

EP2420 *Intrusion Detection - Task 3*

Federico Giarre

December 15, 2023

1 Task 3

In this task, we expect to train a Long Short Term Memory (LSTM) Neural Network by giving the alert values as features and the sequence of action as labels, thus creating a model that, given a sequence of alert values can predict the correct sequence of action issuing those alerts. Data has to be prepared to be understandable for the LSTM (as covered in Section 1.1), and results given have to be then decoded to the discrete mapping of categorical actions (as covered in Section 1.3). Finally, an evaluation of this method is given by using ACC_{start} and ACC_{action} metrics and a comparison between the model and HMM is discussed.

1.1 Data preparation

To process the data, attack categories are mapped as in the previous tasks, while alert values are kept raw and normalized using the Z-score normalization¹ method. The latter method ensures that the obtained normalized values have a mean of 0 and a standard deviation (and variance, as requested) of 1.

1.2 Modelling the LSTM

The LSTM is used to predict the sequence of attack action that produced the alert values given in input. Different numbers of hidden units and different are used in order to assess which combination leads to better results.

1.3 Data Decoding

LSTM returns continuous values, so there is the need to map them back to discrete labels related to the attack type ones. To do so, every resulting number was rounded to the next integer if the number's first decimal is above 0.5, or to the previous integer if the latter is lower than 0.5. Additionally, any negative number was mapped to the value 0.

1.4 Results

Keeping in mind the definition of the two accuracy metrics, ACC_{start} and ACC_{action} , described in the previous task, Table 1 shows the achieved results for those metrics under different hyperparameters configurations.

¹<https://www.statology.org/z-score-normalization/>

# Hidden units	Optimizer	ACC_{start}	ACC_{action}
10	Adam	7.9e-01	9.3e-01
10	AdamW	8.2e-01	9.3e-01
20	Adam	8.9e-01	9.7e-01
20	AdamW	9.6e-01	9.9e-01
30	Adam	9.6e-01	9.9e-01
30	AdamW	9.8e-01	9.9e-01

Table 1: ACC_{start} and ACC_{action} values with respect of varying number of hidden units and optimizer.

The results highlighted in green are the best results achieved with LSTM but, since the results seemed too high, a different dataset (Dataset 2) was tested against the model trained on Dataset 3 (the one assigned) to check for overfitting. Fortunately, the model obtains the same results on both datasets, thus confirming the validity of the results obtained.

1.4.1 Comparison with HMM (updated results of Task 2)

Table 2 shows the comparison of accuracy values of LSTM and the ones obtained by HMM in Task 2

Model	ACC_{start}	ACC_{action}
HMM	9.9e-01	9.1e-01
LSTM	9.8e-01	9.9e-01

Table 2: Comparison of ACC_{start} and ACC_{action} results for HMM and LSTM models.

As it is possible to notice, while both models perform an excellent job in both metrics, HMM seems to be slightly better in predicting the attack starting timestep. In contrast, LSTM performs a better prediction of the full sequence of attack action.