

# ANOMALII

## DETEKCJA

MURAVYTSKYI  
SMENDOWSKI  
SKOŚ

# Dane – rozpoznanie zbioru

Dataset-Unicauca-Version2-87Atts.csv (1.65 GB)									
Detail	Compact	Column	87 of 87 columns						
▲ Flow.ID	▲ Source.IP	# Source.Port	▲ Destination.IP	# Destination.Port	# Protocol	▲ Timestamp	# Flow.Duration	# Total.Fwd.Packets	# Total.Backward
a flow identifier following the next format: SourceIP-DestinationIP-SourcePort-DestinationPort-TransportProtocol	The source IP address of the flow.	The source port number	The destination IP address.	The destination port number.	The transport layer protocol number identification (i.e., TCP = 6, UDP = 17).	The instant the packet was captured stored in the next date format: Dd/mm/yyyy HH:MM:SS	The total duration of the flow	The total number of packets in the forward direction.	The total number of packets in the backward direction.
1522917 unique values	10.200.7.218 8% 10.200.7.217 7% Other (3013817) 84%		10.200.7.8 9% 10.200.7.7 9% Other (2939141) 82%			41915 unique values			
172.19.1.46-10.200.7.7-52422-3128-6	172.19.1.46	52422	10.200.7.7	3128	6	26/04/201711:11:17	45523	22	55
172.19.1.46-10.200.7.7-52422-3128-6	10.200.7.7	3128	172.19.1.46	52422	6	26/04/201711:11:17	1	2	0
10.200.7.217-50.31.185.39-38848-80-6	50.31.185.39	80	10.200.7.217	38848	6	26/04/201711:11:17	1	3	0
10.200.7.217-50.31.185.39-38848-80-6	50.31.185.39	80	10.200.7.217	38848	6	26/04/201711:11:17	217	1	3
192.168.72.43-10.200.7.7-55961-3128-6	192.168.72.43	55961	10.200.7.7	3128	6	26/04/201711:11:17	78068	5	0
172.19.1.56-10.200.7.6-50004-3128-6	10.200.7.6	3128	172.19.1.56	50004	6	26/04/201711:11:17	105069	136	0
192.168.72.43-10.200.7.7-55963-3128-6	192.168.72.43	55963	10.200.7.7	3128	6	26/04/201711:11:17	104443	5	0

# Dane – analiza

- Oryginalny Data Set składa się z danych nominalnych oraz numerycznych - **dane nieetykietowane**.
- **Uczenie Nienadzorowane** - budując klasyfikator nadzorowany bylibyśmy w stanie rozpoznawać tylko dobrze znane przypadki. **Anomalie to ruch nieprzewidziany, niespotkany w sieci**.
- **Parametry dyskretne** wymagały osobnej normalizacji.  
Parametry nie tworzące rozkładu: porty, protokoły, identyfikatory przepływów, nazwy usług.
- **Parametry ciągłe numeryczne** poddane normalizacji max-min (max - 1; min - 0).  
Parametry tworzące rozkład: rozmiar pakietu, liczba bajtów należąca do przepływu.

# Dane – analiza

## - Początkowy rozmiar to 1.65 GB.

Wartości były zapisane na intach, po normalizacji do przedziału [0,1] przejście na float64 było konieczne.

## - Redukcja wymiarowości danych.

Usunięto kolumny flow ID (cecha pochodna na bazie 5 tuple), timestamp - nie uwzględniamy zależności czasowych, 8 kolumn mających 70% wartości które są zerami, kolumny mające jedną tą samą wartość (dane nic nie wnoszące), usunięto mocno skorelowane parametry- manualna inspekcja danych (odeszły parametry statystyczne)

## - One Hot Encoding na protokołach warstwy 3 . Z 1 kolumny zrobiono 3: TCP, UDP, Unknown.

Jeśli ruch UDP to w kolumnie "UDP" zostaje wpisana 1, w "TCP" oraz "Unknown" są zera.

