, "Artificial Neural Network Based Fault Detection and Fault Location in the DC Microgrid," in *Energy Procedia*, vol. 103. Elsevier Ltd, 12 2016, pp. 129–134.

[11] D. Jayamaha, N. Lidula, and A. Rajapakse, "Wavelet based artificial neural networks for detection and classification of dc microgrid faults," in *2019 IEEE Power Energy Society General Meeting (PESGM)*, 2019, pp. 1–5.

[12] A. Abdali, K. Mazlumi, and R. Noroozian, "High-speed fault detection and location in dc microgrids systems using multi-criterion system and neural network," *Applied Soft Computing*, vol. 79, pp. 341–353, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494619301772

[13] I. Almutairy and M. Alluhaidan, "Fault diagnosis based approach to protecting dc microgrid using machine learning technique," *Procedia Computer Science*, vol. 114, pp. 449–456, 2017, complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30 – November 1, 2017, Chicago, Illinois, USA. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050917318136

[14] S. Esmaeilbeigi and H. K. Karegar, "Intelligent fault detection and location scheme for low voltage microgrids based on recurrent and radial basis function neural networks," in *2020 28th Iranian Conference on Electrical Engineering, ICEE 2020*. Institute of Electrical and Electronics Engineers Inc., 8 2020.

[15] A. A. Sharif, H. K. Karegar, and S. Esmaeilbeigi, "Fault detection and location in dc microgrids by recurrent neural networks and decision tree classifier," in *2020 10th Smart Grid Conference (SGC)*, 2020, pp. 1–6.

[16] S. Beheshtaein, R. M. Cuzner, M. Forouzesh, M. Savaghebi, and J. M. Guerrero, "Dc microgrid protection: A comprehensive review," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, pp. 1–1, 2019.