Improving 5G Intrusion Detection with Synthetic Data Generation and Transformer-Based Networks

Sina Eskandari

What is NIDS(Network Intrusion Detection System)?

- NIDS is a cybersecurity tool that monitors and analyzes network traffic for suspicious activities or security threats.
- It identifies potential threats by examining patterns, signatures, and anomalies in the network data, helping to detect and respond to unauthorized access or attacks.

The need for NIDS

- CIA(Confidentiality, Integrity, Availability)

- **600 billion** dollars loss caused by attacks until 2017

How to make a NIDS?

- Patterns, signatures, and anomalies needs to be examined.

- We can use deep learning!

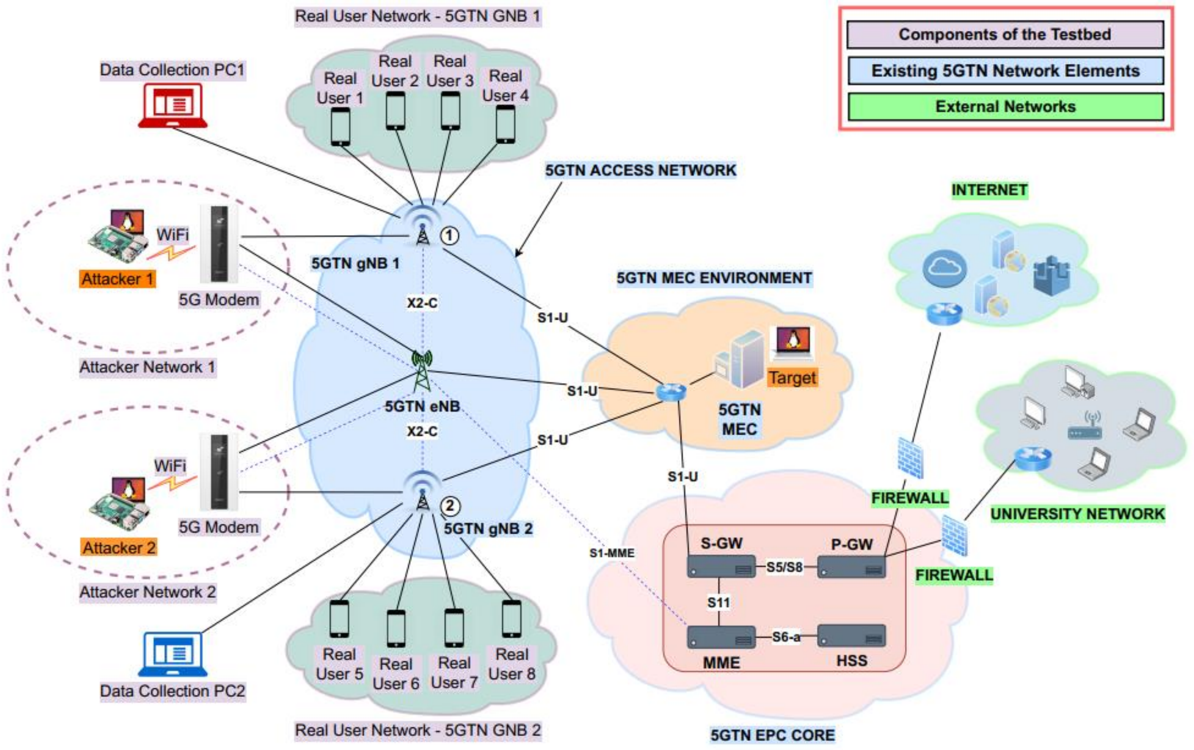
What do we need to make a NIDS using deep learning?

- Deep learning needs data and it needs a lot of it!
- Many available datasets such as: KDD-99, NSL-KDD, CICIDS2017

5G-NIDD Dataset

- A new dataset for 5G traffic, published in December 2022.
- It was collected in University of Oulu, Finland.
- Contains both malicious and benign data with different protocols such as:
    - HTTP
    - HTTPS
    - SSH
    - SFTP

5G-NIDD Testbed



Attacks in 5G-NIDD

- ICMP Flood: ICMP echo requests are sent to target rapidly.
- UDP Flood: Large volume of UDP packets is sent to the target.
- SYN Flood: Large number of SYN requests are sent to target without completing the three-way handshake.

Attacks in 5G-NIDD

- HTTP Flood:A large volume of seemingly legitimate HTTP requests is directed at a web server, overwhelming its capacity and causing service disruption.
- Slowrate DoS: HTTP requests are sent to a web server at slow rate, keeping many connections open simultaneously.
- Port Scans: A Technique used to discover open ports on a target system, providing information about potential vulnerabilities.

Preparing and preprocessing the data

In order to feed our data into deep learning models, we need to do some data cleaning and preprocessing.

Converting packets to Netflow

- There are two main approaches for NIDS regarding the data:
  - Packet-based: The information in packets' headers and payloads are processed.
  - Flow-based: The information in sequences of packets are summarized into one network flow.
- Packet-based approach is computationally expensive and hard to implement in large scale network.
- Network flows are much smaller and better suited for deep learning models.(4GB of packet captures were converted to 260MB of netflow.)
- Argus tool was used for packet to netflow conversion.

Handling missing values

- Having missing values in dataset can influence model’s performance.

- Two types of features:
  - Continuous
  - Categorical

- How we handle missing values depends on the proportion of records that does not have the feature.

Handling missing values

- If a lot of records miss a feature. → remove the feature.

- If few records miss a feature.
  - We can remove the records.
  - Alternatively, we can fill the feature with values.
    - For continuous features → replace missing values with mean, median or mode.
    - For categorical features → Add a new category called “missing”
  - We chose the second option.

Encoding categorical data

There are two approaches:

- Convert each class to a number.
    - This approach has a downside, which a class encoded to 3, will have 3 times the value of the class that encodes to 1.
- Use one-hot encoding.
    - We use an array of size d for d classes where one of the elements is 1 and others are 0.
    - For 3 classes we have: 1 → [0, 0 ,1], 2 → [0, 1, 0], 3 → [1, 0, 0]

Normalizing data

- Different features have different scales. This can cause a problem where one feature is ranged between 0 and 10000 and another feature is ranged between 0 and 1.
- Gradient descent algorithm converges faster when features have same scale.
- We used standard scaling, which changes the data to have mean of 0 and variance of 1.