## - Frequency Encoding na usługach

(liczba występowania danej cechy / całkowita liczba wierszy w tej cenie – tak otrzymana wartość zastępuje cechę)

## - Porty oraz adresy IP.

Metodyka na przykładzie IP: wzięto 100 najczęstszych wartości IP.

Jeśli dany adres występuje w pierwszym 100 to przypisujemy mu wartość 0 (niepodejrzany).

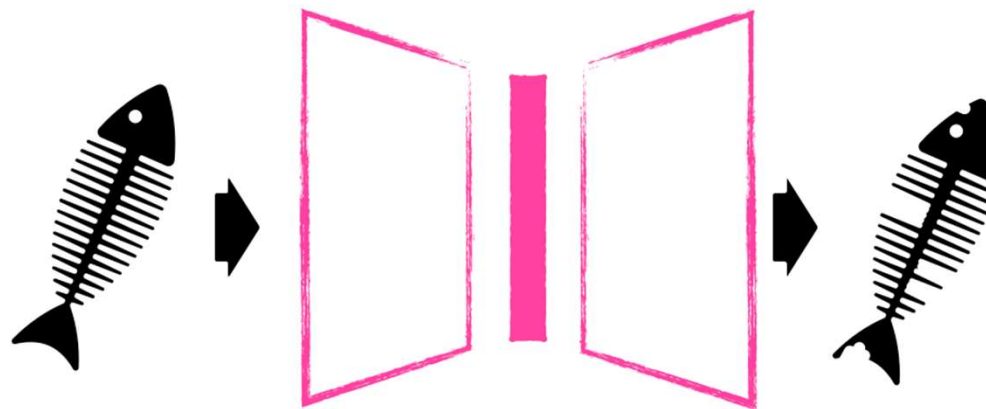
Jeśli to pierwsze wystąpienie adresu IP to przypisujemy mu wartość 1 (podejrzany).

Jeśli nie występuje w pierwszych 100 ale pojawia się nie pierwszy raz to przypisujemy mu wartość 0.5

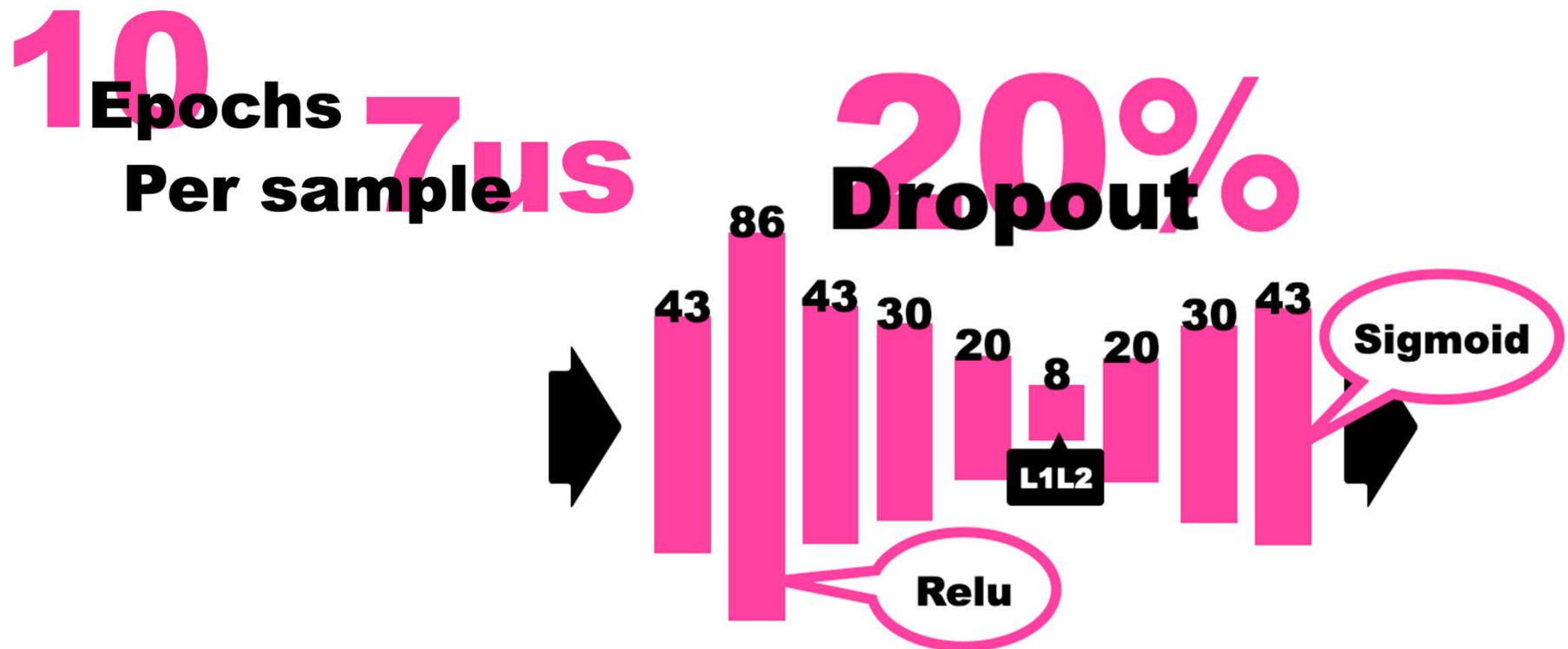
## - Finalny rozmiar danych: 2.15 GB 43 kolumn

