# Anomaly Detection on Unified Network Dataset

Alessandro Bonadei
*Università degli Studi di Brescia*
Brescia, Italy
a.bonadei@studenti.unibs.it

***Abstract*—Through continuous monitoring and modelling of network traffic, anomaly detection systems are able to identify behaviours which deviate from the model. Such behaviours may suggest threats and such systems are able to exploit them. This paper takes advantages of intuitions proposed in "Exploratory Data Analysis of a Unified Host and Network Data set" [1] and makes use of a revisited analysis method from "A Combination of Temporal Sequence Learning and Data Description for Anomaly-Based NIDS" [2]. In particular the paper proposes an anomaly-detection process based on Long Short Term Memory (LSTM) neural network and the use of Local Outlier Factor (LOF).**

***Index Terms*—Anomaly Detection, Cyber Security**

## I. DATA

### A. Data set Description

"The Unified Host and Network Data set" [3] consists of a subset of network and computer (host) events collected from the Los Alamos Laboratory enterprise network. It's divided in two data sets: the Host data set and the NetFlow data set. The Host Data set consists of host logs, which record all Windows authentication events on user machines, and NetFlow data set consists of NetFlow records, which record network activity between two devices. In this paper I worked only on the NetFlow data set, which contains these features:

- **Time** The start time of the event in epoch time format.
- **Duration** Duration of the event in seconds.
- **SrcDevice** Anonymized ID of the device that likely initiated the event.
- **DstDevice** Anonymized ID of the receiving device.
- **Protocol** The protocol number.
- **SrcPort** The port used by SrcDevice.
- **DstPort** The port used by DstDevice.
- **SrcPackets** The number of packets sent by SrcDevice during the event.
- **DstPackets** The number of packets sent by DstDevice during the event.
- **SrcBytes** The number of bytes sent by SrcDevice during the event.
- **DstBytes** The number of bytes sent by DstDevice during the event.

To create such data set, LANL used network stitching on the flows, reconciling opposing unidirectional NetFlow records of bidirectional connections into bi-flows, which are flow for which the features are separated by source activity and destination activity. For instance, rather than having one feature for the number of packets sent, a bi-flow has two features: the number of packets sent by the source device and the number of packets sent by the destination device.
LANL also aggregated records based on five tuples, which are the sets of source device, source port, destination device, destination port and protocol. By formatting the data into bi-flows and by aggregating flows, many of the events contain large duration, packet and byte values.

### B. Data Cleaning

In cases when LANL's snitching was unsuccessful, the number of source or destination packets was 0. As in the exploratory analysis [1] I removed those connections from the data set. I also dropped duplicate rows from the data set.

### C. Data Aggregation

I was interested in collecting time series regarding every single source device in order to build a robust model and to detect suspicious behaviours. The main problem was that in many cases a single source device had many rows with the same time, destination device, protocol, but with different ports. In order to simplify the analysis I therefore grouped such rows dropping the ports features, summing the remaining features.

## II. Proposed Model

The proposed architecture in "Fig. 1" combines the use of LSTM on pre-processed time series X to produce an output O. The time series Err is computed by the Relative Mean Squared Error (RMSE) between X and O.
The anomaly detection is then performed on the Err vectors. A whole time series may be anomalous, but also a single communication within the same time series may be anomalous. For achieving both tasks a good idea was too summarize the vectors using the Mean Pooling and the Max Pooling. These two values, combined together, can be seen as the Data Description (DD) for each time series. The final step uses LOF in order to detect anomalies in the Data Description.

### A. Data Processing

In the exploratory paper [1] the authors found different behaviours comparing the ratio of sent/received packets. I therefore decided to focus on this features in order to find anomalies. I therefore created new features in the data set:

