Deep Autoencoder Ensembles for Anomaly Detection on Blockchain

Francesco Scicchitano[1], **Angelica Liguori**[1], Massimo Guarascio[1], Ettore Ritacco[1] and Giuseppe Manco[1]

[1] **Institute for High Performance Computing and Networking of the Italian National Research Council (ICAR - CNR), Italy**

ISMIS 2020

Outline

Scenario

- Identify anomalies in blockchain networks, i.e. attacks on blockchain
    - These represent extremely rare events and typically do not share common patterns

- An anomaly detection system based on a deep autoencoder ensembles is proposed

Problem definition

- **Data:** a sequence of events observed in a time window
  - *Each event* is represented by a feature vector
- **An anomaly** is a feature vector which deviates from the normality
- **Goal:** identification of deviances

Methodology

- Adapts the snapshot ensemble method to a sequential unsupervised scenario exploiting an encoder-decoder model as model base
  - replicate an input sequence by producing a reconstructed copy from the compressed representation of the input
  - difference between the input and reconstructed sequences represents the anomaly score

- Several base models have been instantiated, whose outlierness score is finally averaged

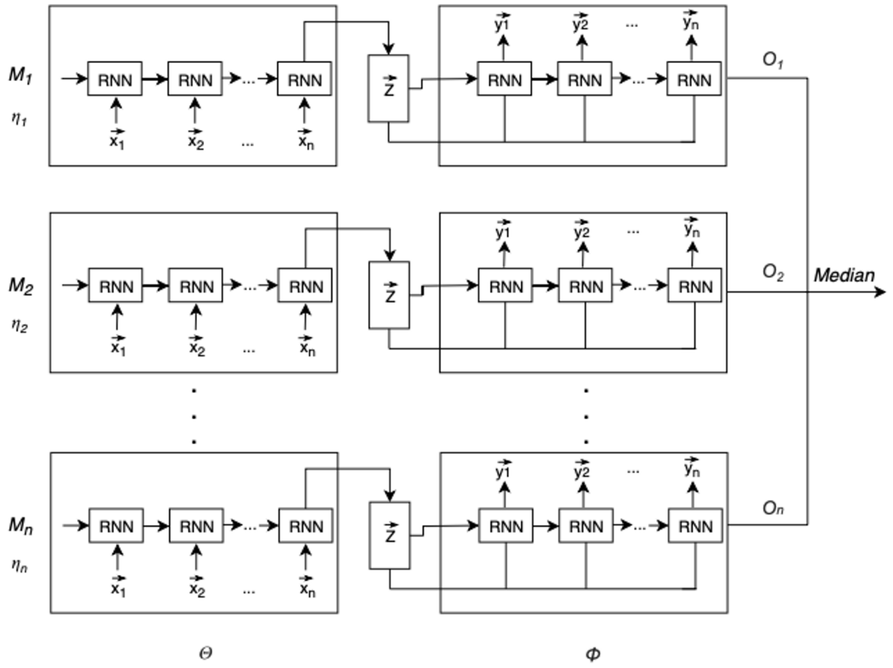
Encoder-Decoder Model

- The model is composed by:
  - Encoder that maps the original input *x* (a sequence of events observed in a time window) into a latent space, producing an embedding *z*
  - Given *z*, the decoder aims at generating an output *y* as close as possible to original input (*x*)

- Encoder and Decoder are modeled as Recurrent Neural Networks (RNNs)
  - Long Short-Term Memories (LSTM) is used

Snapshot Ensemble Encoder-Decoder (SEED)

