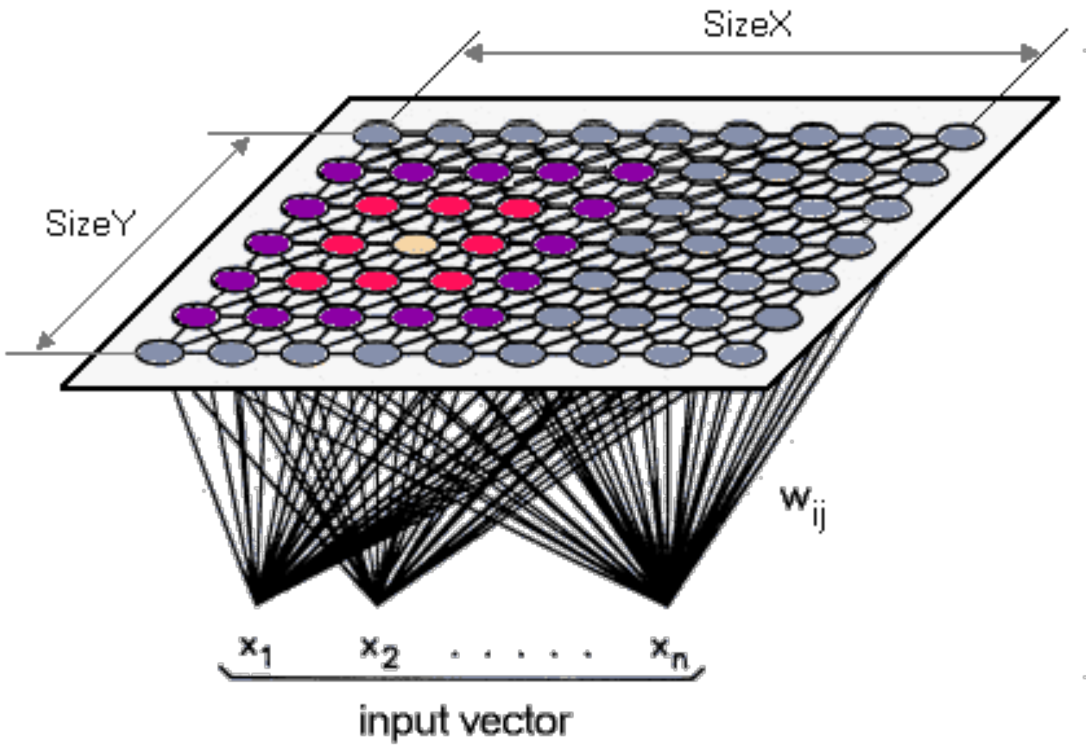
SOM FOR IDS

INTRODUCTION

- Self-Organizing Maps (SOM) is a type of artificial neural network invented by Teuvo Kohonen in the 1980s.
- They are trained with unsupervised learning, which means that the training data is not labeled.
- The goal is dimensionality reduction and visualization of data from high dimensionality in a map
- They differ from other neural networks because they are based on competitive learning.
- They are used for clustering, pattern recognition, feature extraction, and visualization.
- In the field of cybersecurity, they are used in anomaly detection, threat classification, and network behavior analysis.

SOM ARCHITECTURE



- SOM neurons are organized in a two-dimensional grid, in which each neuron has a weight vector of the same size as the input vector. This grid represents the space into which the input data is projected.
- Weight vectors link each neuron to all input attributes. Updating these vectors through learning.
- The spatial structure of the grid helps to visualize and understand the similarities and differences within the dataset, with rectangular or hexagonal configurations.