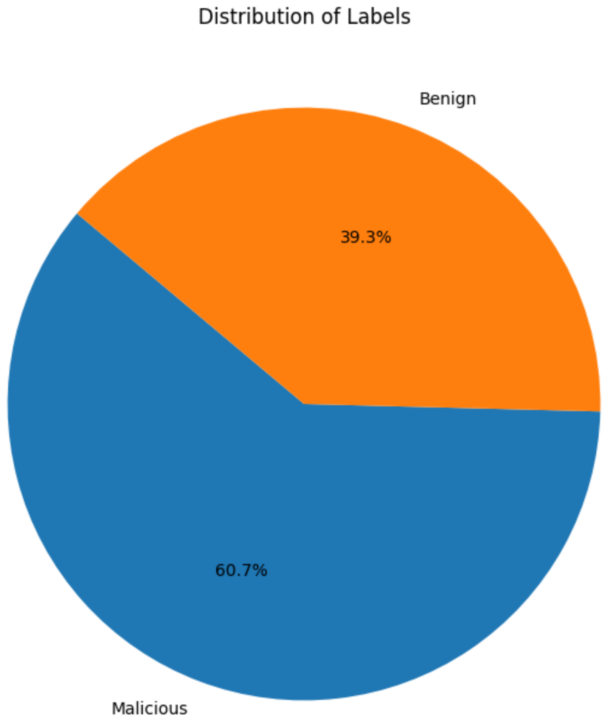
Splitting train and test dataset

- It is common to have different datasets for training and evaluating a deep learning model.
- In the past, when we had limited data, the split ratio was usually around 70-30 or 80-20.
- Now we have access to much more data, it is a waste to use a lot of data for test dataset. We have almost 1.2 million records and we split them to train and test with ratio of 97-3.

Unbalanced dataset problem

- Having an unbalanced dataset can lead to biased model.
- Many datasets in NIDS have this problem.
- When datasets are unbalanced, evaluation metrics like accuracy can be misleading.

Unbalanced dataset problem



Balancing the dataset

- Previous works: SMOTE(Synthetic Minority Over-sampling Technique)
  - Generates samples by interpolating a data sample with its nearest neighbors.
  - Problems: sample overlapping, noisy samples, difficulty in choosing ideal number of neighbors.

- Generative models can learn data distribution and generate data.

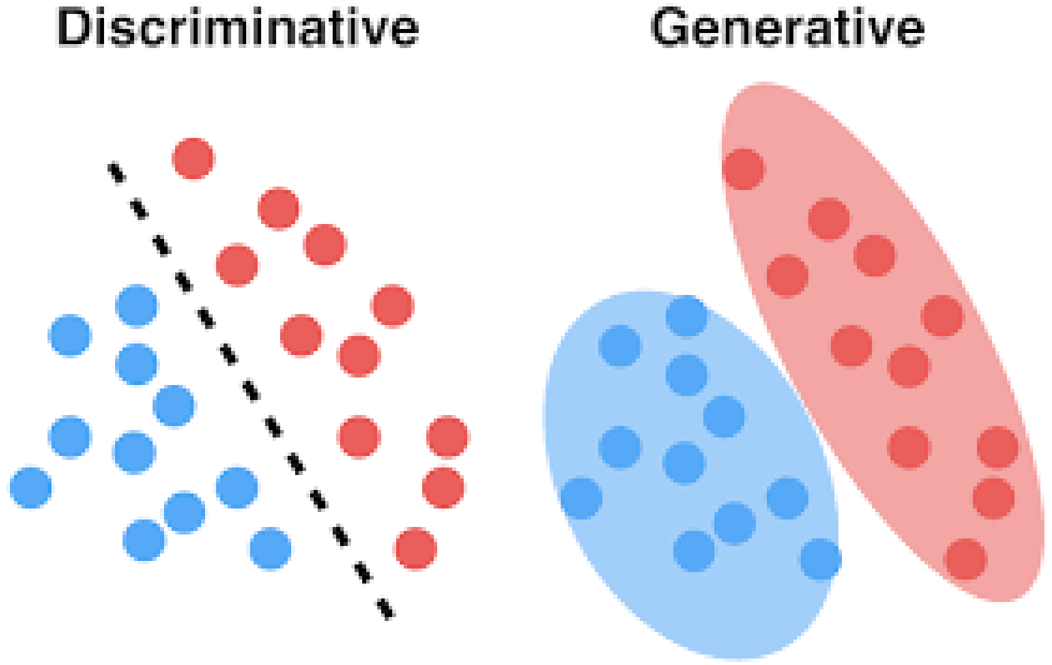
Types of models in machine learning

There are generally two types of models in machine learning:

- Discriminative models: learns $P(y|x)$ and models a decision boundary between different classes.

- Generative models: learns $P(x, y)$

Types of models in machine learning



Generative adversarial network(GAN)

- First introduced in 2014 for image generation, but now they are used in various domains.

- They can help with balancing datasets and data augmentation.

- They can improve the quality of data as well as adding more variety.

Generative adversarial network(GAN)

GANs are consist of two main component:

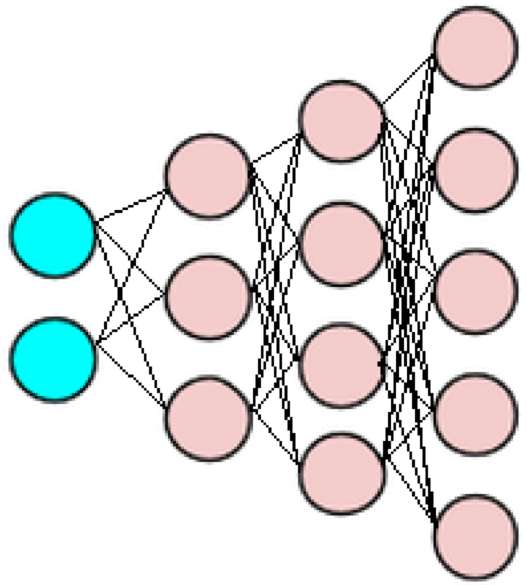
- Generator network
- Discriminator network

Generator network

- It is a neural network which gets a noise as input and converts it to a meaningful sample.
- Usually starts with a low dimensional noise and converts it to higher dimensional sample.
- Experiments show that the noise distribution is not that important. So we use something that is easy to sample from, such as uniform or normal.

Generator network

noise



output

Discriminator network

- It serves as the adversary of the generator network.
- Its goal is to differentiate between synthetic and real data.
- Through the training, it improves its ability to detect fake data and pushes generator to creating more realistic data.

GAN’s training process

- Generator aims to create samples that are indistinguishable from real data.
- Discriminator aims to become better at recognizing fake data.
- There is an adversarial training in GAN and both networks become better through having competition.