$$SrcOnDstPackets = \frac{SrcPackets}{DstPackets} \tag{1}$$

$$SrcOnDstBytes = \frac{SrcBytes}{DstBytes} \tag{2}$$

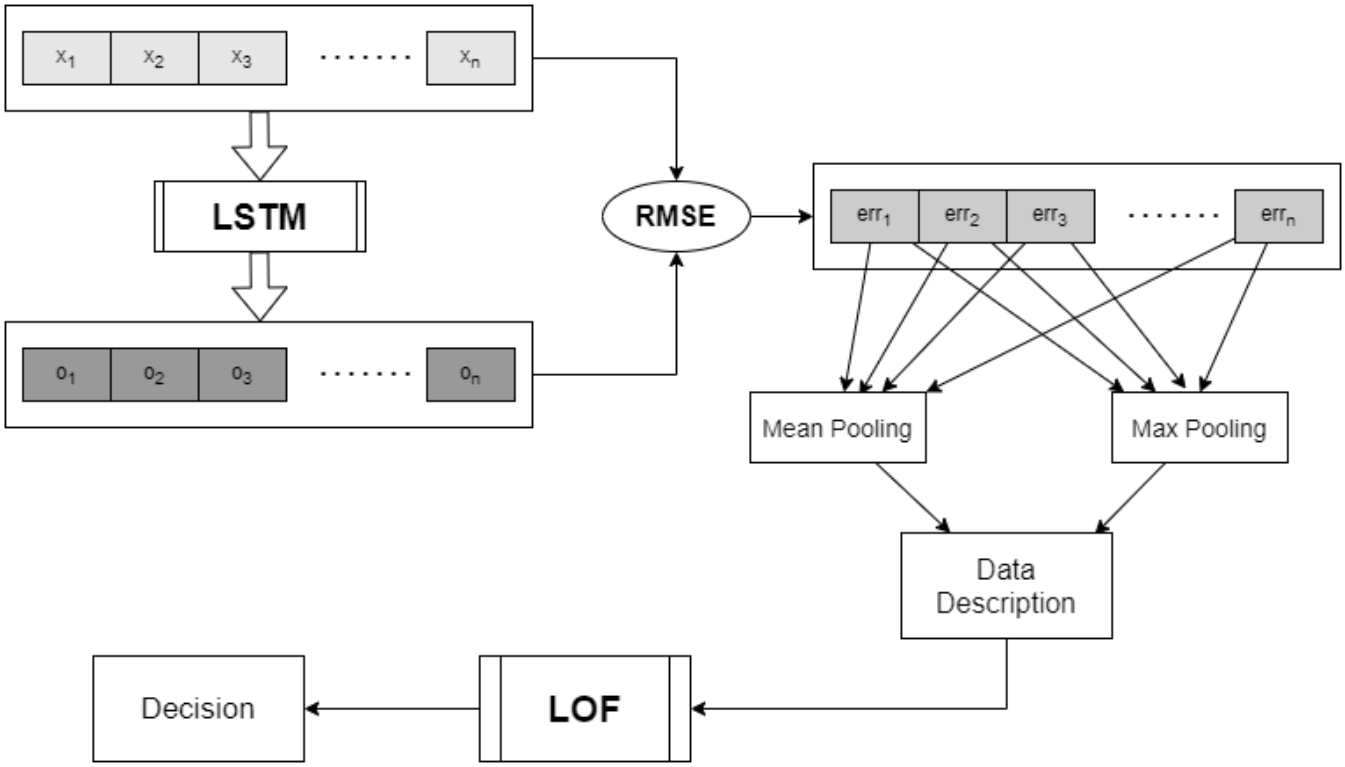


Fig. 1. Model Architecture.

Many other features could have been calculated as ratios of the initial ones. I decided to limit the complexity and to focus on the paper [1], but computing other interesting features may be useful to detect anomalies. Another advantage of calculating such ratios was to reduce the dimensionality of the feature space.

### *B. Scaling and Principal Component Analysis*

In order to support the LSTM training phase, I used a standard scaler on every feature. I then used the Principal Component Analysis (PCA) on the new computed features (1) and (2), in order to reduce the dimensionality. The choice of the features to process into the PCA and the final dimensionality may impact the performance and the training time of the LSTM.
I choose to process only SrcOnDstPackets and SrcOnDstBytes, and to reduce the dimensionality to 1 in order to simplify the LSTM training.

### *C. Time Series Extraction and Padding*

I extracted for each source device its own time series. For each time series, the time parameter was normalized as a step value. The only feature taken into account were the ones calculated by the PCA.
Each time series had its own length, and this was a problem because the only way to put them into a LSTM was to use the Stochastic Gradient Descent (SGD) setting the batch size to one. This method would have been too expensive for the LSTM, so I decided to choose a fixed input length. Shorter time series were pre-padded with zeros, longer time series were trimmed to the chosen length.
The choice of the length is an operation that has to be tuned: short values will long some time series to lose information; large values will force short time series to get almost only zeros due to the padding process. I decided to fix the length to 600 to make the the LSTM training phase feasible.