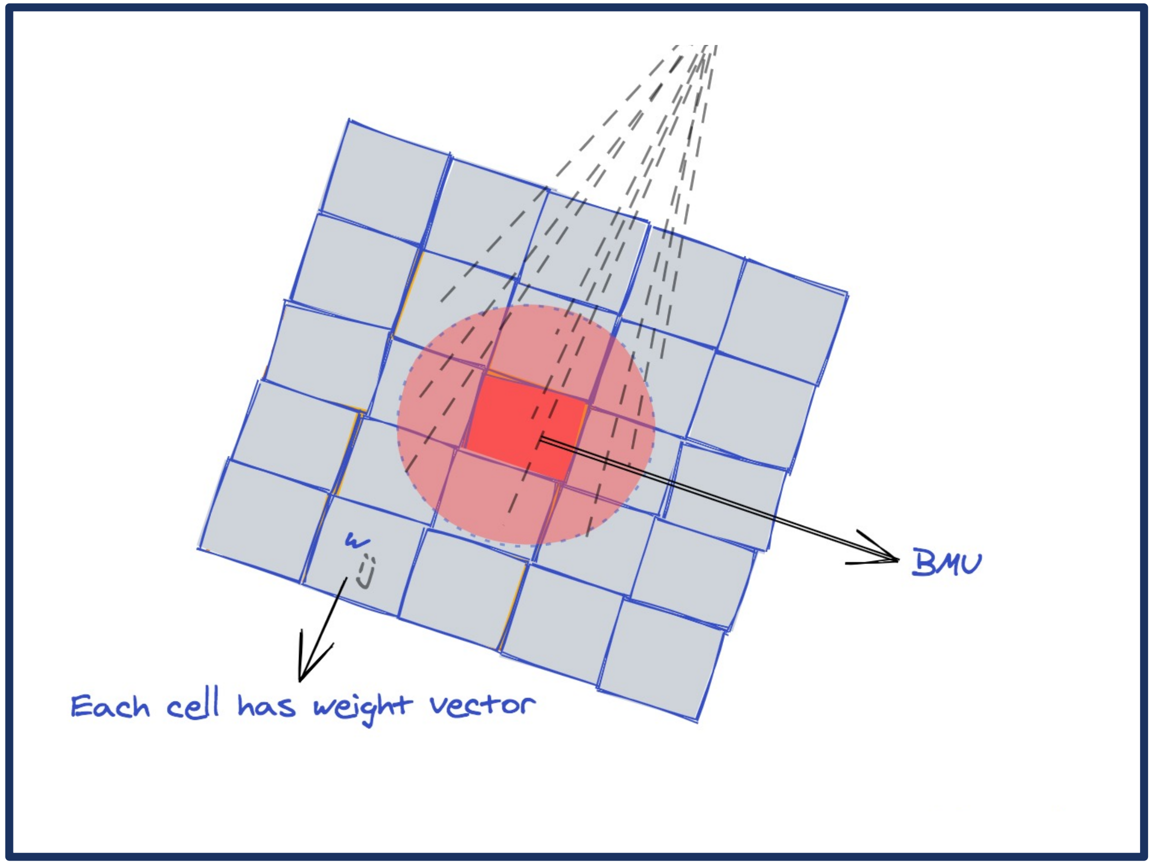
OVERVIEW OF THE LEARNING PROCESS

- SOM Objective: Transform complex, high-dimensional data into simplified representations while preserving topological relationships. Allowing the user a greater understanding.
- Stages of the Algorithm: The learning process consists of four phases - Initialization, Competition, Cooperation, and Adaptation.
- The algorithm proceeds iteratively, presenting sequences of input vectors and updating the weights of neurons to reflect structures and patterns in the data.
- Learning proceeds until no more substantial changes are observed in the map, indicating that the algorithm has reached an equilibrium.

INITIALIZATION PHASE

- Choice of hyperparameters, before learning: the initial learning rate, the number of iterations, the size of the grid, the proximity function are defined and the metric to be used for competition between neurons is determined.
- These choices affect the effectiveness of the training and the quality of the resulting map. In addition, as far as the initialization of the weight vector is concerned, there are 2 strategies:
- Random initialization: The weight vectors of each neuron in the grid are initialized with random values. This ensures a non-prejudicial start for the process of self-organization.
- Data-driven initialization : Good initialization can accelerate algorithm convergence and improve map representativeness.

COMPETITION STAGE (1)



- Each time a new input vector is introduced to the SOM, a crucial phase of competition between neurons begins. It is at this time that you decide which neuron will become the representative of that specific input.
- Data Representation: Competition ensures that each input is represented on the map by the most suitable cell, preserving the similarities and differences between the different inputs.

COMPETITION STAGE (2)

- The SOM algorithm identifies the Best Matching Unit (BMU), based on a metric such as the Euclidean (or, e.g., Manhattan) distance between the input vector and the weight vectors of each neuron. The neuron with the shortest distance is elected as the BMU.