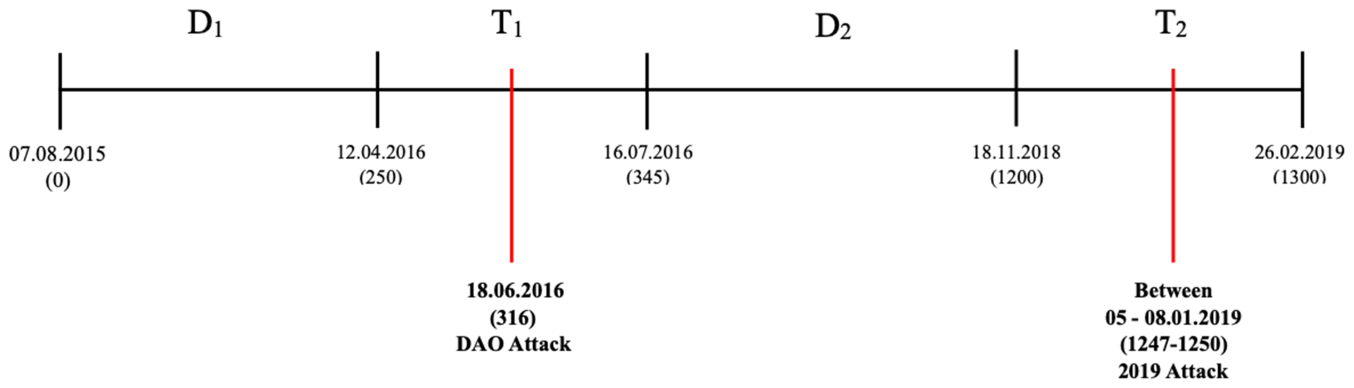
- Idea:
  - An autoencoder is randomly initialized devised as M_1
  - Then, the procedure iteratively learns model M_i by re-training M_{i-1} with the initial learning rate $\boldsymbol{\eta}$ for a fixed number of epochs
  - At each epoch $\boldsymbol{\eta}$ is progressively lowered
  - M_i is then collected in the ensemble and the learning rate is reinitialized



Study case

- SEED model is applied to the Ethereum Classic (ETC) blockchain to identify attacks
- Dataset is available on Kaggle
- Sample of ETC blockchain spanning over a period of four years (July 2015 - July 2019)
- It has experienced two (known) successful attacks: DAO (18 June 2016) and 51% (5-8 January 2019)

