Preliminary study on AI methods for cybersecurity threat detection in computer networks based on raw packets data

WMLQ 05.06.2024

Aleksander Ogonowski, Michał Żebrowski, Arkadiusz Ćwiek

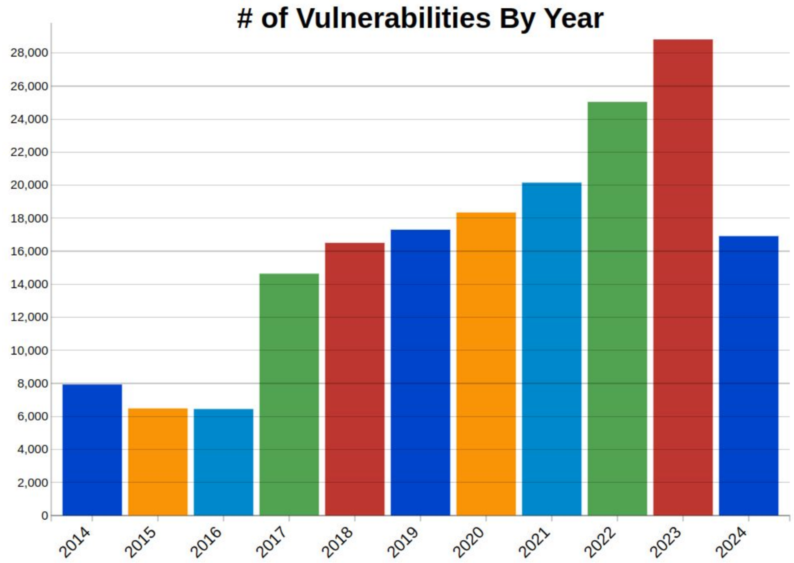
Motivations

EURO²

- Simplify real time network monitoring,
- Curiosity of the deep learning methods performance in Intrusion detection systems (IDS).

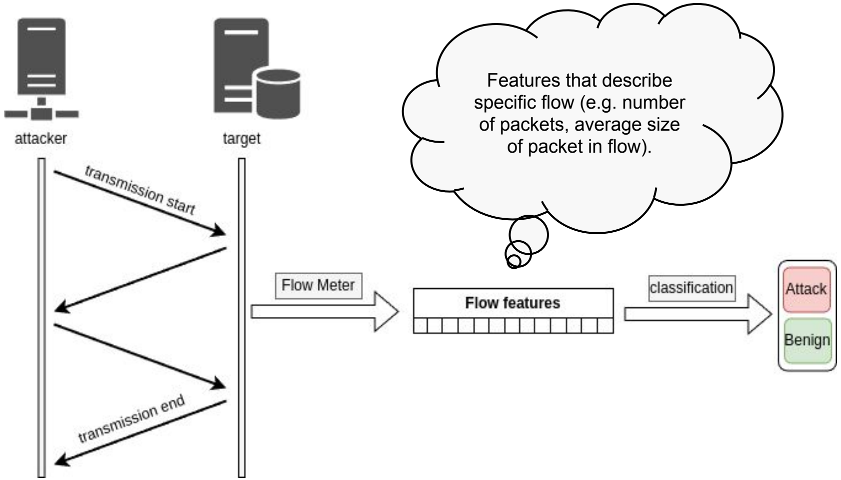
"A weakness in the computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or availability."

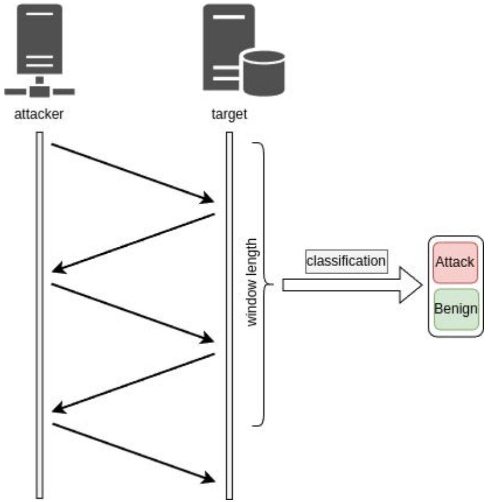
```
# notice_ssh_guesser.zeek

@load protocols/ssh/detect-bruteforcing

redef SSH::guessing_timeout = 30 mins;
redef SSH::password_guesses_limit = 10;

hook Notice::policy(n: Notice::Info)
    {
        if ( n$note == SSH::Password_Guessing )
            add n$actions[Notice::ACTION_LOG];
    }
```

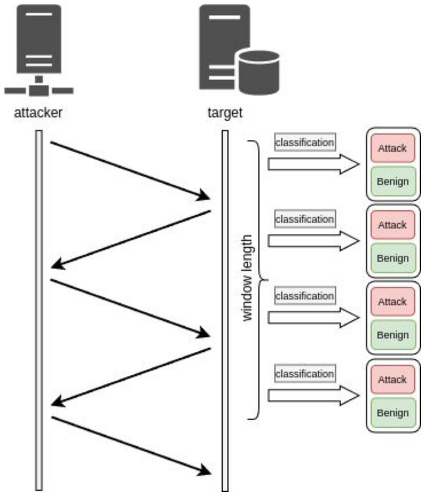


source: https://nvd.nist.gov/vuln/search/statistics?form_type=Basic&results_type=statistics&search_type=all&isCpeNameSearch=false

Types of classification







flows classification

- based on flow features
- most popular solution

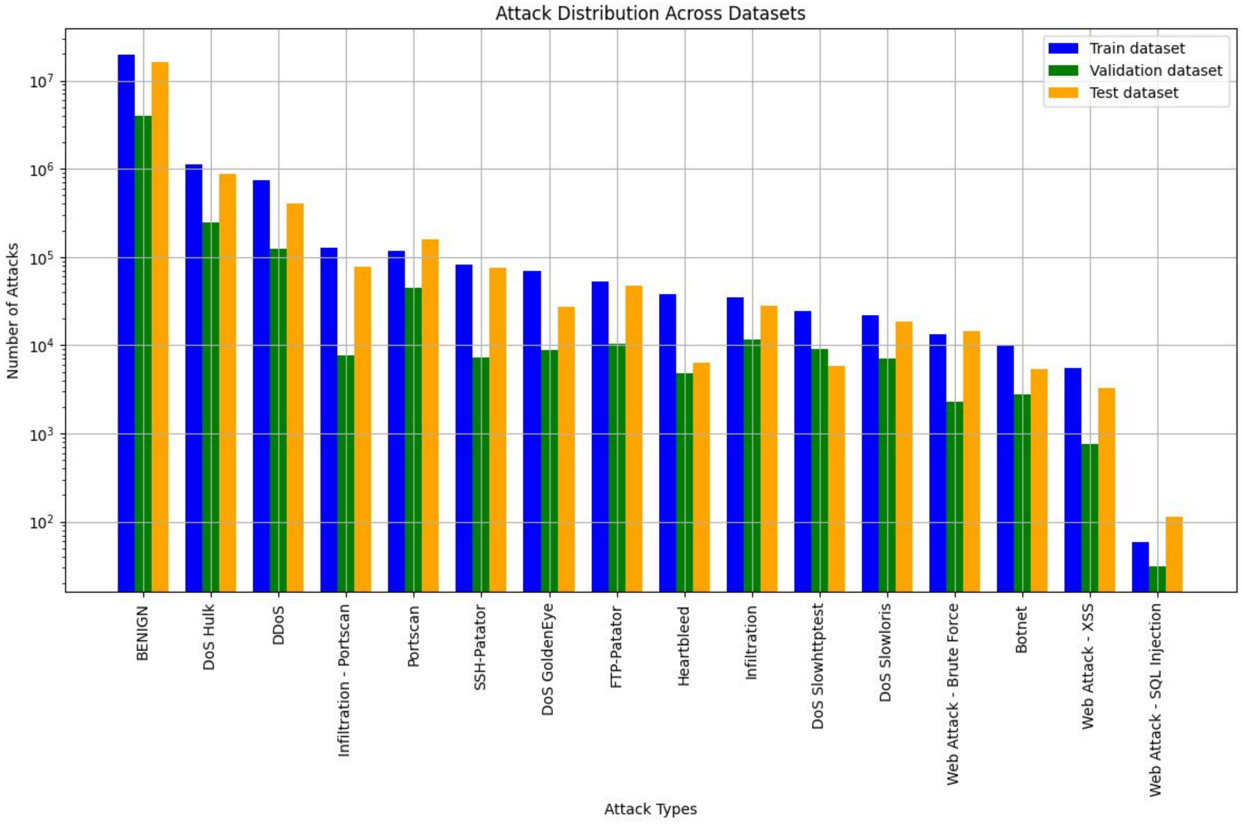
windows classification

- based on packets
- packets can be mixed within many flows
- real time monitoring

packets classification

- based on packets
- packets can be mixed within many flows
- real time monitoring
- the chosen solution