# Autoencodery

- Zadaniem autoencodera jest zakodowanie i zrekonstruowanie danych wejściowych. Liczymy błąd porównując wejście z wyjściem.
- Nauczyliśmy model na danych właściwych dla sieci tak, aby umiał zrekonstruować nieanomalne dane poniżej ustalonego progu błędu.
- W przypadku otrzymania przekroczy ww. próg.



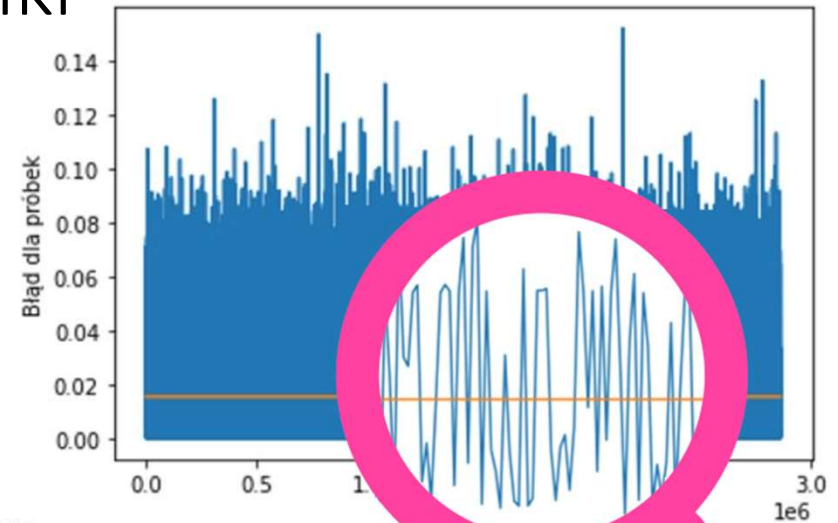
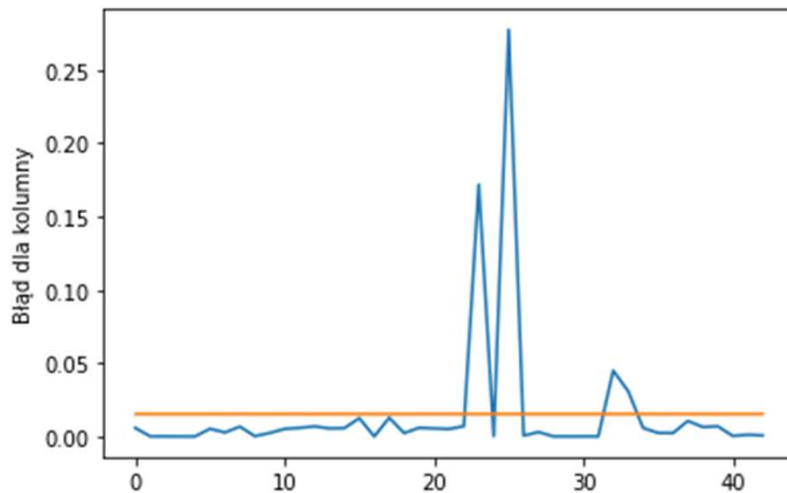
# **Autoencodery** – kilka statystyk