Discriminator objective function

$$\frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(x^{(i)}\right)+\log\left(1-D(G(z^{(i)}))\right)\right]$$

Generator's objective function

$$\frac{1}{m}\sum_{i=1}^{m}\left[\log\left(1 - D(G(z^{(i)}))\right)\right]$$

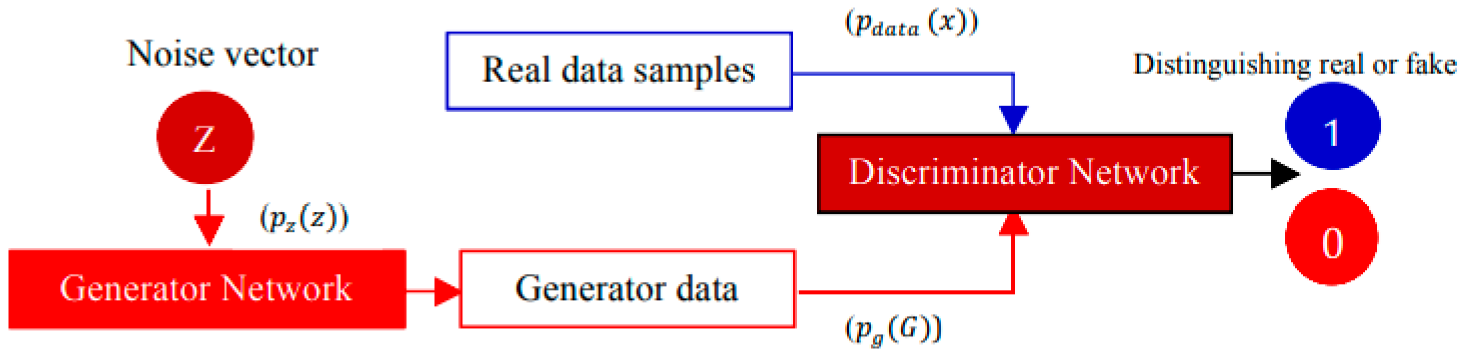
Training loop in GAN

1. Generator and discriminator are initialized with random parameters.
2. We choose m noise samples and m real data samples.
3. Output of generator for noises are calculated.(fake samples)
4. Output of discriminator for fake samples and real data are calculated.
5. Discriminator parameters gets updated by using gradient ascent on its objective function.

Training loop in GAN

6. We choose m noise samples.
7. Output of generator for noises are calculated.(fake samples).
8. Output of discriminator for fake samples are calculated.
9. Using gradient descent algorithm on generator’s objective function.
10. Steps 2 to 9 are repeated for each epoch of training.

GAN architecture



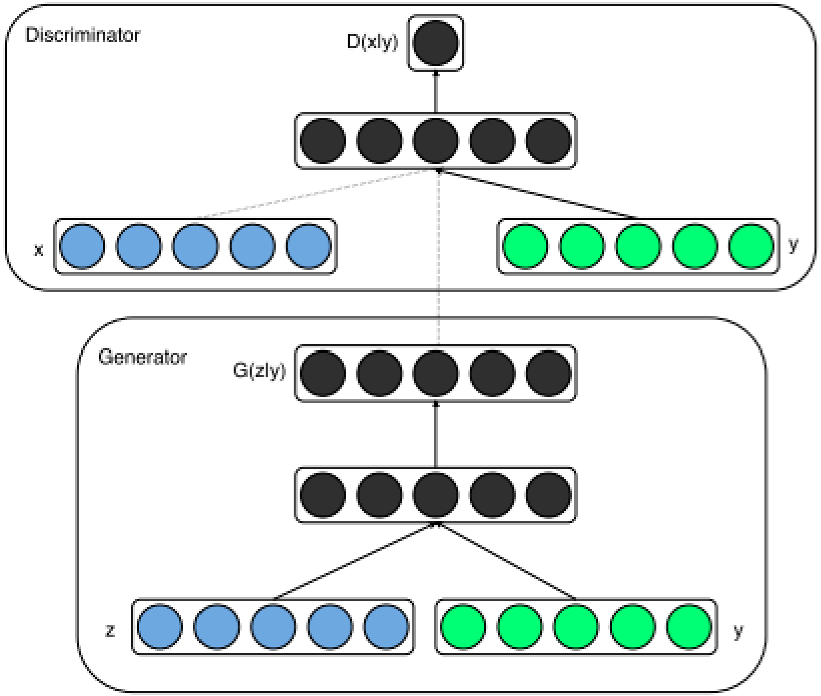
Simple GAN’s limitation

- After training, generator gets noise samples and generates data using them.
- There is no control on the synthetic data that gets generated.
- In some scenarios we want to generate data from a specific class or have a condition met by the generated sample.
- Simple architecture of GAN is incapable of doing this, since it only gets a noise as input.

Conditional GAN(CGAN)

- We can include another input for generator to provide the condition we want.
- This condition can be the label of data, regulations on the data or any condition we want.
- The objective function become conditional and in training process we make sure to choose data samples that met the condition.

Conditional GAN architecture



Hyperparameters in our implementation of CGAN

- 2 hidden layers with each having 256 neurons in generator and discriminator
- We used dropout with probability of 0.5 for preventing overfitting.
- Due to hardware limitations, feeding the entire dataset is not possible and we feed the data in batches with size of 500.
- We trained the model for 20 epochs.
- Adam optimization algorithm was used with learning rate of $2*10^{-4}$

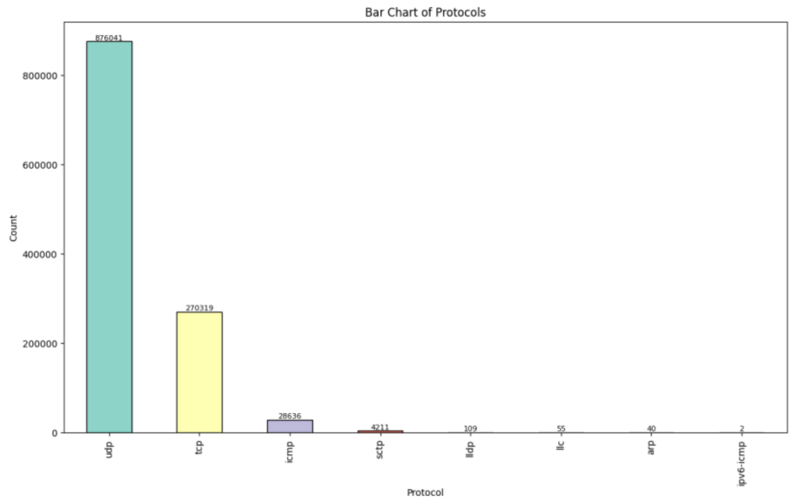
How to evaluate the generated data?

- GANs were introduced in the field of computer vision and their performance was evaluated using Inception score.
- It uses a pre-trained model to evaluate and compare the accuracy of real and synthetic data.
- This method is not possible in our case, because in tabular data, features are heterogeneous. Hence it is impossible to have a pre-trained model to use in different datasets.