Attack types in CIC IDS 2017



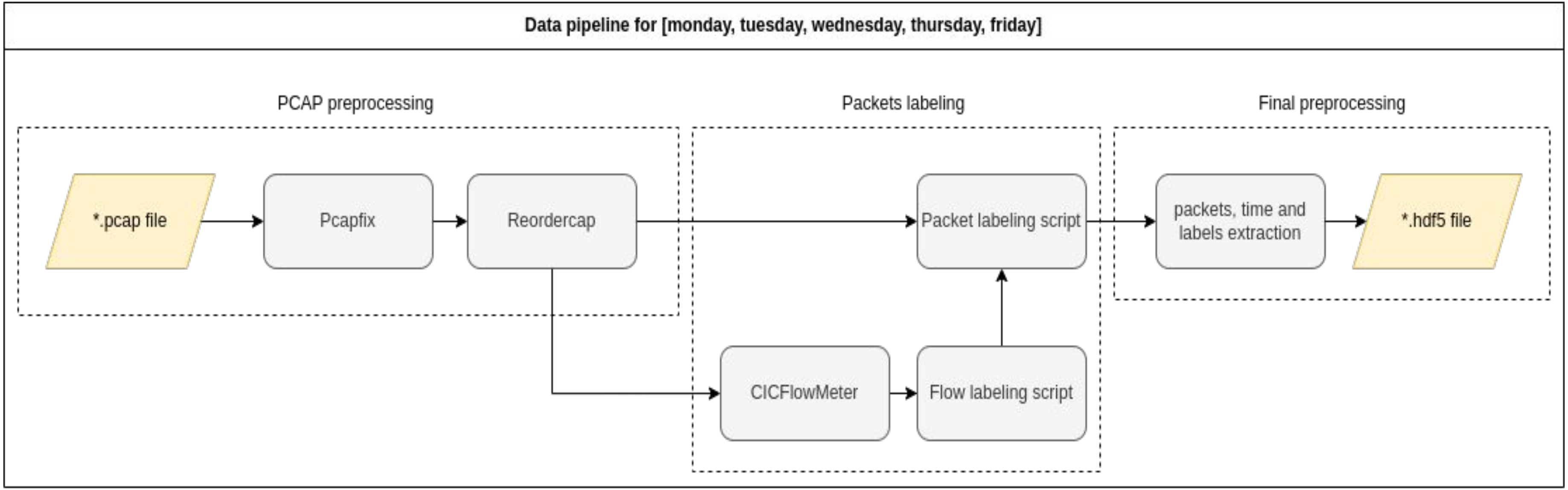
- over 50 GB of raw traffic data
- 5 days
- 15 types of attacks + normal traffic
- files
  - *.pcap - raw traffic data
  - *.csv - flow features + labels

- dataset split:
  - training set: 50%,
  - validation set: 10%,
  - test set: 40%.

- Benign packets in
  - train dataset: 88.96%
  - validation dataset: 89.04%
  - test dataset: 90.21%

Centrum Informatyczne Świerk
Świerk Computing Centre

Data preprocessing pipeline



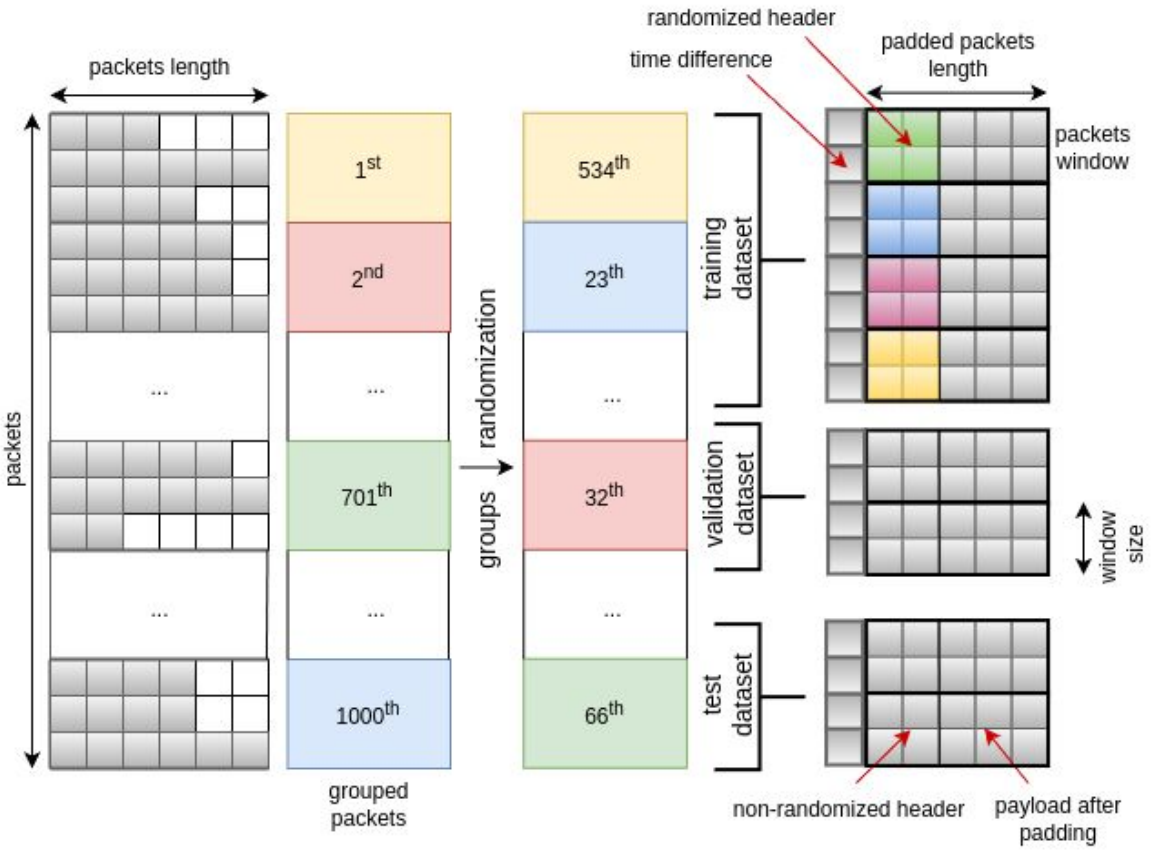
Related works using CIC-IDS-2017 dataset

Related works on intrusion detection using CIC-IDS-2017 dataset.