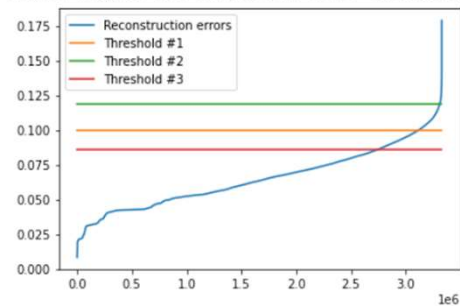
# Autoencodery – wyniki

2 kolumny z największym błędem:

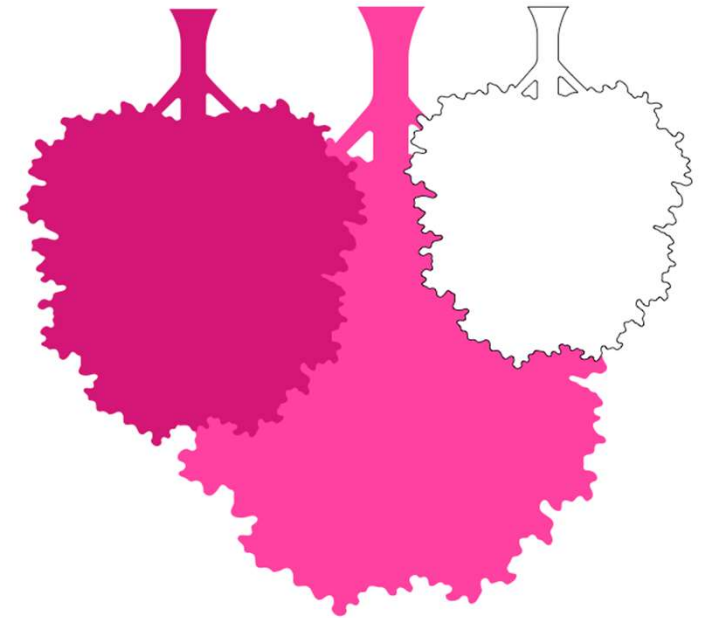
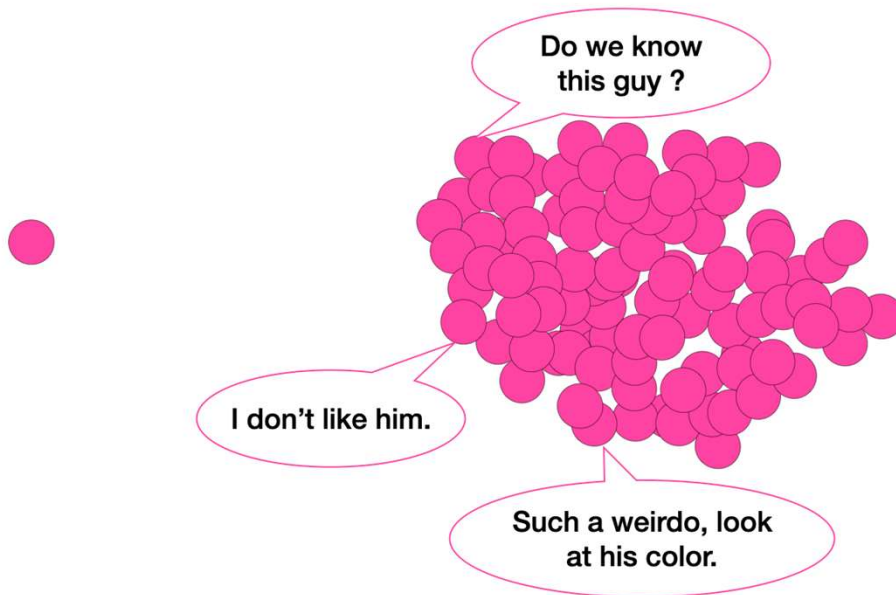
- SYN.Flag.Count
- URG.Flag.Count