- Euclidean Distance Formula:$d(\vec{x}, \vec{w}) = \sqrt{\sum_{i=1}^{n} (x_i - w_i)^{\,2}}$

  Where: $\vec{x}$ represents the input vector and$\vec{w}$ the weight carrier of the candidate neuron to become BMU.

- The BMU is the neuron that bears the greatest resemblance to the current input

- The selection of the BMU is crucial because it determines how the data will be represented on the map.

COOPERATION AND ADAPTATION PHASE(1)

- The Best Matching Unit (BMU), once identified, initiates a phase of cooperation with surrounding neurons. This interaction propagates beyond the BMU, affecting a surrounding area defined as the neighborhood.
- The neighborhood is the area around the BMU where neurons undergo an adjustment of their weights to better align with the current input vector. The magnitude of this change is regulated by a variable learning rate, which is usually progressively reduced during the training process.
- The proximity function (e.g., Gaussian or Triangle) determines how intensely the weights of neurons in the neighborhood are updated in response to the input vector. This function is strongest for neurons immediately adjacent to the BMU and decreases as distance within the neighborhood increases.
- Adjustment Formulas: The weights of the neurons near the BMU are updated according to the formula:

$$w_v(t+1) = w_v(t) + \theta(u,v,t) \cdot \alpha(t) \cdot \left(x(t) - w_v(t)\right)$$

$w_v(t)$:vettore peso del neurone v ; $\theta(u,v,t)$:funzione di vicinanza tra il BMU u e il neurone v ;
$\alpha(t)$ tasso di apprendimento, $x(t)$ vettore di input.

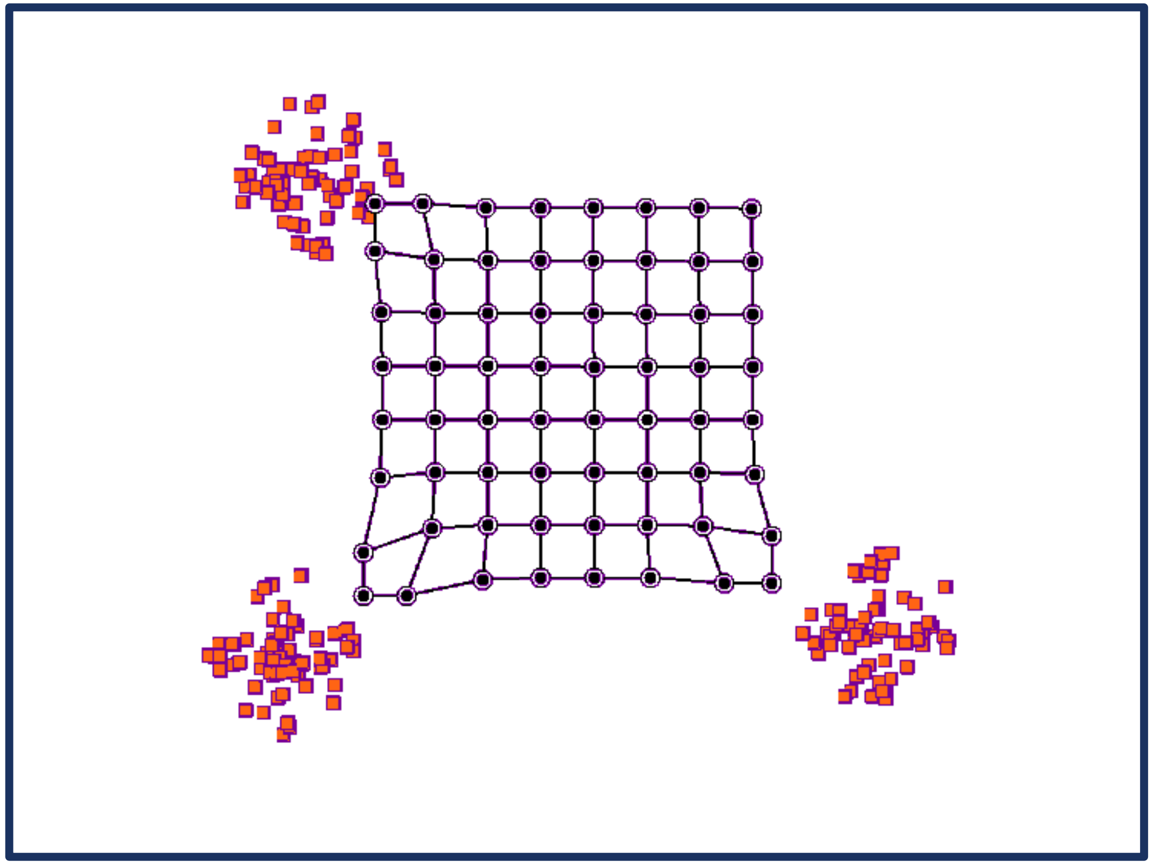
COOPERATION AND ADAPTATION PHASE(2)