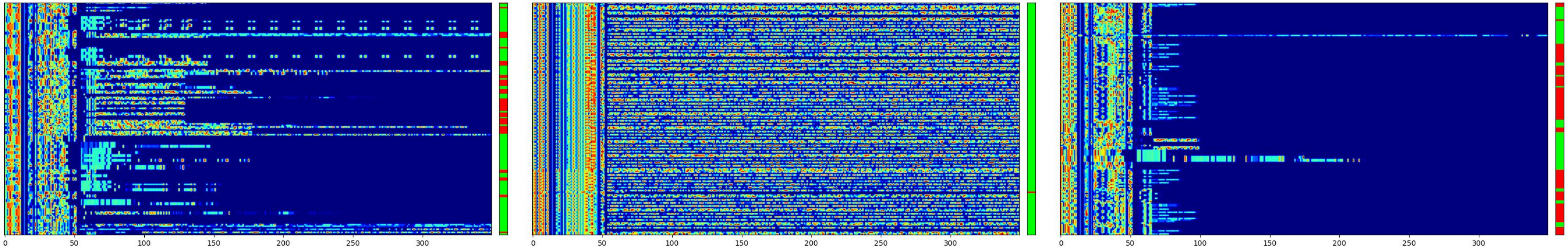
Method	Accuracy [%]	Recall [%]	Precision [%]	Input Type	Classification (of)	Dataset
RF [4]	99.99	99.99	99.99	Flow features	Flow	CIC IDS 2017
DCNN [7]	99.96	99.96	99.96	Flow features	Flow	CIC IDS 2017
ET [9]	99.95	99.95	99.95	Flow features	Flow	CIC IDS 2017
RF [9]	99.94	99.94	99.94	Flow features	Flow	CIC IDS 2017
DT [9]	99.91	99.91	99.91	Flow features	Flow	CIC IDS 2017
CNN [8]	99.61	95.00	97.05	Flow features	Flow	CIC IDS 2017
XGB [9]	99.65	99.65	99.65	Flow features	Flow	CIC IDS 2017
CNN-LSTM [5]	99.48	99.69	99.25	Flow features	Flow	CIC IDS 2017
EP-FCNN [1]	99.50	-	-	Flow features	Flow	CIC IDS 2017
CNN-LSTM [3]	99.78	-	-	Flow features	Flow	CIC IDS 2017
CNN [3]	99.23	-	-	Flow features	Flow	CIC IDS 2017
EP-CNN [1]	98.80	-	-	Flow features	Flow	CIC IDS 2017
DT [2]	98.80	97.30	-	Flow features	Flow	CIC IDS 2017
EP-LSTM [1]	98.60	-	-	Flow features	Flow	CIC IDS 2017
DBN [3]	98.59	-	-	Flow features	Flow	CIC IDS 2017
SVM [3]	98.20	-	-	Flow features	Flow	CIC IDS 2017
LSTM [8]	97.67	95.95	94.96	Flow features	Flow	CIC IDS 2017
DNN [8]	90.61	84.60	80.85	Flow features	Flow	CIC IDS 2017
DID (LSTM) [6]	-	**99.80**	**99.20**	**Packets frame**	**Packets frame**	**CIC IDS 2017**

*references can be found on the last slide

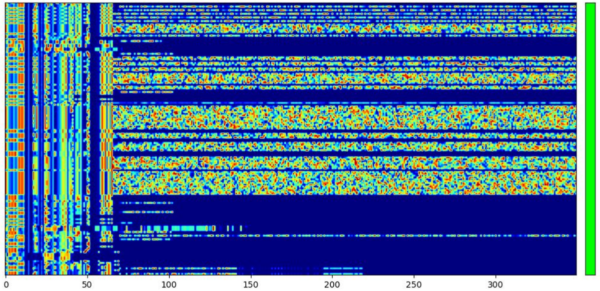
Packets preprocessing



Packets windows



- Examples of windows that contain packets marked as an attack.

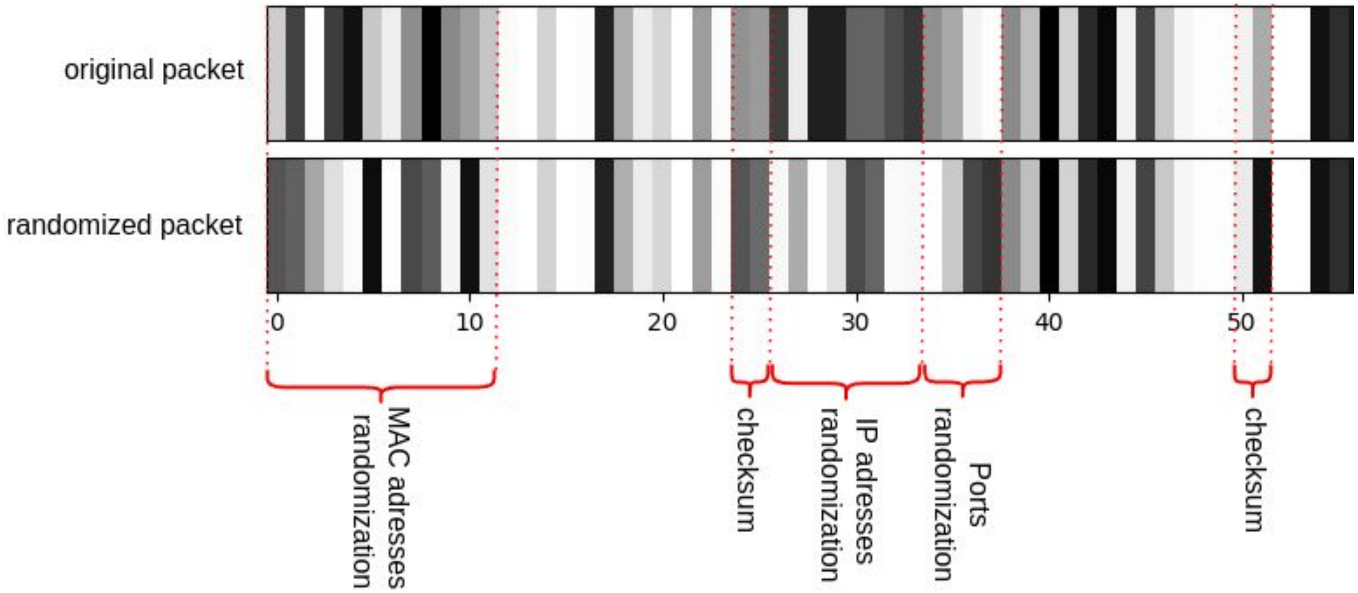


- Example of benign window.

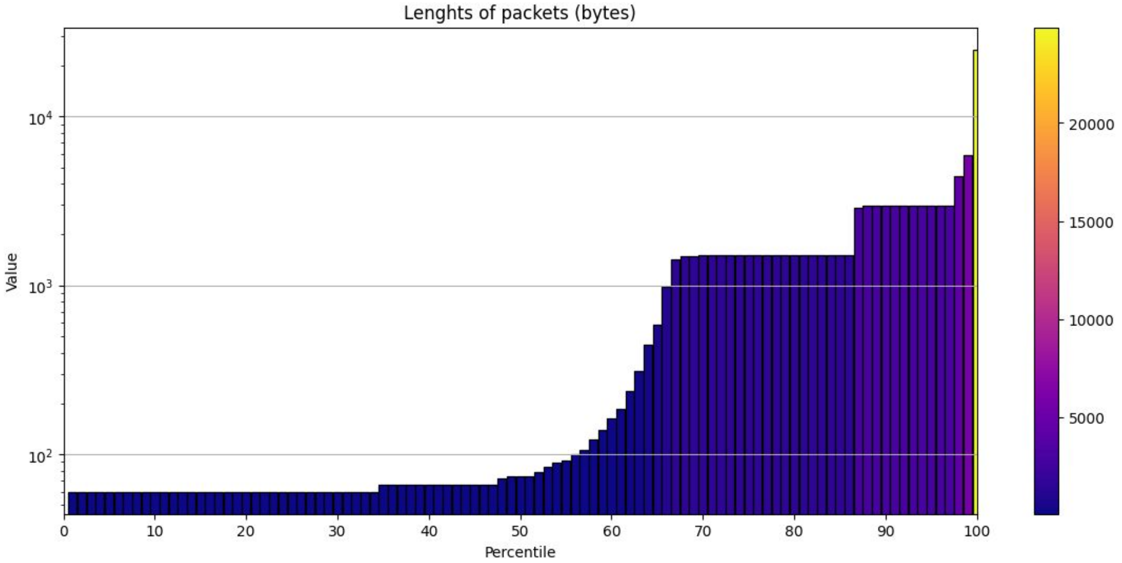
- ~20% of windows contain packets that are marked as an attack.
- Packets marked as an attack account ~10% of the dataset.
- Shorter packets are filled with zeros.