Threshold 1 - mean.  
Threshold 2 - quantile 99.5% .  
Threshold 3 - mean + std.  
207508 samples are suspicious feat. threshold #1.  
16637 samples are suspicious feat. threshold #2.  
575933 samples are suspicious feat. threshold #3.

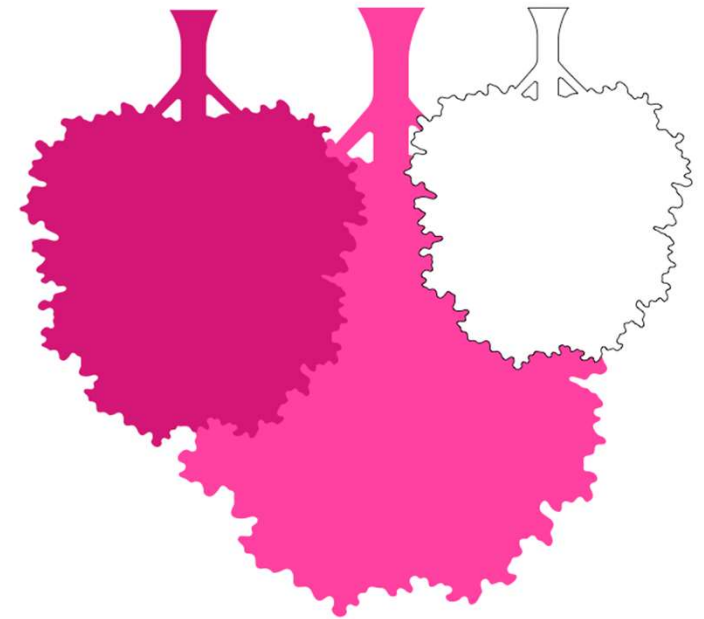
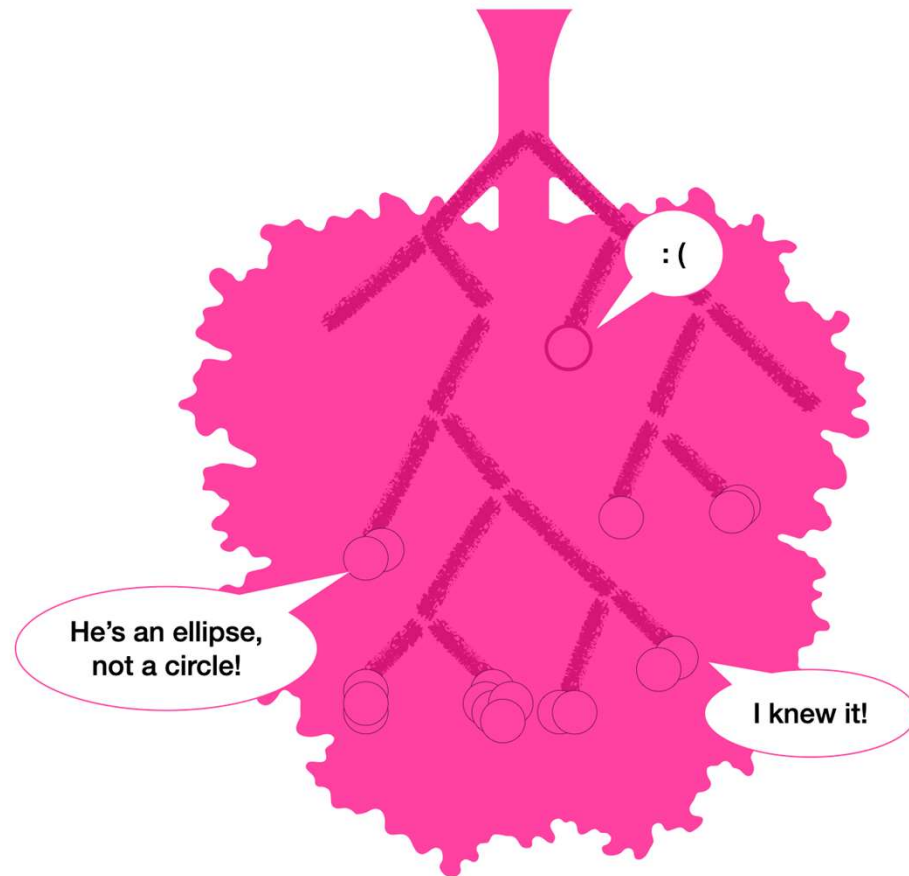