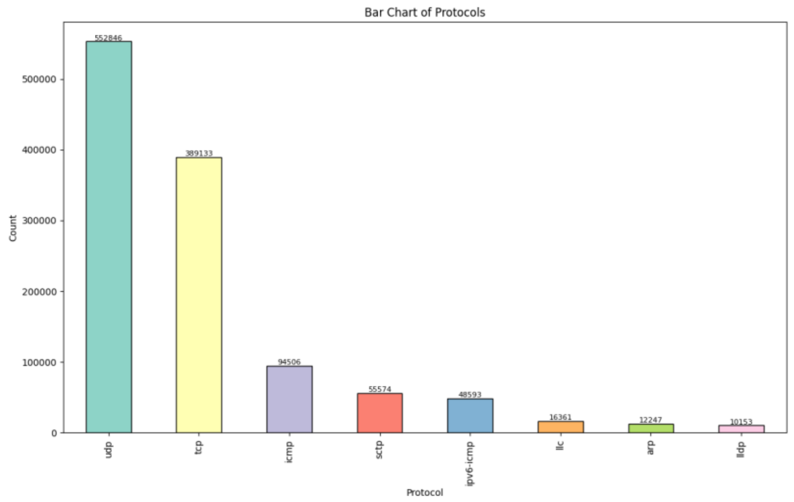
How to evaluate the generated data?

- Using the idea of Inception score, we created a simple neural network with 1 hidden layer and trained it on the real data. We used the same model to evaluate the synthetic data.
- With this model, we achieve accuracy of 96 on train data and 92 on synthetic data.
- We can say that our data is good enough to improve our dataset and add more variety to the data.

Comparing real data and synthetic data



Real data



Fake data

Types of learning in machine learning

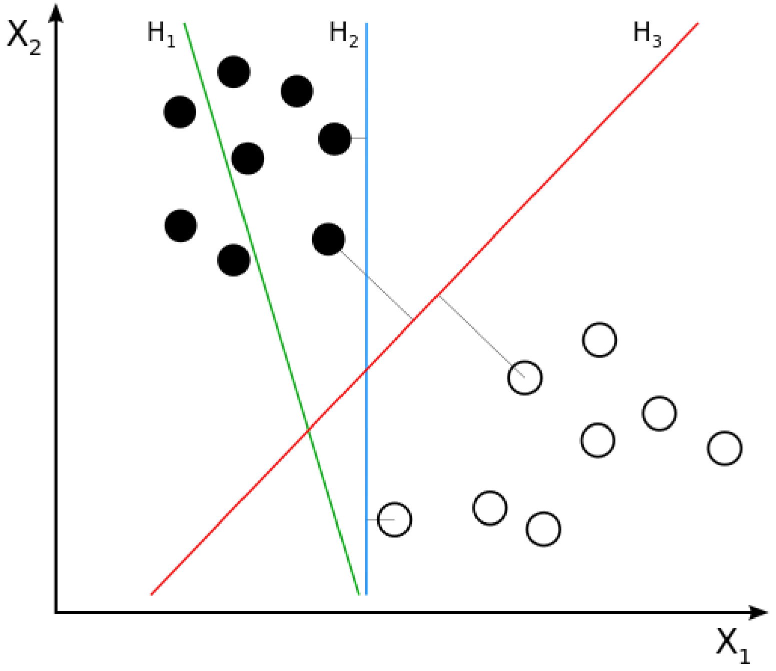
- Supervised learning
  - Dataset has labels for data and model predicts the label.
- Unsupervised learning
  - Dataset has no labels and model learns structural properties of data. It is commonly used for finding similarities between data points, clustering and finding anomalies in data.

Classification

- Classification is often a supervised task.
- We need to have a robust classifier for having a good NIDS.
- In previous works, many machine learning and deep learning algorithms were used.
- We review some of these methods.
- Then we introduce evaluation metrics for classification tasks.
- Eventually, we propose a state-of-the-art model for classification.

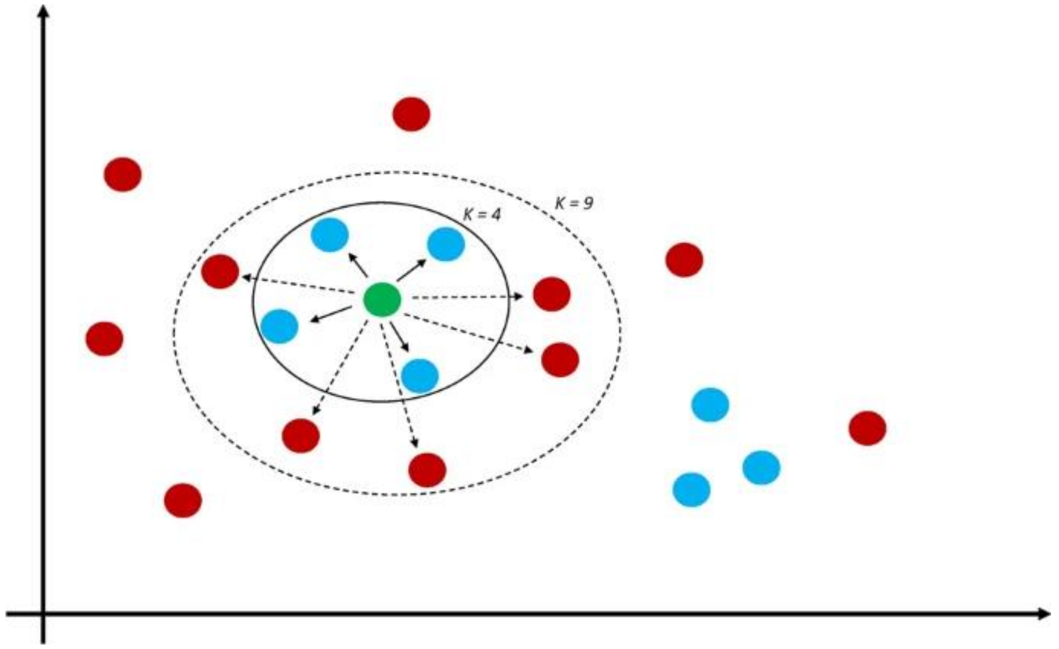
Previous works

- Support vector machines:



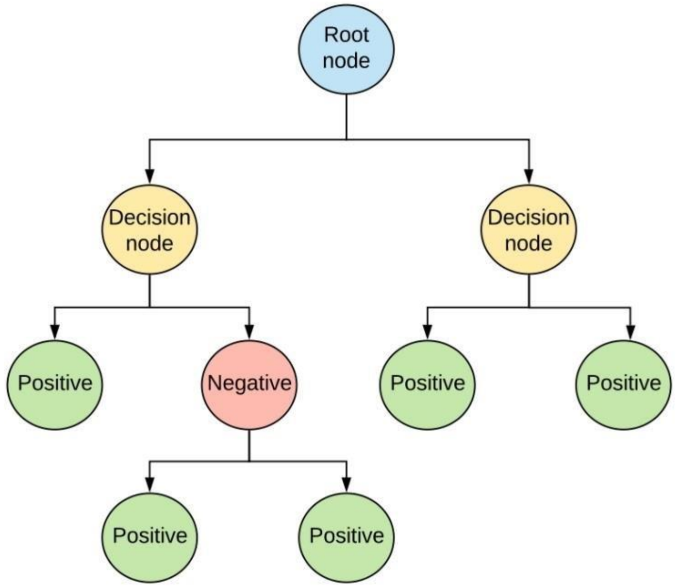
Previous works

- K nearest neighbors:



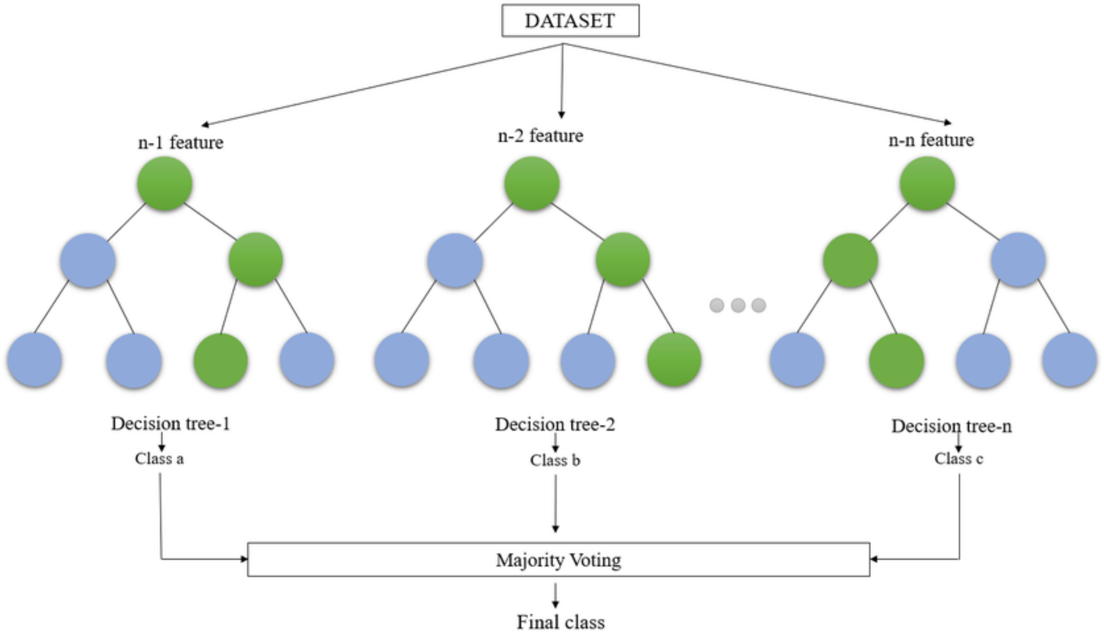
Previous works

- Decision Trees:



Previous works

- Random forest:



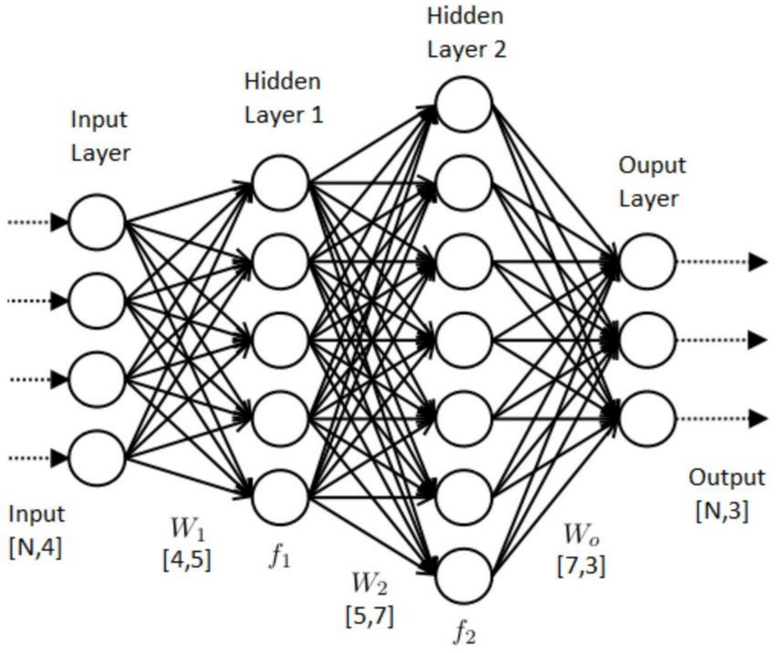
Previous works

- Naive Bayes

$$P(y|x_1, x_2, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, x_2, \ldots, x_n)}$$

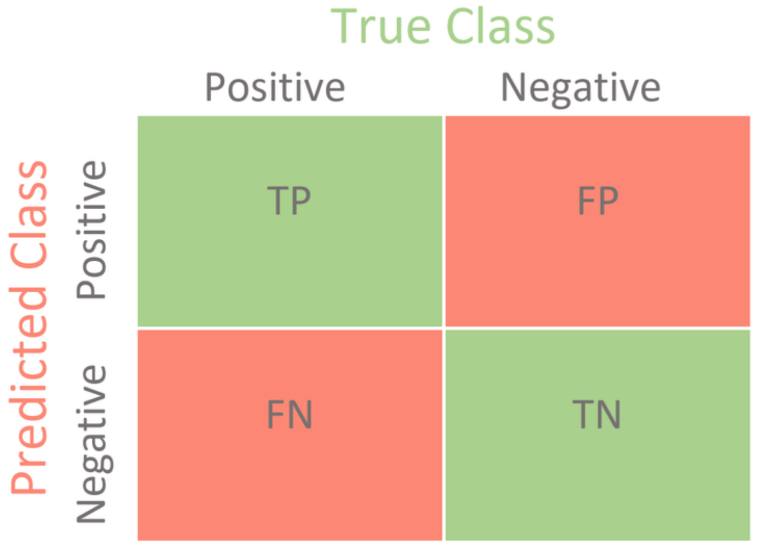
Previous works

- Neural networks:



Evaluation metrics

- Confusion matrix



		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Evaluation metrics

- Accuracy: Ratio of all correctly predicted to all predictions

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Evaluation metrics

- Precision: Ratio of all positive samples that were predicted as positive to all samples that were predicted as positive.

$$Precision = \frac{TP}{TP + FP}$$

Evaluation metrics

- Recall: Ratio of all positive samples that were predicted as positive to all positive samples. It is also called true positive rate.