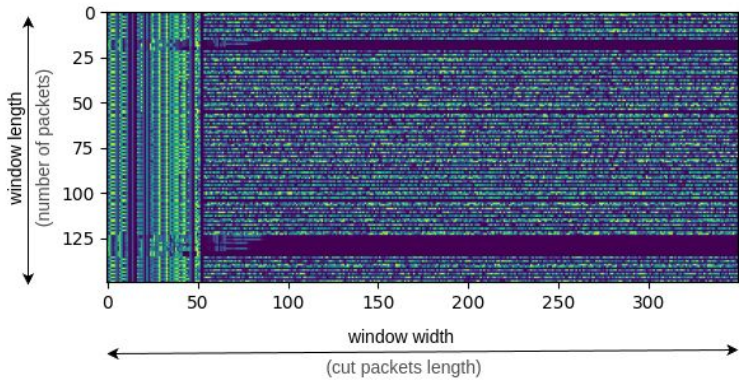
Packets randomization

- Model should not be adjusted to the specific data.
- Most of the other solutions assume cut out this particular parts of packet header.
- Randomization is done within each packets window - randomized replacement.
- Example below shows:
  - the window of a packet length,
  - the packet with TCP protocol (the most common).



Windows shape





- The maximum lengths of the packets and windows were limited by hardware.
- The lengths of the packets were selected based on the histogram of packet lengths:
  - the final selected value was 350 bytes.
- The length of windows were selected experimentally:
  - the final selected value was 150 packets.
- The FCNN receives a 1D input - window of 1 packet.
- We plan to implement dynamic window sizing in batches in the future.

Training and labeling

- Many types of deep learning algorithms were tested and developed.
- Four types of architectures were chosen as promising:
  - fully connected neural network (FCNN),
  - CNN-LSTM neural network,
  - CNN neural network,
  - pretrained EfficientNet-B0 neural network.

- Dataset balancing was tested:
  - oversampling windows with attack packets,
  - attack packets oversampling (FCNN).

- Two types of labelling were tested:
  - response from target to attacker labeled as an attack (Fig. 1),
  - only movement from attacker labelled as an attack (Fig. 2).

- Four cost functions were tested:
  - binary crossentropy (chosen),
  - focal loss,
  - dice loss,
  - IoU loss.

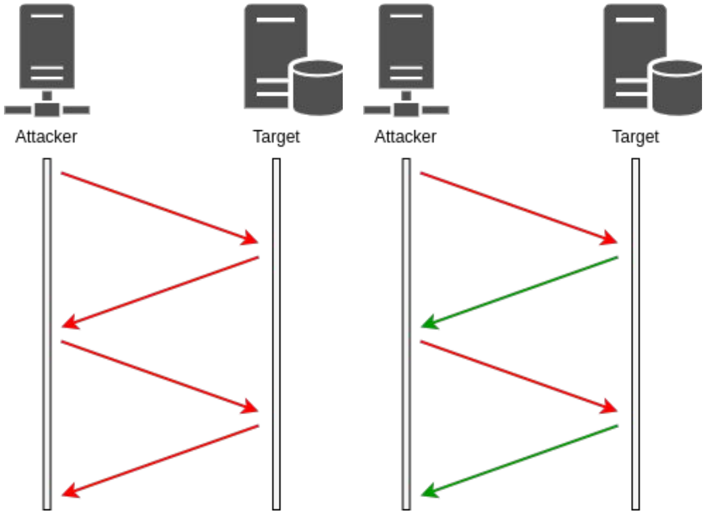
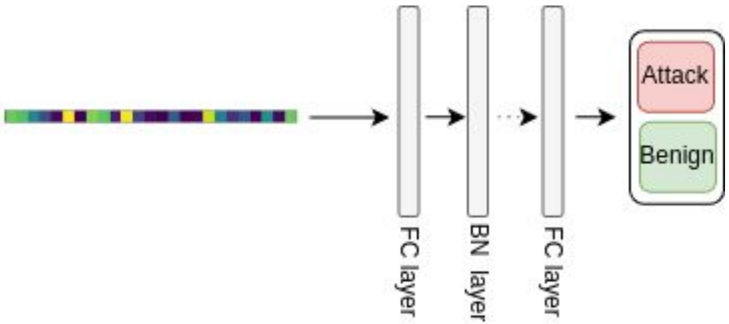


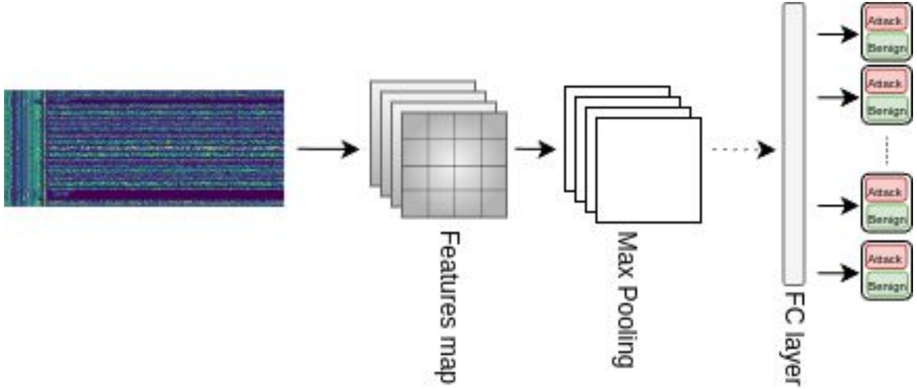
Figure 1 *Figure 2*

Deep learning architectures

- Fully connected neural network (*FCNN*):
  - **input 1D: 1 × 350+1,**
  - **output: 1**,
  - initial learning rate: 0.001,
  - optimizer: Adam,
  - batch size: 8096.

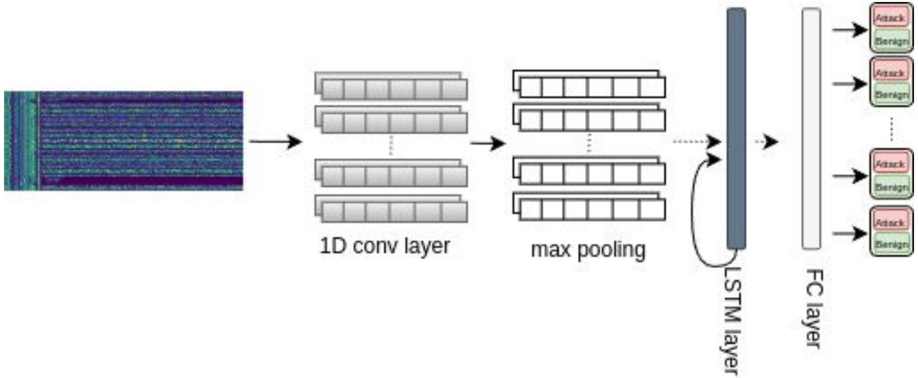


- Convolutional neural network (*CNN*):
  - **input 2D: 150 × 350+1,**
  - **output: 150,**
  - initial learning rate: 0.001
  - optimizer: Adam,
  - large convolutional filters,
  - batch size: 64.

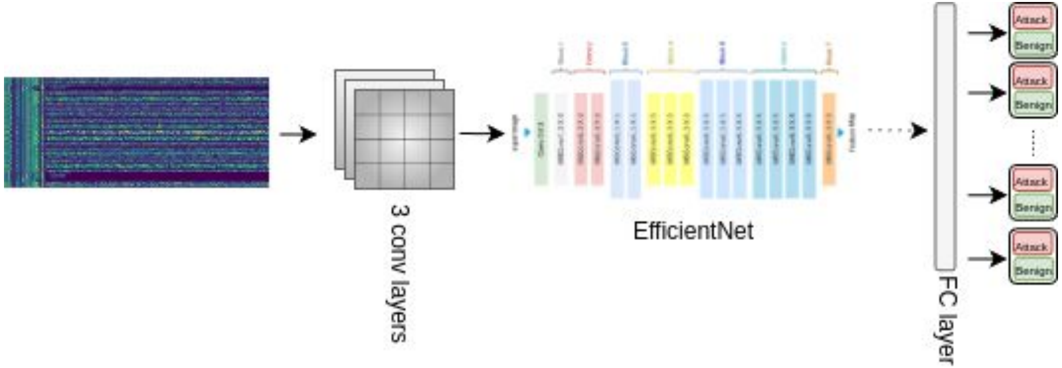


Deep learning architectures

- Hybrid neural network (*CNN-LSTM*):
  - **input 2D: 150 × 350+1,**
  - **output: 150,**
  - initial learning rate: 0.0005,
  - optimizer: Adam,
  - batch size: 64.