- The goal of the cooperation phase is to make the map "elastic", allowing neurons to adapt not only to the current input but also to similar inputs that may be presented in the future.
- Through cooperation, the SOM maintains the topology of the input data. Neurons representing similar inputs are located close together on the map.
- Cooperation is essential for learning in a SOM. It is through this process that the map learns to organize itself so that similar patterns of input result in neurons firing in nearby regions of the map.

SUMMARY

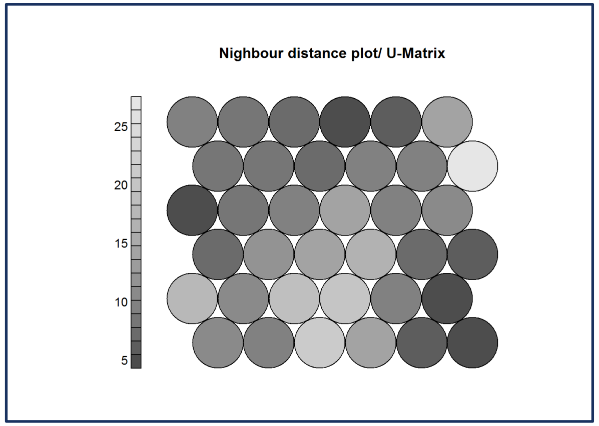
- Initialization: You start by defining the hyperparameters and assigning the initial weights to the neurons on the map.
- Competition: The network is subject to competition between the neurons on the map to determine the Best Matching Unit (BMU), which is the neuron with the weights most similar to the input vector according to a distance metric.
- Cooperation: Neurons in the neighborhood increase their similarity to the input vector. The size of the neighborhood, defined by a radial function, tends to decrease over the course of iterations.
- Adaptation: The BMU and its neighbors adjust their weights to get even closer to the input vector. This adaptation phase is driven by a learning rate that reduces over time, making the changes finer as the algorithm progresses.
- Iteration: The phases of Competition, Cooperation, and Adaptation are repeated over numerous cycles. Any presentation of an input vector and the consequent fit of weights is called an "epoch." With multiple epochs, the map self-organizes, until a convergence is reached, where changes in the weights of the neurons become minimal, indicating that the map has stabilized and adequately represents the input domain.