### *D. LSTM Training Phase*

Long Short Term Memory (LSTM) is a recurrent neural network architecture suitable to manage time series. I therefore built a simple LSTM network and I trained it using the time series previously pre-processed. Using regularizing techniques the LSTM was capable to learn recurrent pattern in network traffic without over-fit on training data. The training phase entails a grid search over hyper-parameters in order to obtain a model that performs well on every sample. Such tuning process was not performed properly due to lack of high computational power, time and connection stability. I therefore trained the LSTM to provide a reasonable degree of accuracy, but better result might be obtained.

### *E. LSTM Prediction Phase*

The trained LSTM is now capable to make predictions. The time series X are then the input of the LSTM that will output a new time series O. The vector Err is the Root Mean Squared Error (RMSE) between X and O, as we can see in “Fig. 2”.

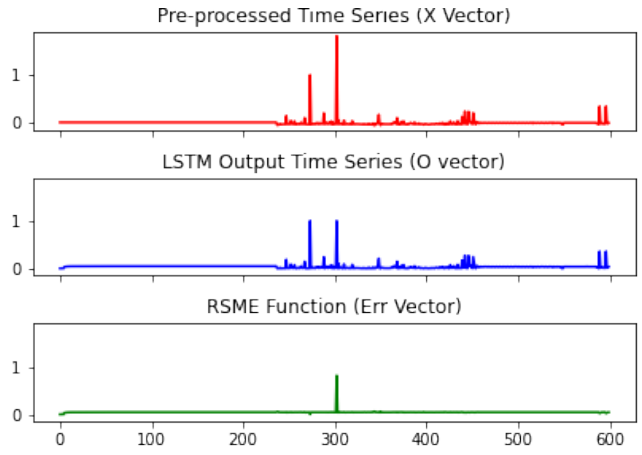


Fig. 2. The first plot shows a single time series X; the second plot shows the LSTM output time series O; the third plot shows the RMSE vector between X and O.

### F. Data Description

To detect anomalies in every sequence or between series, I had to summarize the information provided by the Err vectors. I took into account two kinds of anomalies, which can be described by two kinds of behaviours in the Err function: the first one is an anomalous time series, which will have a constant high error function; the second one is an anomalous instant in a time series, where a single point has a large Err value, like in "Fig. 2". In order to detect such anomalies I opted to summarize the Err vectors in Data Description scalars, computed by the mean pooling and the max pooling. The mean pooling computes the mean of the Err function and the max pooling computes the max of the Err function. For example the Data Description of the time series shown in "Fig. 2" are (0.04496, 0.81806). This process is computed over all the time series. In "Fig. 3", we can see the effect of the mean pooling and the max pooling over the time series.

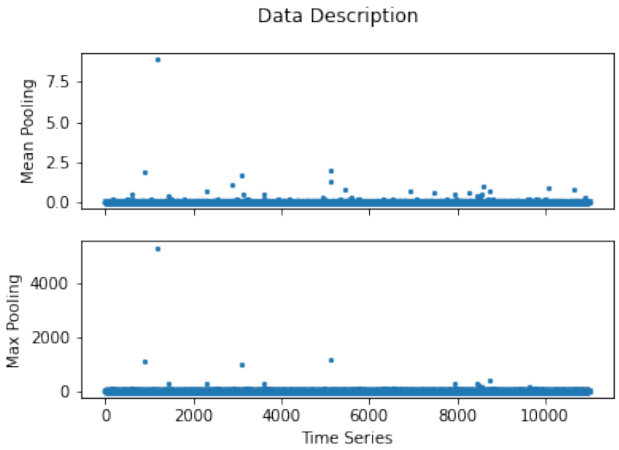


Fig. 3. The first plot shows the mean pooling value for each time series; the second plot shows the max pooling value for each time series.

In "Fig. 4" it's shown the combination of data description. The dimensions are scaled in order to make these value more suitable for the LOF process. The plots have a different level of detail in order to show properly the data description.