- *EfficientNet* based neural network:
  - **input 2D: 150 × 350+1,**
  - **output: 150,**
  - initial learning rate: 0.001,
  - optimizer: Adam,
  - pretrained on *imagenet,*
  - batch size: 16.

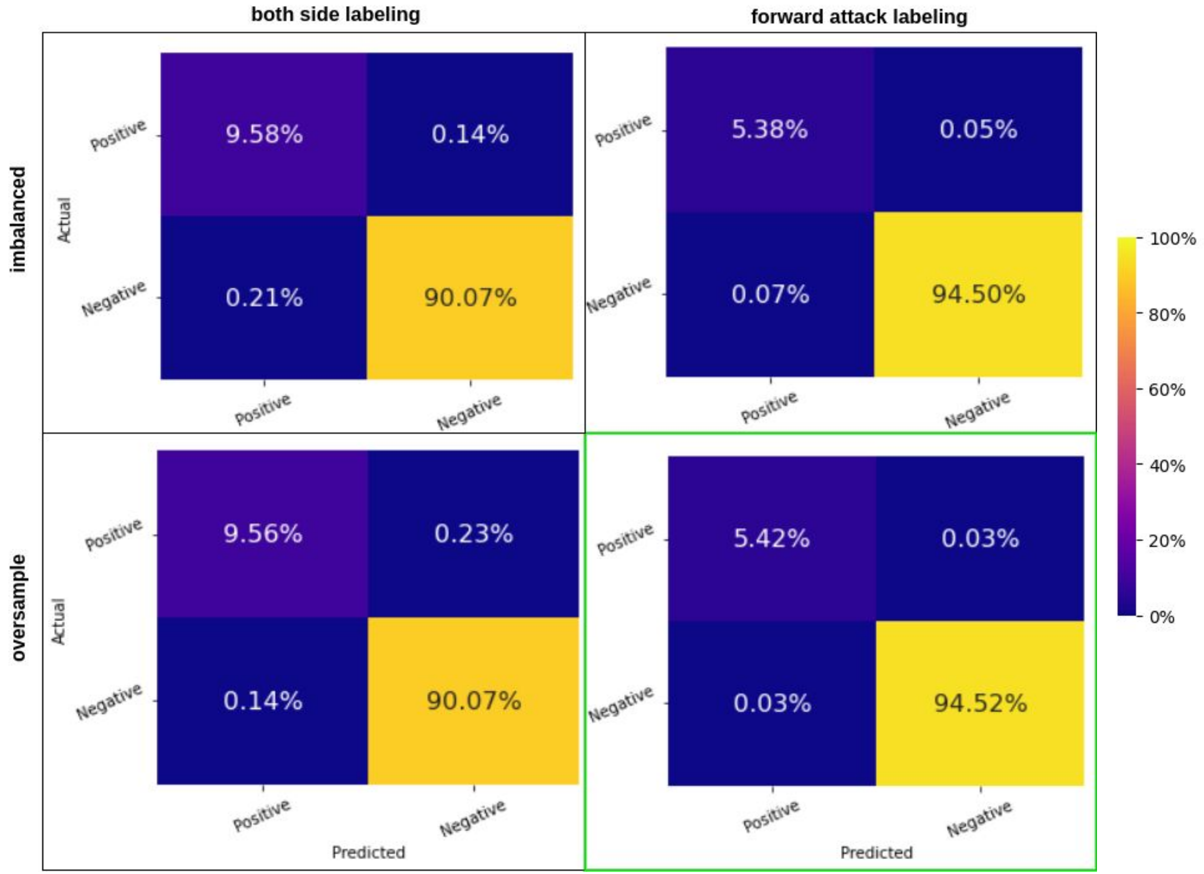


* *EfficientNet Architecture Source: https://www.researchgate.net/figure/Architecture-of-EfficientNet-B0-with-MBConv-as-Basic-building-blocks_fig4_344410350*

* *ImageNet: https://www.image-net.org/*

Results - Fully connected neural network

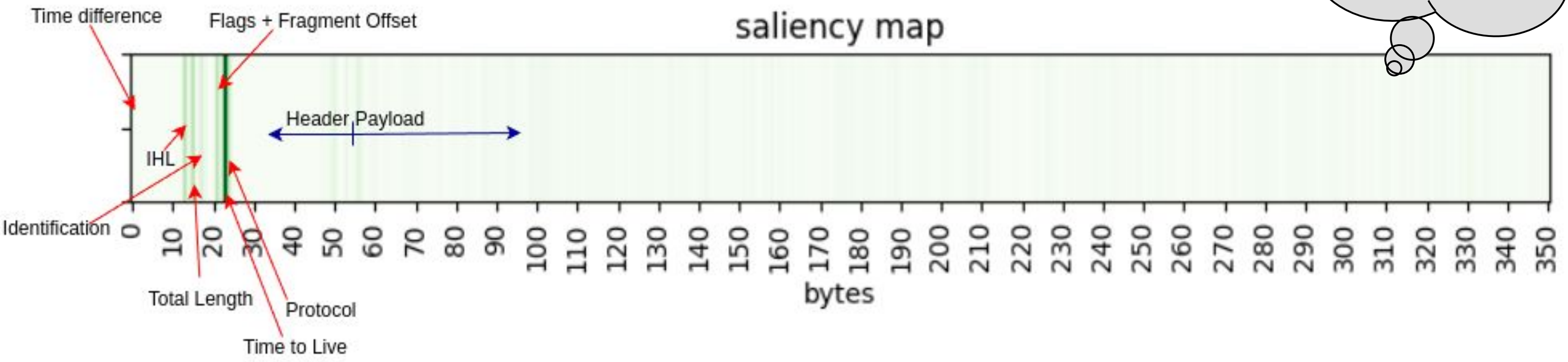
- Results on the test dataset
- Best results:
  - Binary Accuracy: 0.9993
  - Precision: 0.9941
  - Recall: 0.9837



Results - Fully connected neural network

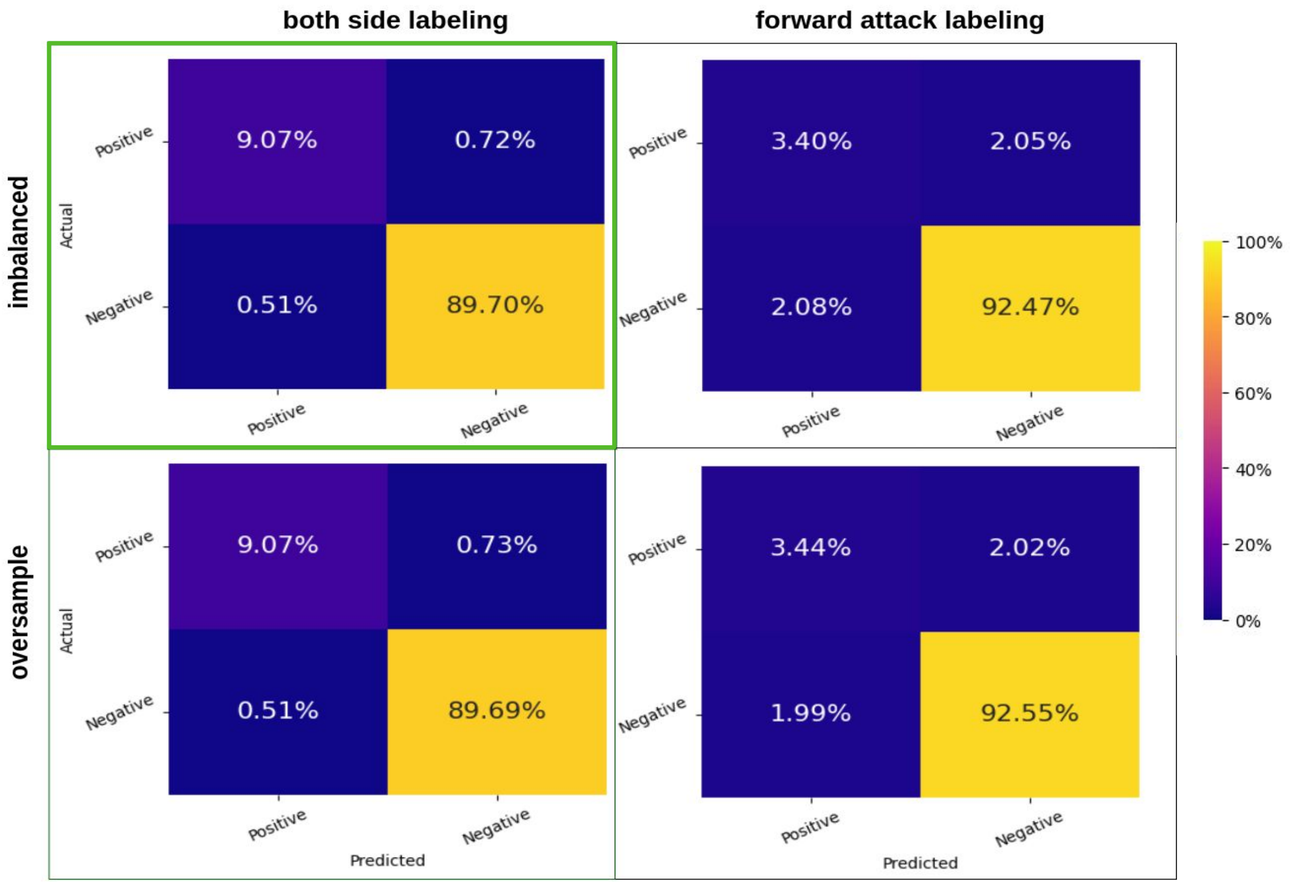
- Training loss history plot:
  - from the model with the highest accuracy,
  - epoch with best validation accuracy: 24.
- Saliency map
  - averaged over the entire batch.