$$Recall = \frac{TP}{TP + FN}$$

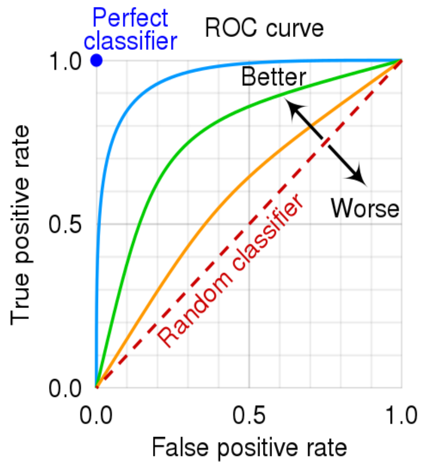
Evaluation metrics

- F1-score: Harmonic mean of precision and recall.

$$F1-Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

Evaluation metrics

- Receiver Operating Characteristic curve: a graphical representation that illustrates the trade-off between true positive rate and false positive rate across different thresholds. AUC is the area under this curve and the closer to 1 the better.



Transformer model

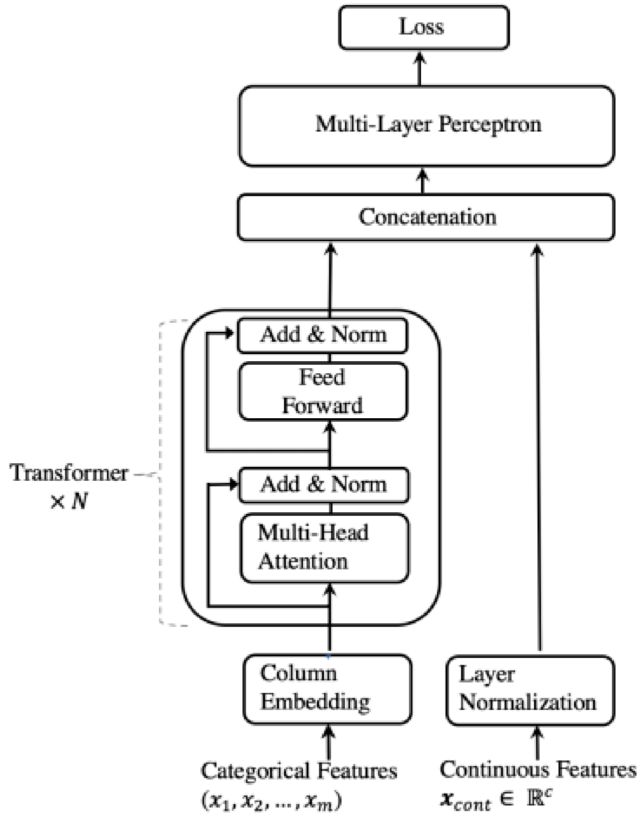
- First introduced in 2017 in NLP and machine translation.
- Main idea was self-attention.
- Was able to extract patterns and dependencies in sequences of data.
- After its huge success in NLP, it was extended to other domains such as image processing and tabular data analysis.

Self-attention mechanism

- It allows the model to have more or less attention on different parts of the input.
- In our case, each row in table contains features including continuous and categorical.
- Using attention score, importance of each feature with respect to other features is measured in each sample.

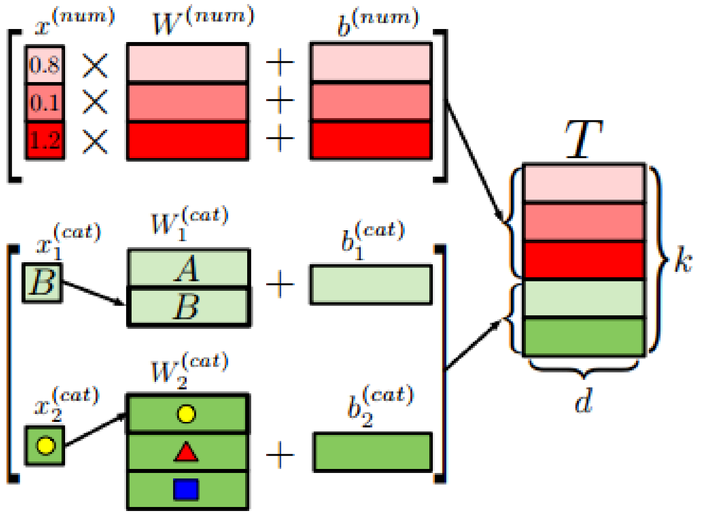
TabTransfomer

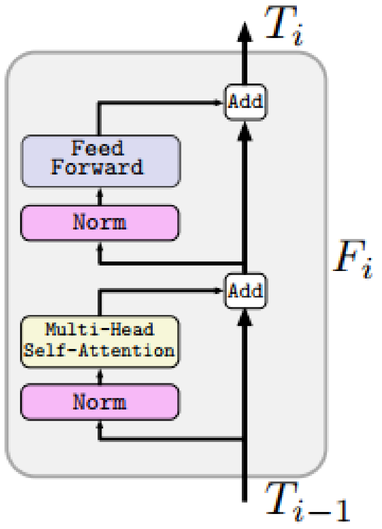
- First usage of transformers in tabular data.