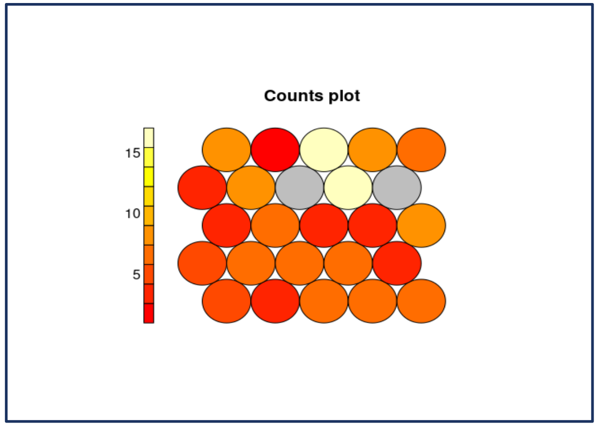
REMARKS

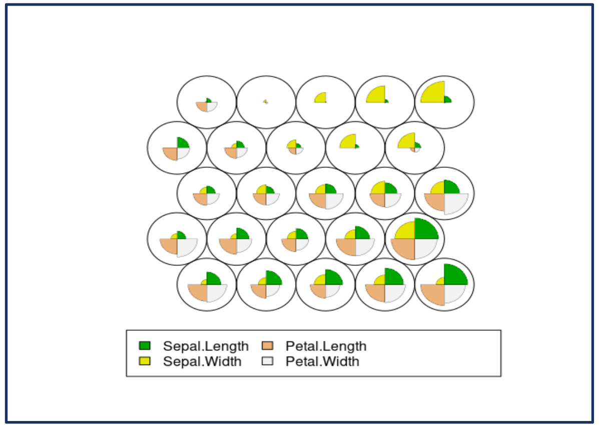


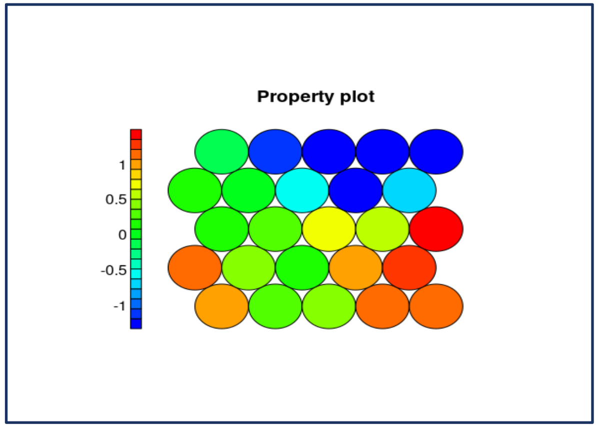
- As a result of the learning process, the weight vectors associated with each neuron are updated to optimize each neuron's ability to more accurately represent input data.
- The distance between the neurons in the map does not change physically, because the grid is fixed.
- Updating the weights changes the 'functional distance', which is used to indicate how similar the representations are in representing a given input on the map.
- Neurons representing similar data are considered close in terms of functional distance.

SOM REPRESENTATIONS









- After training the net, there are various ways to visualize it, the most famous are:
- Distance between Neighbors (U-Matrix): represents the functional distance between each node and its neighbors. Areas with a short distance indicate groups of similar nodes.
- Node Count: Displays how many samples are mapped to each node on the map. It can measure the quality of the map. Empty nodes indicate that the size of your map is too large for the number of samples.
- Node Weight Vectors/Codebooks: These are collections of weight vectors that represent input variables in a normalized way. They are visualized with fan diagrams to highlight the relevance of each variable in the different nodes, giving an insight into the distribution of the data in the map.
- Heatmaps: show the distribution of specific variables on the SOM map, helping to identify patterns and areas of interest while keeping the position of samples fixed.

EVALUATION OF THE SOM

- The main quality measures used to evaluate SOMs are quantization error (QE) and topographic error (TE):
- Quantization Error (QE): Measures the average distance between data points and the map nodes to which they are mapped. Lower values indicate better model fit. QE is useful for comparing different maps.

  $QE(M) = \frac{1}{n}\sum_{i=1}^{n}\|\varphi(x_i) - x_i\|$ dove:

  n è il numero di punti dati nei dati di allenamento, x_i rappresenta il *i*-esimo punto dato, $\varphi(x_i)$ è il nodo della mappa SOM a cui il punto x_i è mappato, $\|\varphi(x_i) - x_i\|$ calcola la distanza in base alla metrica scelta per esempio euclidea tra il punto dato e il suo nodo corrispondente nella mappa.

- Topographic Error (TE): Assesses the map's ability to preserve the topological structure of the input space in its low-dimensional output space. TE focuses on local mapping discontinuities, counting as errors cases where the best matching nodes are not close to each other.
- $\mathrm{TE}(M) = \frac{1}{n}\sum_{i=1}^{n} t(x_i)$ dove:

  $t(x)$ è una funzione che vale 0 se l'unità che meglio corrisponde ($\mu(x)$) e la seconda migliore corrispondente ($\mu_0(x)$) a un punto dato x sono vicine (ad esempio, sono vicini nella mappa), e 1 altrimenti.