# Isolation Forest

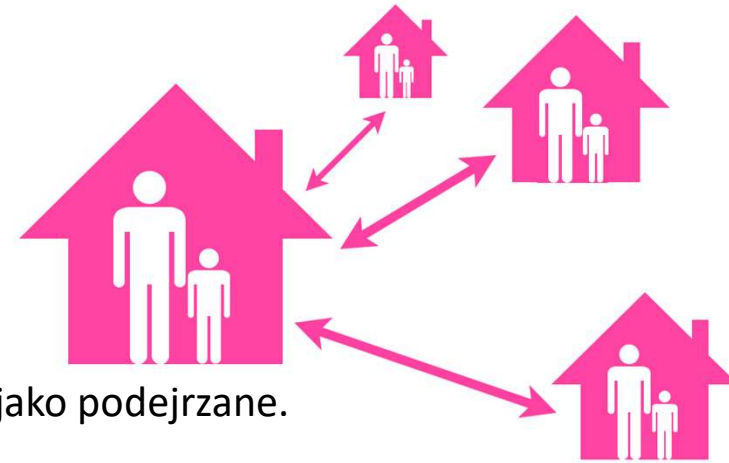




# Isolation Forest



# K Nearest Neighbours



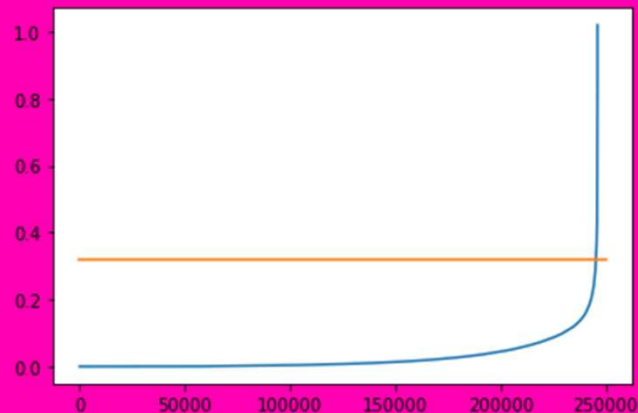
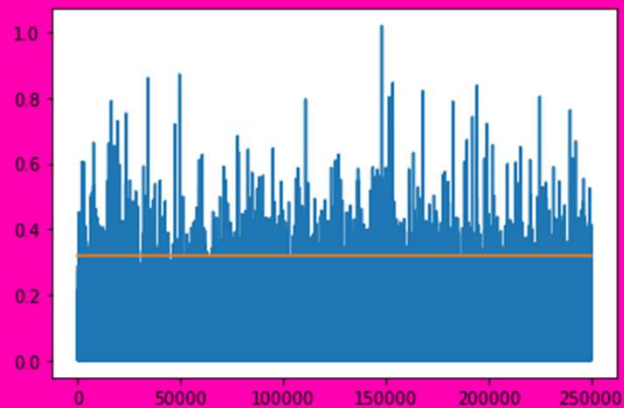
- Zastosowano algorytm w wariacie **Unsupervised KNN**
- Administrator często manualnie sprawdza wpisy zakwalifikowane jako podejrzone.
- Warto mieć "**rozsądnie**" **więcej danych** do weryfikacji niż mniej.  
Koszt False Positive jest o wiele większy niż True Negative.

