We express our gratitude to the reviewer for providing insightful comments and suggestions on our chapter. We have thoroughly revised and augmented our chapter to address the concerns and suggestions raised. We highlight in yellow the answers in the text of the paper, except for the comments concerning spelling errors.

Reviewer1

I consider it necessary to clarify how the disruptions used in the training and test sets were generated; from the information mentioned in Table 1, it indicates that only 6 disruptions were simulated. The authors said that the length of the input subsequence is L0 = 14, so how was the data segmented to feed the models?

We appreciate the reviewer for pointing out the lack of detail in this regard. As indicated by the reviewer, the input sequences to the network were of length 14. To build the training and test sets, the data were divided into subsequences of 14 samples. If a subsequence contained samples within the disruption, it was labeled as a disruption; if the subsequence contained no samples from the disruption, it was labeled as non-disruption. This is why the confusion matrices report 30 disruptions, referring to the total of 30 subsequences classified as disruptions. We have added this information to the chapter text.

It is mentioned that the proposed model was compared with two other models, one based on SVM and the other on an encoder-decoder type LSTM network. It would be important to mention the hyperparameters used in training said models, for example, in the case of the SVM-based model, what type of kernel function was used, and what strategy was followed to adjust its hyperparameters?

We once again thank the reviewer for their patience with our lack of attention to detail. For the SVM, a regularization parameter of 1 and a radial basis function (RBF) kernel were used. The sklearn library was utilized, and the default configuration was selected without any optimization. The LSTM-based network used two LSTM layers with 104 units each; the first was in sequence-to-sequence mode, and the second returned only the last output. The decoder had the same configuration, but at the end, a dense network with four layers consisting of 150, 50, 10, and 1 neuron was used. All layers had ReLU activation, except for the last one, which used a sigmoid activation function. As mentioned in the chapter, this architecture was based on [19]. We have added this information to the chapter text.

There are some typographical and writing errors that need to be addressed. Some of these have been pointed out in the attached file, along with suggestions for modifications for your review.

We have incorporated the changes suggested by the reviewer, thank you very much. Regarding the question about whether the layer used by the proposed network is a 1D convolutional layer, it is indeed a 1D convolutional layer, and we have clarified this in the text.