Saliency map is used to identify features that influence the model's predictions. Color intensity is proportional to its importance.



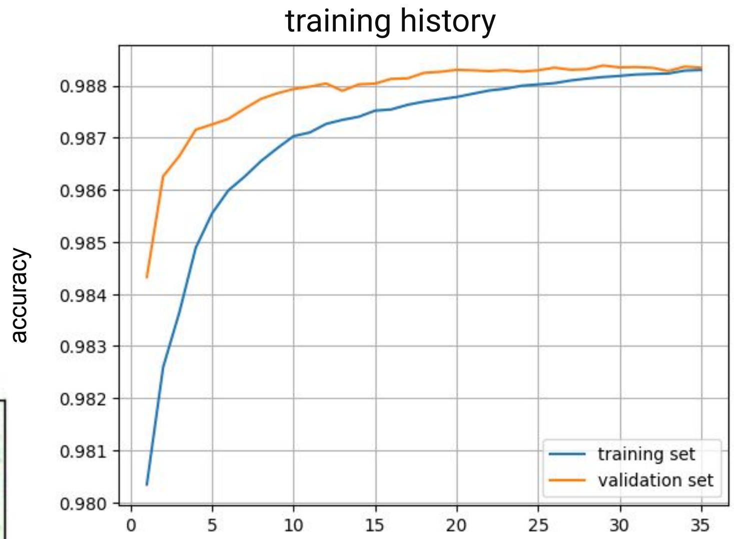
Results - Convolutional neural network

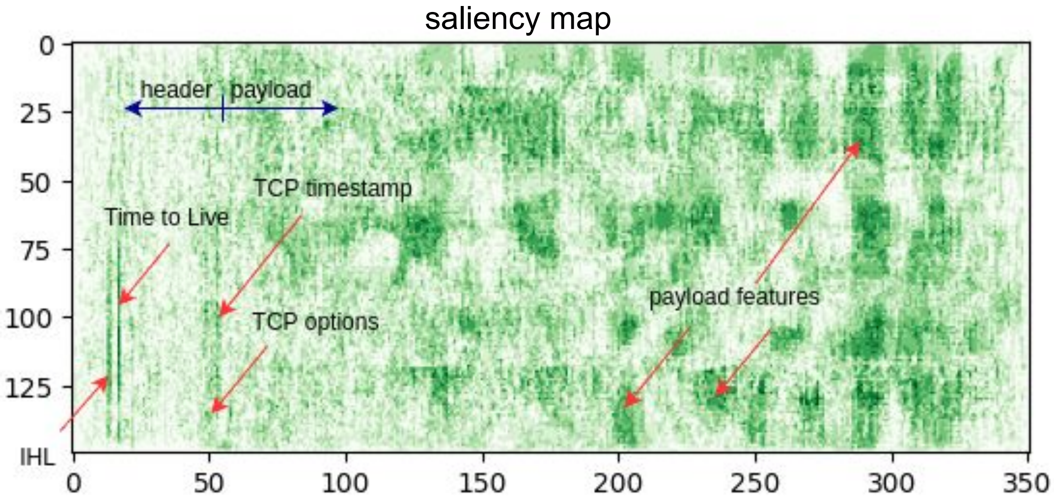
- Results on the test dataset
- Best results:
  - Binary Accuracy: 0.9877
  - Precision: 0.9466
  - Recall: 0.9265



Results - Convolutional neural network

- Training history plot:
  - from the model with the highest accuracy,
  - epoch with best validation accuracy: 29.

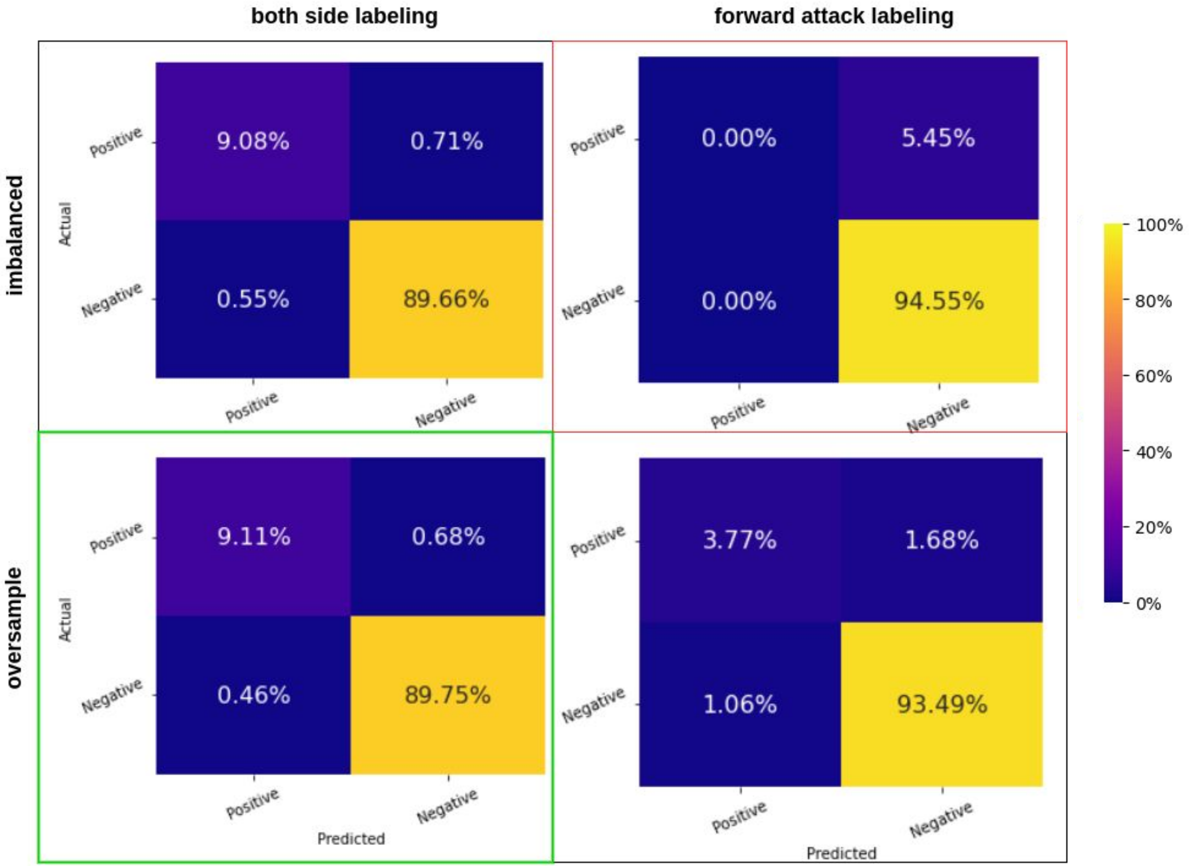




- Saliency map:
  - averaged over the entire batch.