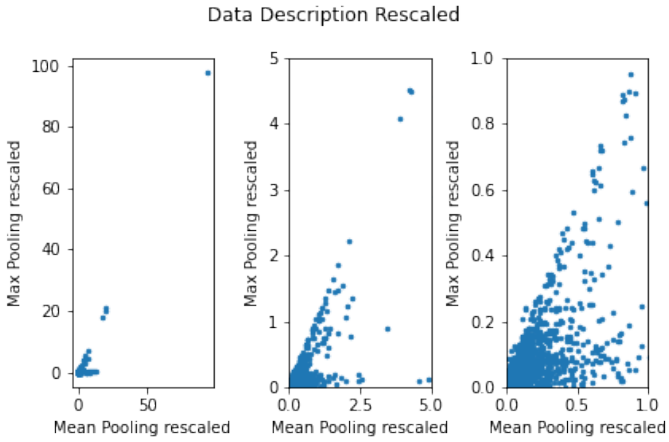


Fig. 4. The three plots represent the data description of the time series. The plots are zoomed in order to show properly the outliers.

### G. LOF Decision Process

As we can see in "Fig. 4" there are many outliers which can be detected applying the Local Outlier Factor (LOF) algorithm. I then used a LOF function with the number of neighbours parameter equal to 30. The choice of this parameter has been made visualizing the plots. It can be further tuned using a posterior process capable to validate the anomalies. In "Fig. 5" the anomalies are colored purple. The time series taken into account are 11000, and LOF found 365 anomalies: 3.31% of the connections.

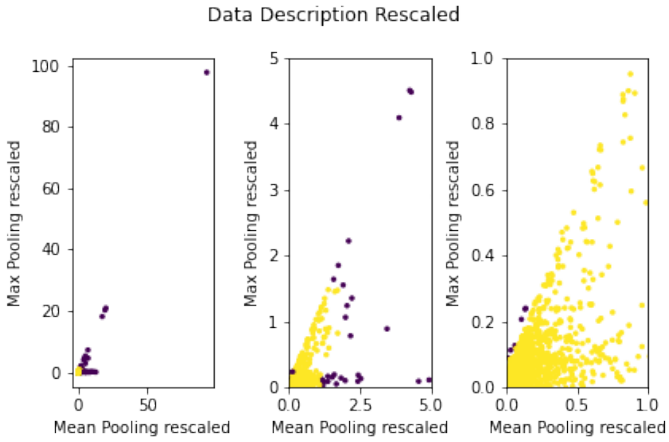


Fig. 5. In these plots anomalies are colored purple; normal samples are colored yellow.

## III. Conclusion and Future Work

In this paper I propose an approach combining LSTM, data description model and LOF to detect anomalous network traffic, analyzing the Unified Network Dataset. This is an

unsupervised learning method that needs a ground truth for being validated and to detect false positives or true negatives. This model needs a sequence of different steps. Each of these steps can be tuned properly to improve the global performance of the model:

- **Data Pre-processing** Data may be pre-processed in a different way with different methods.
- **LSTM Input Length** The choice of the input length may be an hyper-parameter of the LSTM, that can be optimized performing a grid search.
- **LSTM Performance** The choice of the hyper-parameters must be optimized in order to increase the performance without overfitting. This can be done performing a grid search.
- **Data Description** In this paper I used mean and max pooling to represent the time series. Other values may improve the performance of the model.
- **LOF Performance** In order to improve the performance, LOF hyper-parameters may be tuned in order to detect properly the anomalies and to avoid mistakes.

Such analysis needs large computational costs and a lot of time to be performed, but It may allow the model to be a good anomaly-based network intrusion detection system.

## REFERENCES


[1] Catherine Beazley, Karan Gadiya, Rakesh Ravi K U, David Roden, Boda Ye, Brendan Abraham, Donald E. Brown, Malathi Veeraraghavan, "Exploratory Data Analysis of a Unified Host and Network Dataset ," University of Virginia.

[2] Nguyen Thanh Van, Tran Ngoc Thinh, Le Thanh Sach, "A COMBINATION OF TEMPORAL SEQUENCE LEARNING AND DATA DESCRIPTION FOR ANOMALYBASED NIDS", International Journal of Network Security & Its Applications (IJNSA) Vol. 11, No.3, May 2019.

[3] M. Turcotte, A. Kent and C. Hash, "Unified Host and Network Data Set", in Data Science for Cyber-Security. November 2018, 1-22.