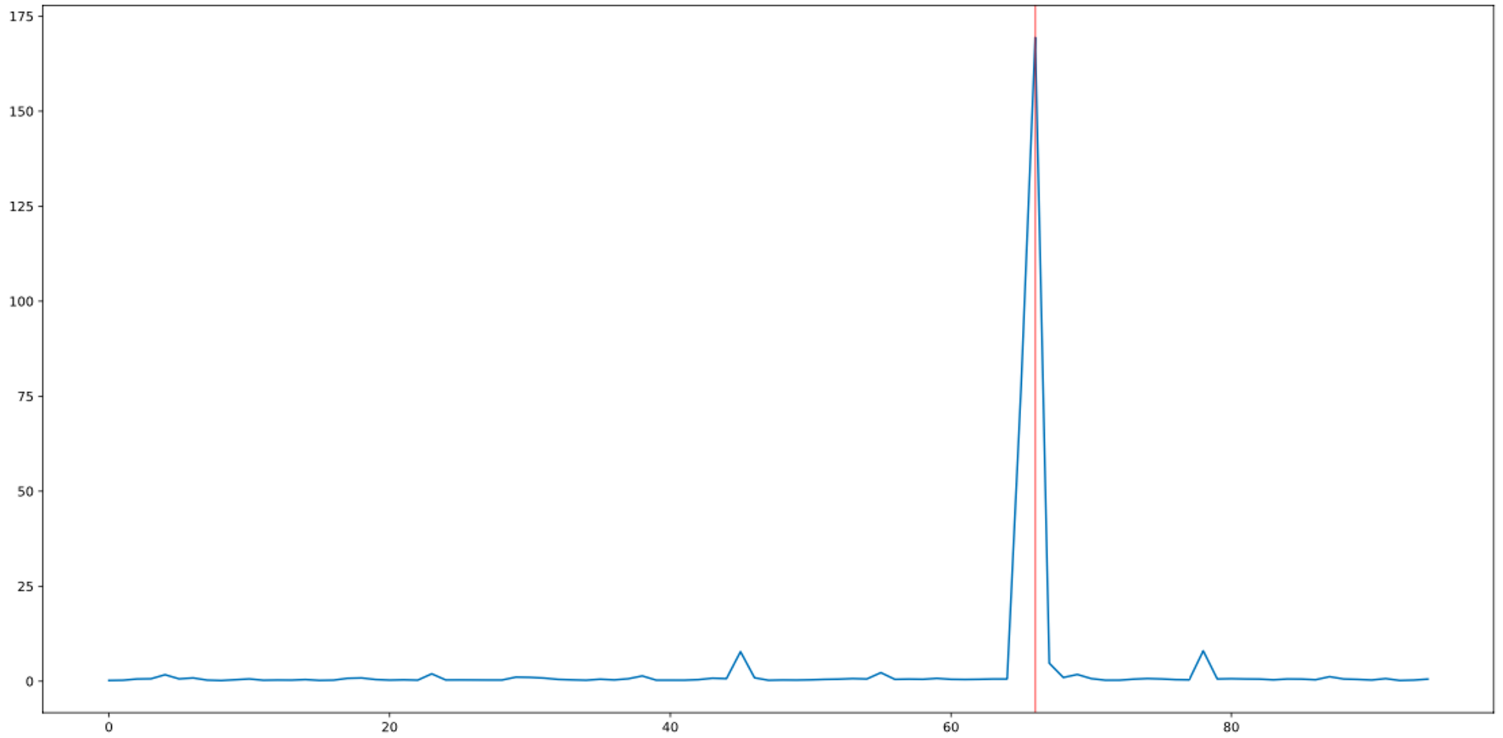
Evaluation protocol



- Evaluation steps:
  1. we train model on D_1 and test it on T_1 (DAO attack)
  2. we train model on D_1 and test it on T_2 (51% attack)
  3. we train model on $D_1 \cup D_2$ and test it on T_2 (51% attack)