100 000 wierszy dla K=5 to 27 podejranych wpisów, dla K=10 to 2 podejrane wpisy.

- **Podejście 1:** [Punkt wyjścia] Detekcja punktów odstających ze zbioru.
- **Podejście 2:** Symulacja działania sieci w czasie rzeczywistym - klasyfikacja i detekcja na interfejsie.  
Losowo wybrane 250k wpisów tworzy zbiór danych referencyjnych na bazie którego wyznacza się threshold dla danej wartości K. Pozostałe dane to dane testowe po których iterując dokonuje się klasyfikacji.
- Wyznaczenie thresholdu to jednorazowy effort -> (**Avg processing time of single sample 2.5 ms**)
- **Threshold (centyl 99.5)** - wartość powyżej której stwierdzamy że dana próbka jest anomalna.

# K Nearest Neighbours


Entire Data Set has 3577296 samples.  
Reference Data Set has 250000 samples.  
Testing Data Set has 3327296 samples.



Oś Y: Średnia arytmetyczna odległości do K sąsiadów  
Oś X: Indeks próbki ze zbioru referencyjnego

**Centyl 99.5 -> Empiryczny dobór**

# Autoencoder vs KNN vs Isolation Forest


 Detekcja 1 próbki: 1,5ms (KNN) vs 7us (Autoencoder)  
vs 12us (Iso Forest)