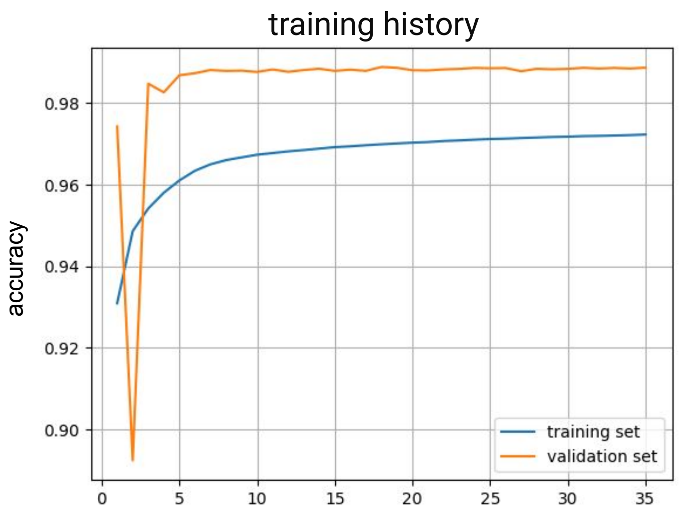
Results - Conv1D+LSTM neural network

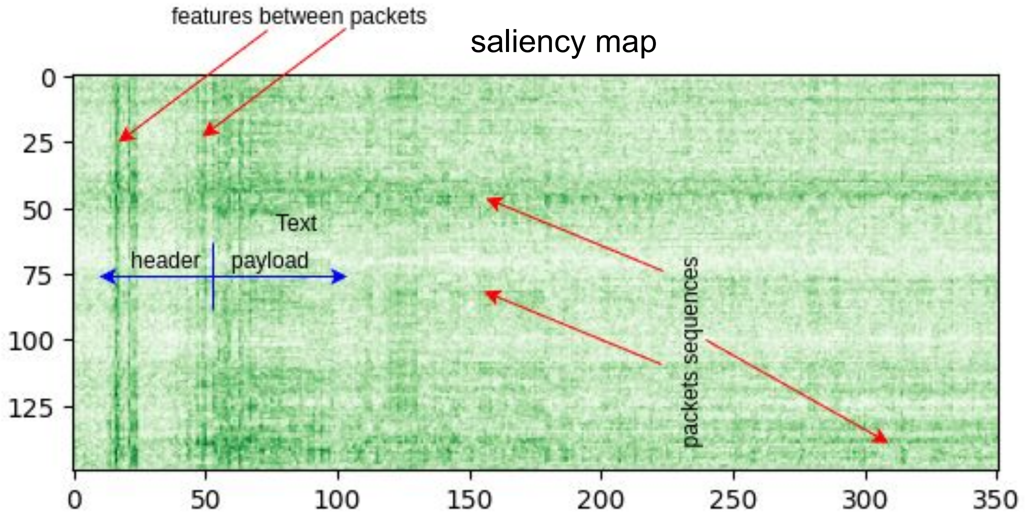
- Results on the test dataset
- Best results:
  - Binary Accuracy: 0.9885
  - Precision: 0.9518
  - Recall: 0.9301



Results - CNN+LSTM neural network

- Training history plot:
  - from the model with the highest accuracy,
  - epoch with best validation accuracy: 18.

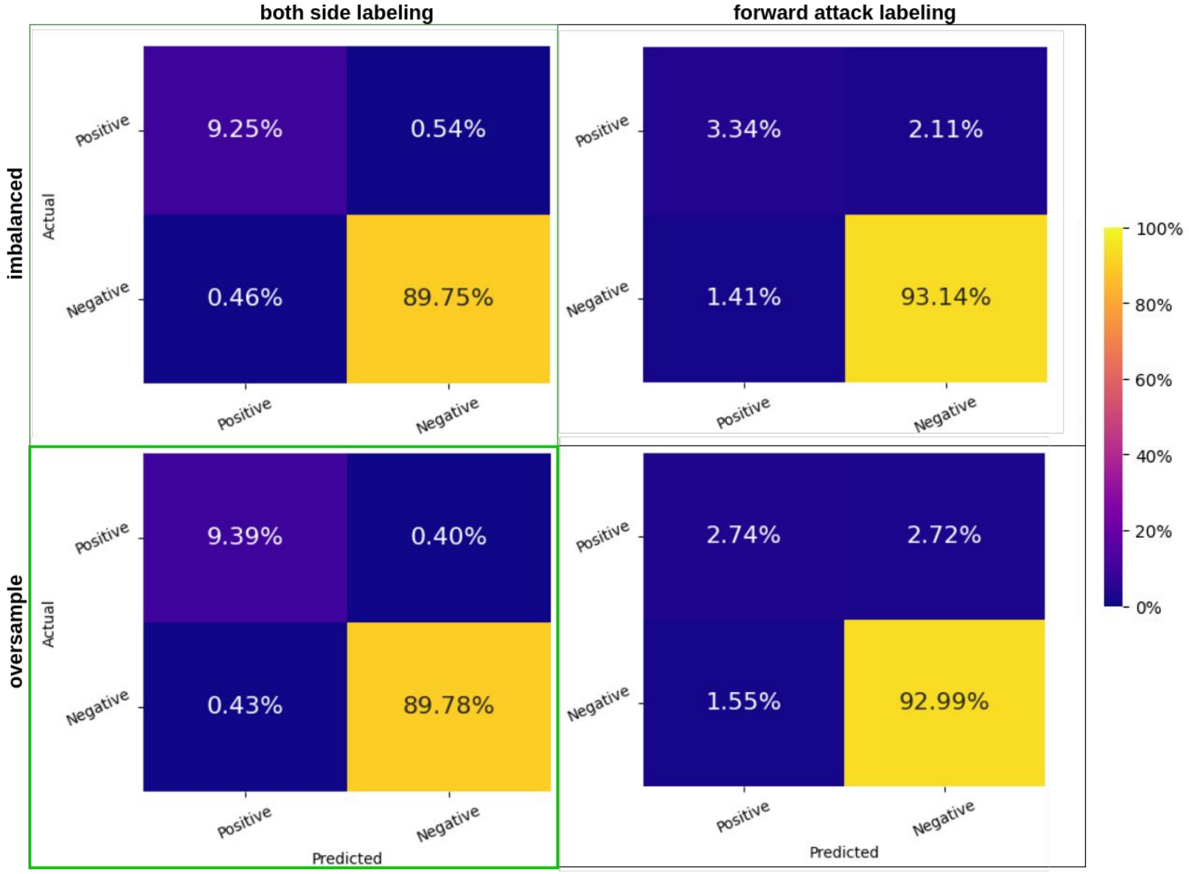




- Saliency map
  - averaged over the entire batch.

Results - EfficientNet

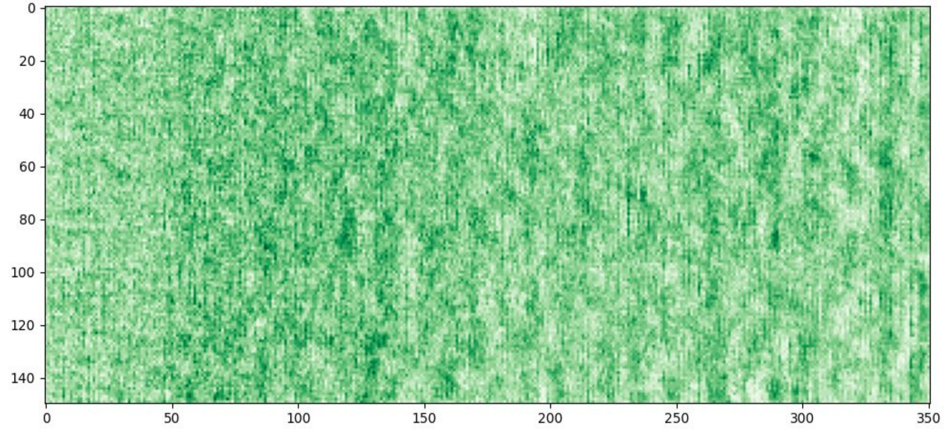
- Results on the test dataset
- Best results:
    - Binary Accuracy: 0.9917
    - Precision: 0.9561
    - Recall: 0.9588