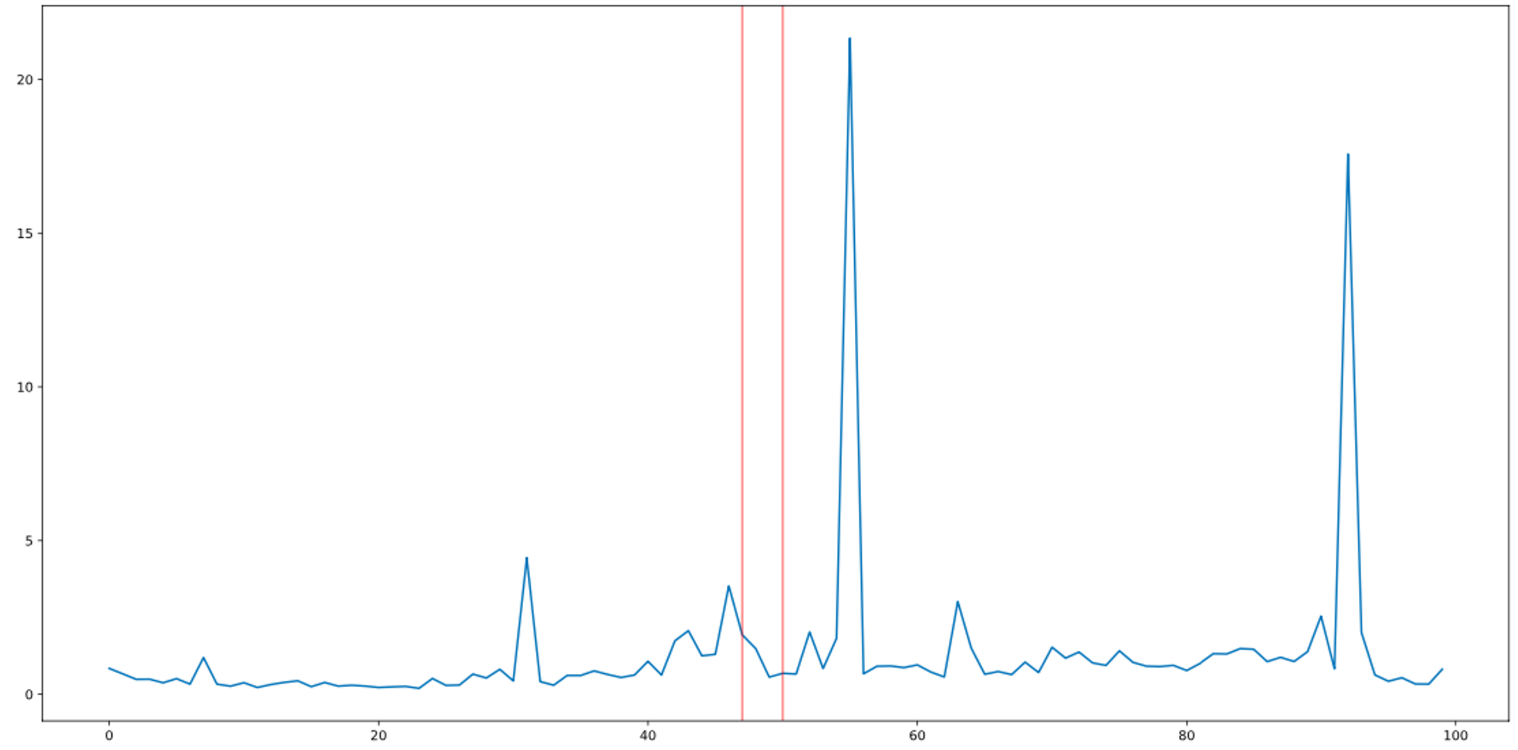
Experiment 1: DAO attack

D_1

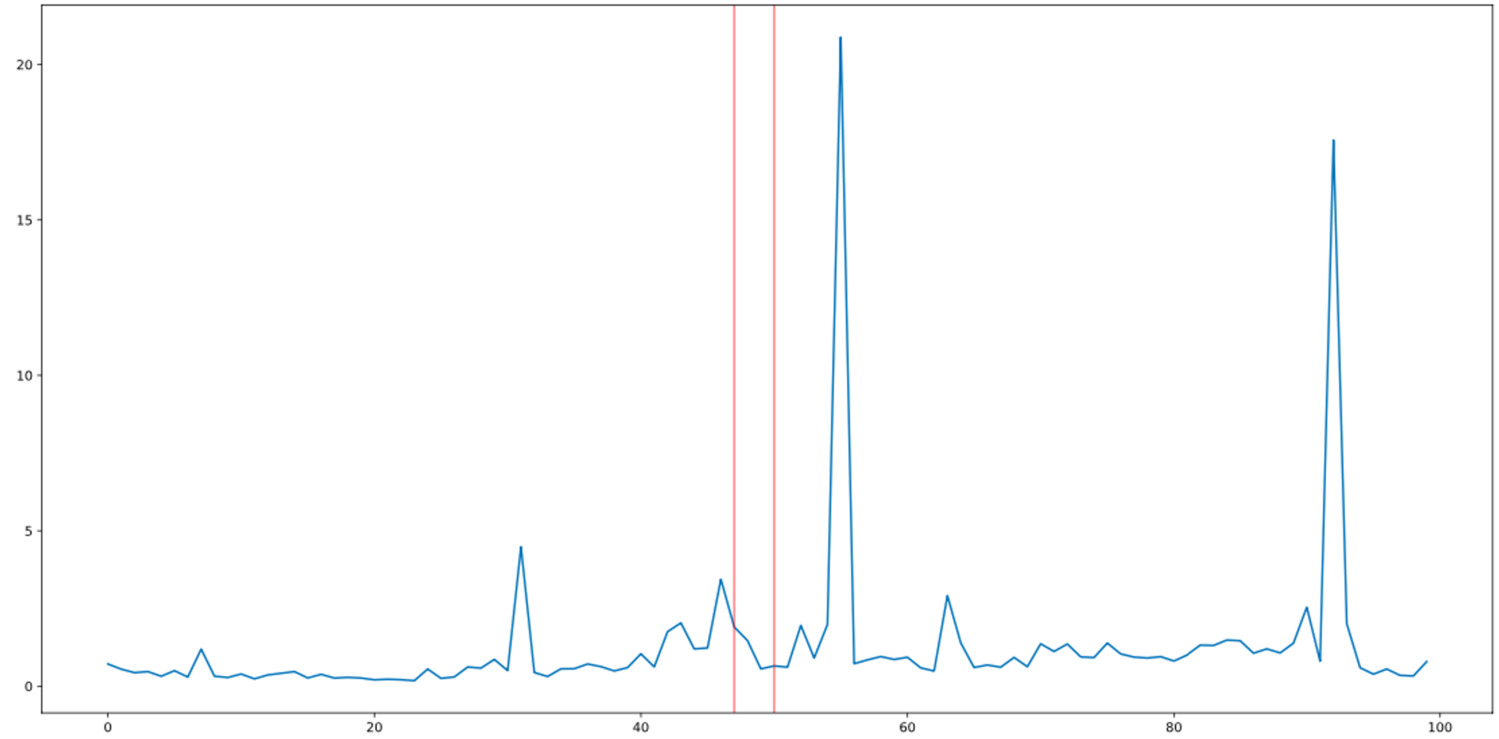


Experiment 2: 51% attack

D_1



Experiment 3: 51% attack

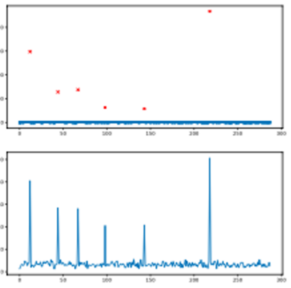


$D_1 \cup D_2$

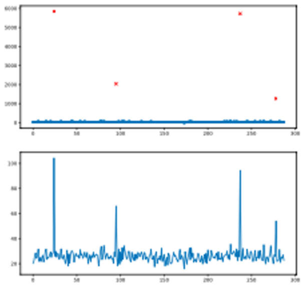
Experiments on Synthetic Data

- Sensitivity analysis of SEED in a controlled scenario
- Data were generated according to the following procedure:
  - produce a sequence D (is a matrix whose rows represent the sequence events and the columns represent the feature for each event). Each element is generated from a fixed gaussian distribution
  - randomly select *n* points and for each point select a feature of relevance
  - Inject uniform noise in the chosen feature

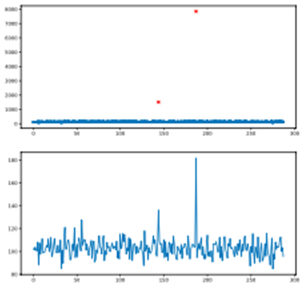