 Autoencodery potrzebują więcej danych do nauki

 Autoencoder można doszkolić

 Autoencodery potrafią wykrywać złożone zależności

# Autoencoder vs KNN vs Isolation Forest

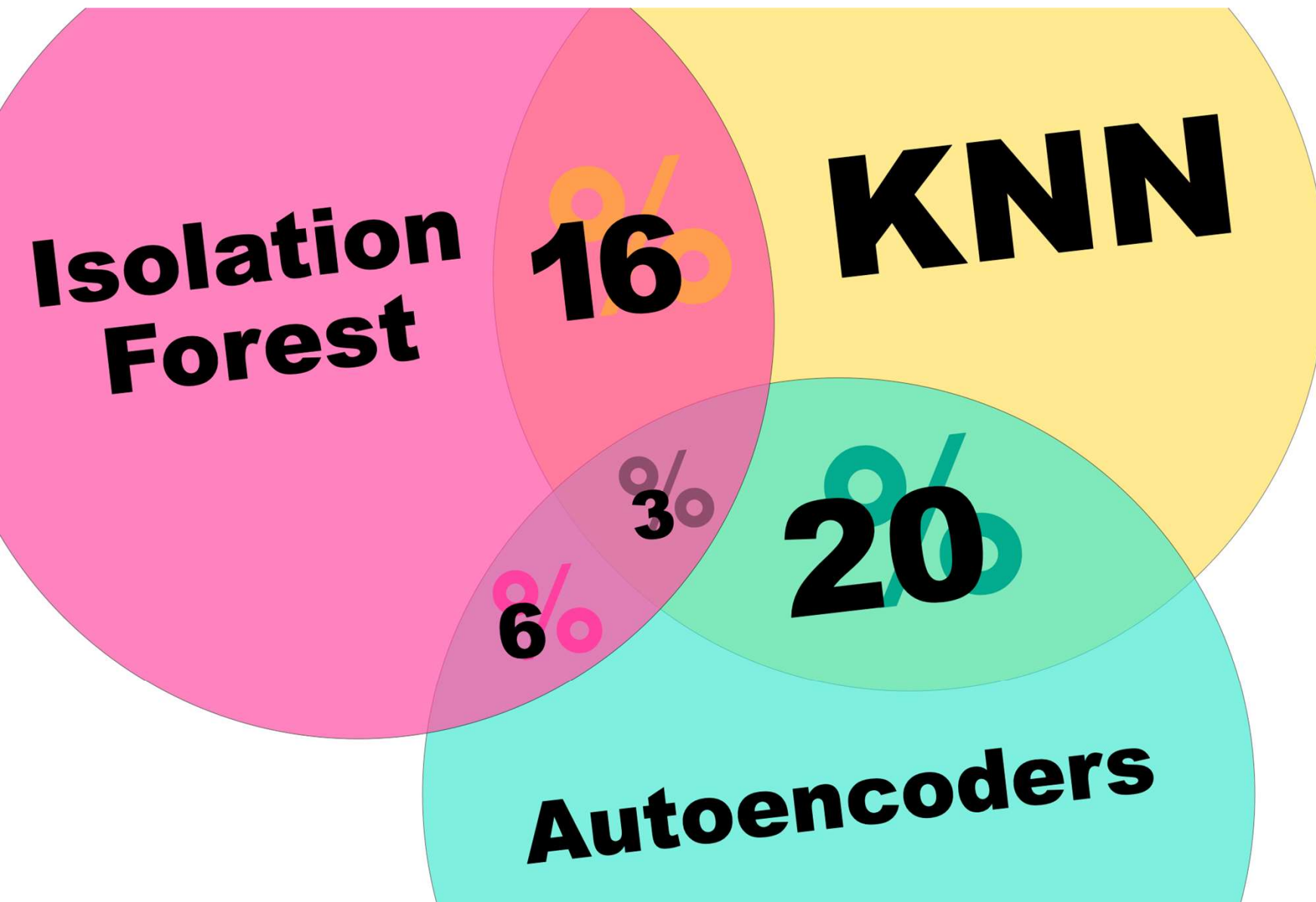
 Detekcja 1 próbki: 1,5ms (KNN) vs 7us (Autoencoder)  
vs 12us (Iso Forest)

 Autoencodery potrzebują więcej danych do nauki

 Autoencoder można doszkolić

 Autoencodery potrafią wykrywać złożone zależności

 Dlaczego "vs" jeśli można "stacking" ?!



# Bibliografia

<https://www.kaggle.com/jsrojas/ip-network-traffic-flows-labeled-with-87-apps/home>

<http://afitts.github.io/2019/01/26/netflow/>

<https://www.sciencedirect.com/science/article/pii/S1084804520303519>

<https://github.com/knowledgedefinednetworking/Unveiling-the-potential-of-GNN-for-network-modeling-and-optimization-in-SDN>

<https://knowledgedefinednetworking.org>

<https://www.kaggle.com/jsrojas/ip-network-traffic-flows-labeled-with-87-apps/home>

Python  
K N N  
Detekcj  
Kaggle  
Pandas  
Outlier  
Detection  
Jupiter  
AI ML

Uczenie  
Maszynowe  
NumPy  
tensorflow  
Autoencoder  
Il you zombie ML  
Anomalii  
Isolation Forest  
brakuje RAMu  
1h na ten  
slajd