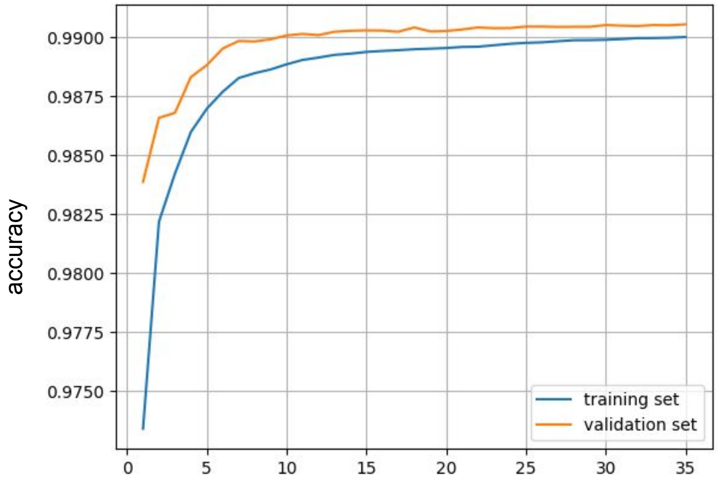
Results - Convolutional neural network

- Training history plot:
    - from the model with the highest accuracy,
    - epoch with best validation accuracy: 35,
    - model should be trained on more epochs.

saliency map



training history



- Saliency map
    - averaged over the entire batch.

Summary - the results comparison

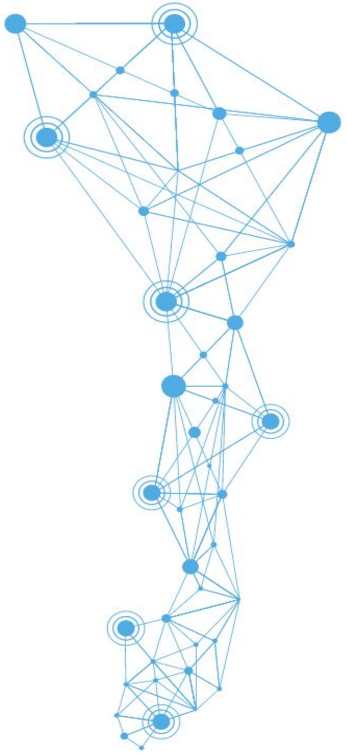
Related works on intrusion detection using CIC-IDS-2017 dataset.

Method	Accuracy [%]	Recall [%]	Precision [%]	Input Type	Classification (of)	Dataset
RF [4]	99.99	99.99	99.99	Flow features	Flow	CIC IDS 2017
DCNN [7]	99.96	99.96	99.96	Flow features	Flow	CIC IDS 2017
ET [9]	99.95	99.95	99.95	Flow features	Flow	CIC IDS 2017
RF [9]	99.94	99.94	99.94	Flow features	Flow	CIC IDS 2017
DT [9]	99.91	99.91	99.91	Flow features	Flow	CIC IDS 2017
CNN [8]	99.61	95.00	97.05	Flow features	Flow	CIC IDS 2017
XGB [9]	99.65	99.65	99.65	Flow features	Flow	CIC IDS 2017
CNN-LSTM [5]	99.48	99.69	99.25	Flow features	Flow	CIC IDS 2017
EP-FCNN [1]	99.50	-	-	Flow features	Flow	CIC IDS 2017
CNN-LSTM [3]	99.78	-	-	Flow features	Flow	CIC IDS 2017
CNN [3]	99.23	-	-	Flow features	Flow	CIC IDS 2017
EP-CNN [1]	98.80	-	-	Flow features	Flow	CIC IDS 2017
DT [2]	98.80	97.30	-	Flow features	Flow	CIC IDS 2017
EP-LSTM [1]	98.60	-	-	Flow features	Flow	CIC IDS 2017
DBN [3]	98.59	-	-	Flow features	Flow	CIC IDS 2017
SVM [3]	98.20	-	-	Flow features	Flow	CIC IDS 2017
LSTM [8]	97.67	95.95	94.96	Flow features	Flow	CIC IDS 2017
DNN [8]	90.61	84.60	80.85	Flow features	Flow	CIC IDS 2017
DID (LSTM) [6]	-	**99.80**	**99.20**	**Packets frame**	**Packets frame**	**CIC IDS 2017**
FCNN	**99.93**	**99.41**	**99.37**	**Packets Frame**	**Packets**	**Corr. CIC IDS 2017**
CNN	**98.77**	**94.66**	**92.65**	**Packets Frame**	**Packets**	**Corr. CIC IDS 2017**
CNN+LSTM	**98.85**	**95.18**	**93.01**	**Packets Frame**	**Packets**	**Corr. CIC IDS 2017**
EffNet	**99.17**	**95.61**	**95.88**	**Packets Frame**	**Packets**	**Corr. CIC IDS 2017**

Summary and outlook

Summary:

- FCNN model:
  - allows to obtain best metrics values:
  - results are comparable or better than the most of flows based solution,
  - model strongly based on the headers of the packets,
  - model can have difficulties to work with other datasets.
- Window based models:
  - obtained worse metrics values than FCNN,
  - pretrained EfficientNet provides best results,
  - labeling only forward networking significantly impedes to find features in windows,
  - models take into account most of the window: both header and payload,
  - models potentially can work with other datasets.



Outlook:

- Tune models hyperparameters with KerasTuner.
- Add dynamic windows shape.
- Check how LSTM and CNN would work with pretrained image-data.
- Introduce a way to classificate type of attack.
- Create Random Forest model that combine FCNN with 2D-window based methods.
- Verify how models predict data on other datasets and with on-line data.
- Perform models fine-tuning on other datasets

Thanks!

EuroCC2 project enables us to demonstrate usage of presented models on yours data!
Interested?
Mail or talk to us and ask about Proof-of-Concept possibilities.

NATIONAL CENTRE FOR NUCLEAR RESEARCH ŚWIERK

ŚWIERK COMPUTING CENTRE

Aleksander Ogonowski, Michał Żebrowski, Arkadiusz Ćwiek
National Centre for Nuclear Research, Świerk Computing Center
*https://**ai.ncbj.gov.pl***

EuroHPC Joint Undertaking

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101101903. The JU receives support from the Digital Europe Programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Türkiye, Republic of North Macedonia, Iceland, Montenegro, Serbia

References

[1] Jonghoon Lee et al. “Cyber threat detection based on artificial neural networks using event profiles”. In: Ieee Access 7 (2019), pp. 165607–165626.

[2] Yong Zhang et al. “PCCN: parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows”. In: IEEE Access 7 (2019), pp. 119904–119916.

[3] K Praanna et al. “A CNN-LSTM model for intrusion detection system from high dimensional data”. In: J. Inf. Comput. Sci 10.3 (2020), pp. 1362–1370.

[4] Gints Engelen, Vera Rimmer, and Wouter Joosen. “Troubleshooting an intrusion detection dataset: the CICIDS2017 case study”. In: 2021 IEEE Security and Privacy Workshops (SPW). IEEE. 2021, pp. 7–12.

[5] Asmaa Halbouni et al. “CNN-LSTM: hybrid deep neural network for network intrusion detection system”. In: IEEE Access 10 (2022), pp. 99837–99849.

[6] Mahdi Soltani, Mahdi Jafari Siavoshani, and Amir Hossein Jahangir. “A content-based deep intrusion detection system”. In: International Journal of Information Security 21.3 (2022), pp. 547–562.

[7] Vanlalruata Hnamte and Jamal Hussain. “Dependable intrusion detection system using deep convolutional neural network: A novel framework and performance evaluation approach”. In: Telematics and Informatics Reports 11 (2023), p. 100077.

[8] Jinsi Jose and Deepa V Jose. “Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset”. In: International Journal of Electrical and Computer Engineering (IJECE) 13.1 (2023), pp. 1134–1141.

[9] Md Alamin Talukder et al. “Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction”. In: Journal of Big Data 11.1 (2024), p. 33.