Experiments on Synthetic Data



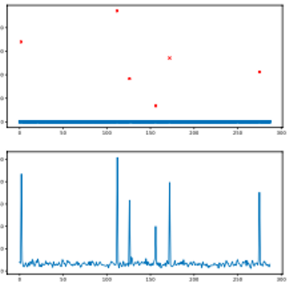
(a) $n = 1\%,\ nFeat = 8$



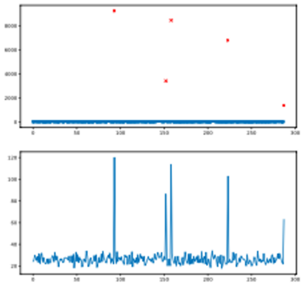
(b) $n = 1\%,\ nFeat = 32$



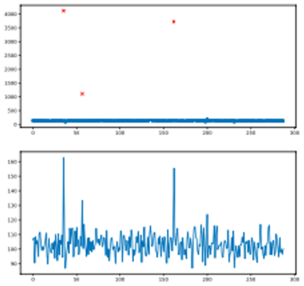
(c) $n = 1\%,\ nFeat = 128$



(d) $n = 3\%,\ nFeat = 8$



(e) $n = 3\%,\ nFeat = 32$



(f) $n = 3\%,\ nFeat = 128$

Conclusions

- In this work we proposed an unsupervised ensemble deep architecture for anomaly detection
- We evaluate the model on both real data, the ETC blockchain, and synthetic ones
- Experiments prove our model capability to effectively detect attacks
- We plan to study effective strategies for selecting the best weak learners during the learning phase to improve the detection capabilities in both unsupervised and supervised scenarios

Thank you for your attention!