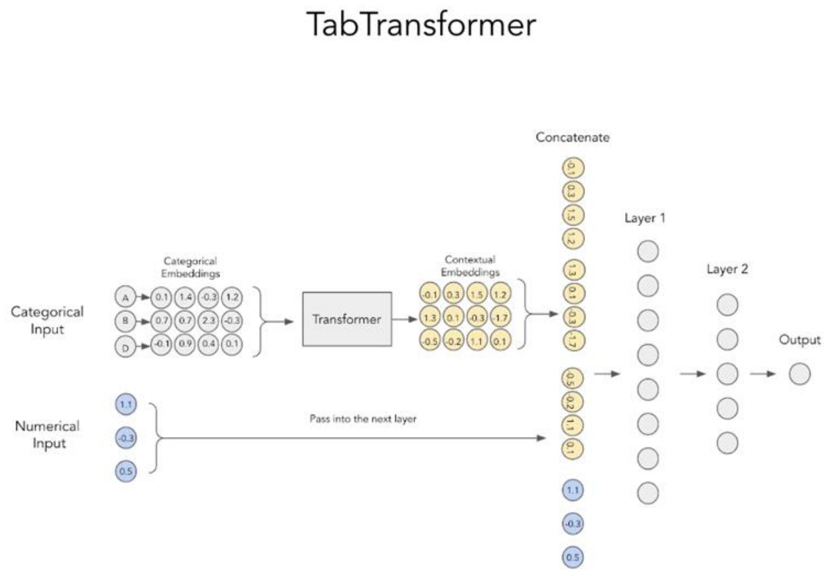
FT-Transformer

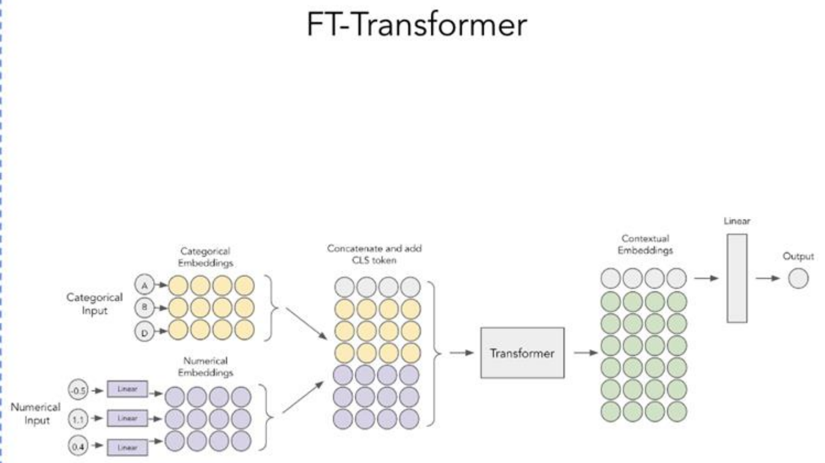
- Extension of TabTransformer





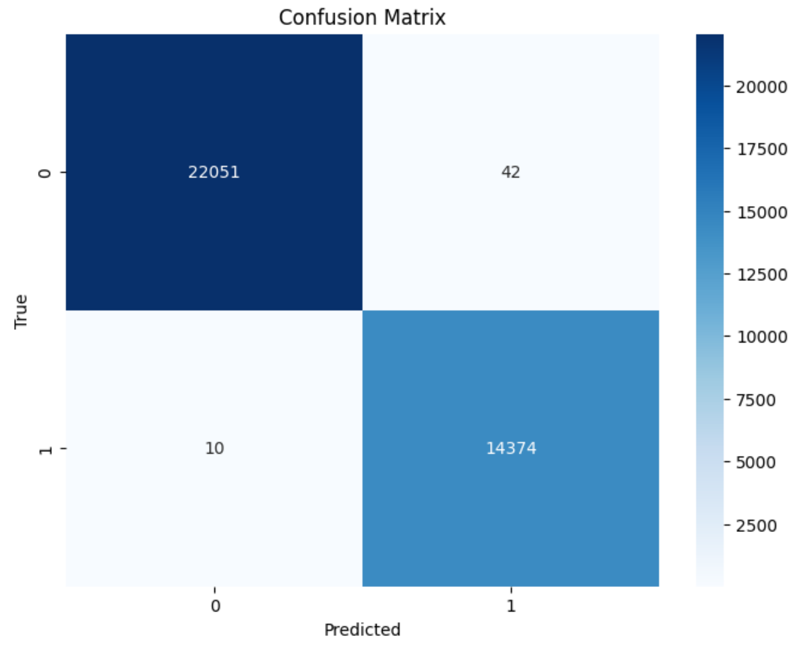
TabTransformer vs FT-Transformer





Results

- Confusion matrix:

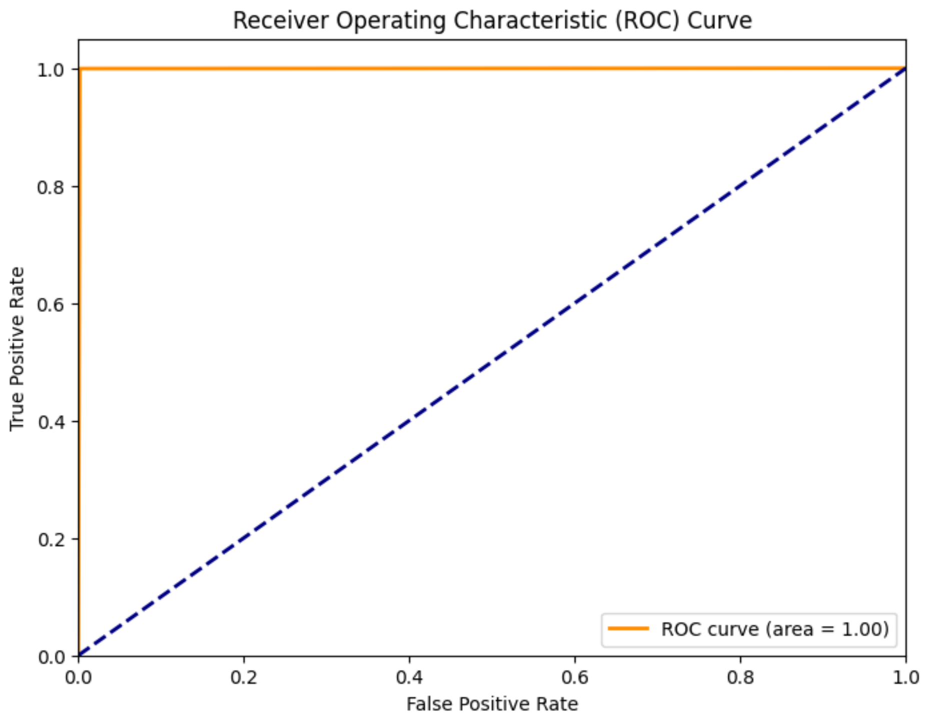


Evaluation

Metric	Value
Accuracy	0.999
Precision	0.997
Recall	0.999
F1-Score	0.998

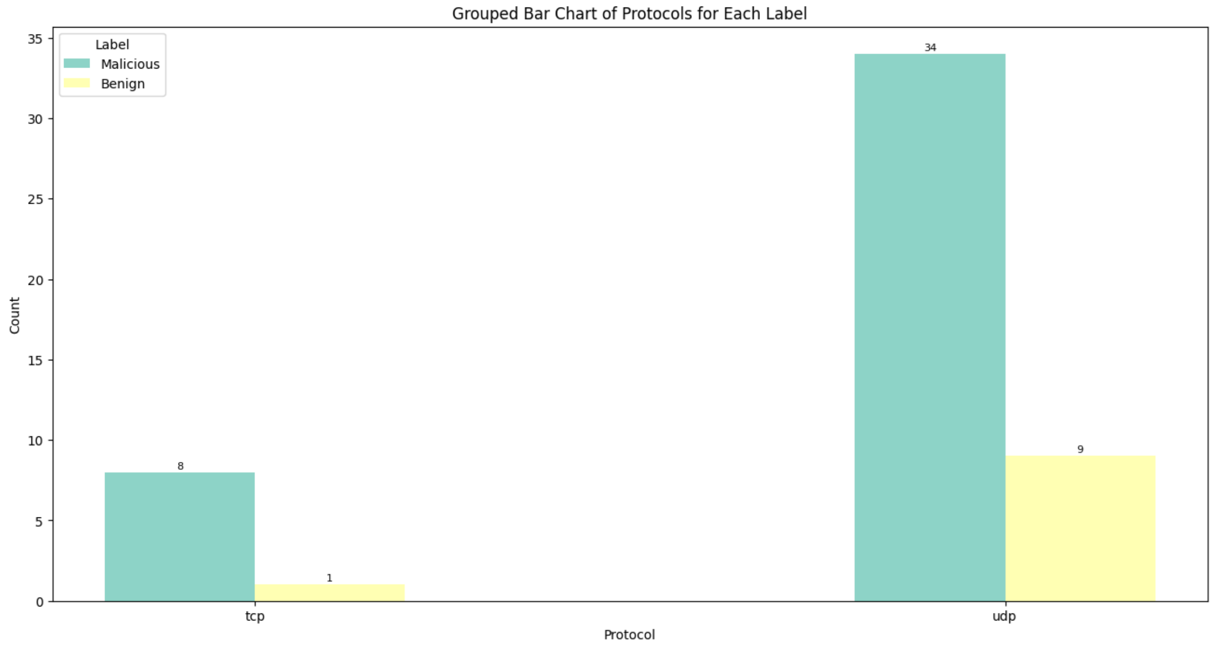
Evaluation

- ROC curve and AUC



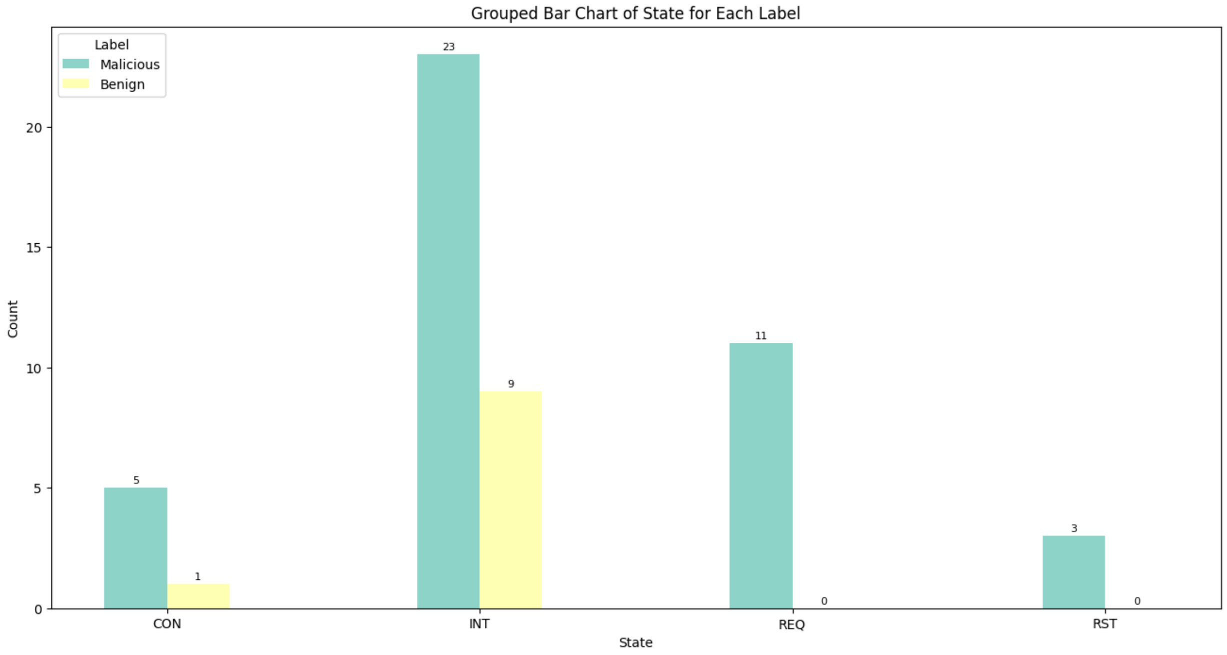
Stats about wrong predictions

- Protocol of samples that are wrongly predicted



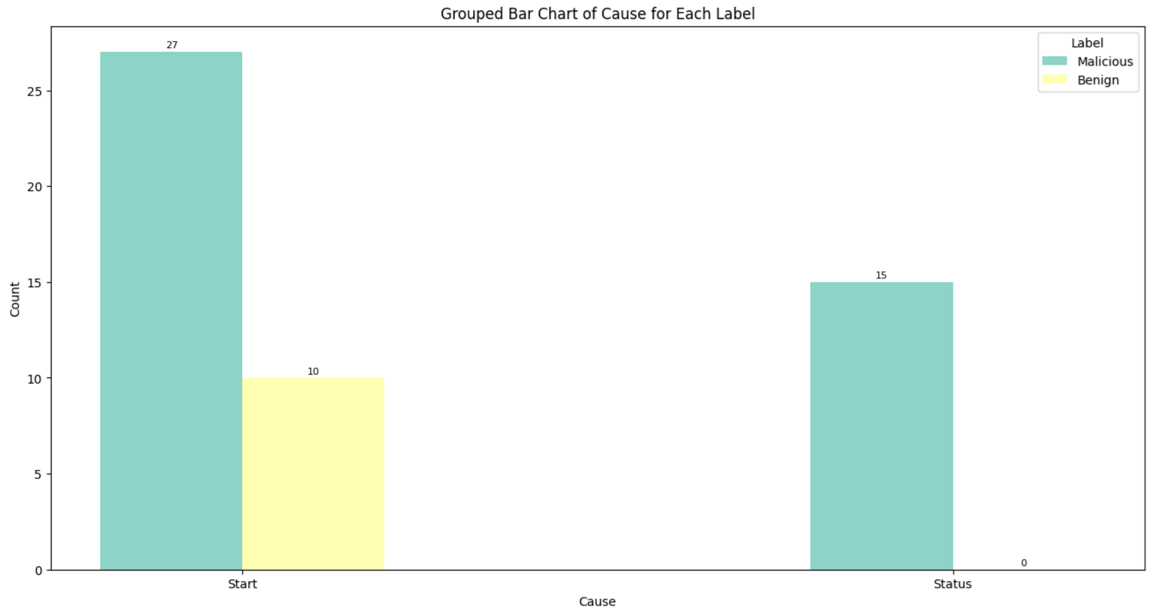
Stats about wrong predictions

- State of samples that are wrongly predicted



Stats about wrong predictions

- Cause of samples that are wrongly predicted


Grouped Bar Chart of Cause for Each Label
Label
Malicious
Benign
Count
Cause
Start
Status