SOM IMPLEMENTATIONS IN PYTHON

- The implementation of Self-Organizing Maps (SOM) in Python can be made accessible and flexible through different libraries, each with its own strengths and ideal use cases.
- MiniSom is a practical library for fast prototypes and exploratory analysis, with easy setup and visualization.
- Somoclu: stands out for handling large datasets on advanced hardware, offering parallelization.
- SUSI is suitable for in-depth assessments of the impact of learning models.
- The choice of SOM library in Python depends on the size of the data (Somoclu is great for large sets), need for visualization (MiniSom and SUSI for advanced details), and user level (MiniSom ideal for beginners).

CHOICE OF LIBRARY

- I have selected the libraryMiniSom to implement SOMs because, according to the Snyk.io It showed excellent maintenance and safety with a score of 77/100, higher than Somoclu (51/100) and SUSI (63/100), considering aspects such as safety, popularity and community support. This score helps you choose reliable and well-supported libraries. BasicUsage Minisom

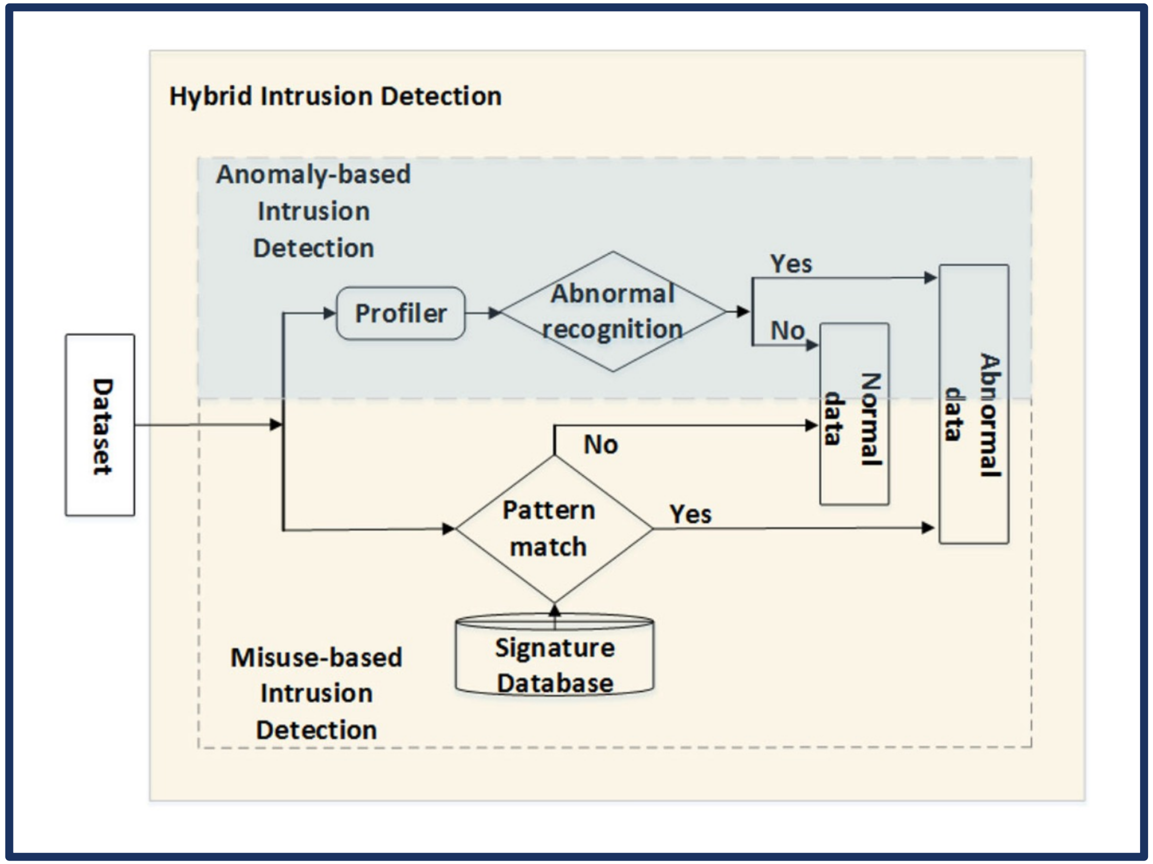
- Examples of project implementations with Minisom:

1. HandwrittenDigits
2. Identifying Breast Cancer Clusters
3. Fraud Detection

SOM FOR IDS

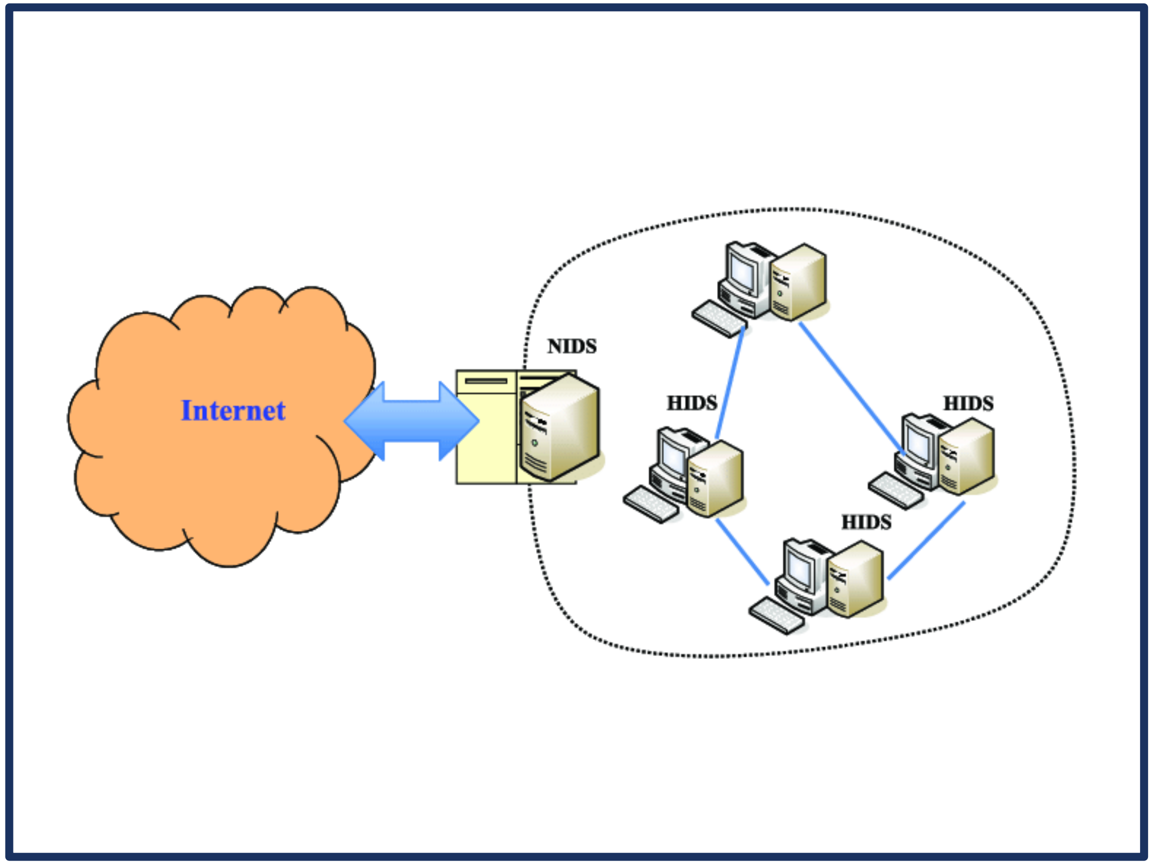
- SOMs can be used in a variety of fields such as medicine, marketing, bioinformatics, and financial analysis. But we will agree on the use of this neural network in Cybersecurity, more specifically in IDS.
- IDS is used to identify computer intrusions into computer networks and has a real-time response. It gathers information from different points and performs analysis to uncover security breaches.
- There are different types of intrusion detection systems, based on their functionality and the approach with which they manage intrusions and anomalies, the two main ones are:
- Signature-based (misuse-based) systems contain a database of signature-generated systems that are used to recognize existing malicious entities.
- Anomaly-based systems maintain a baseline of a system's normal behavior, which is used to recognize whether a system's behavior deviates in any way from this baseline.

SIGNATURE-BASED, ANOMALY-BASED E HYBRID



- Most of the studies have been carried out on systems based on anomalies.
- There are only a few systems that can be considered signature-based in the traditional sense. All of these systems are hybrid systems, combining both anomaly-based and signature-based techniques in order to achieve the best possible detection capabilities.

HIDS E NIDS

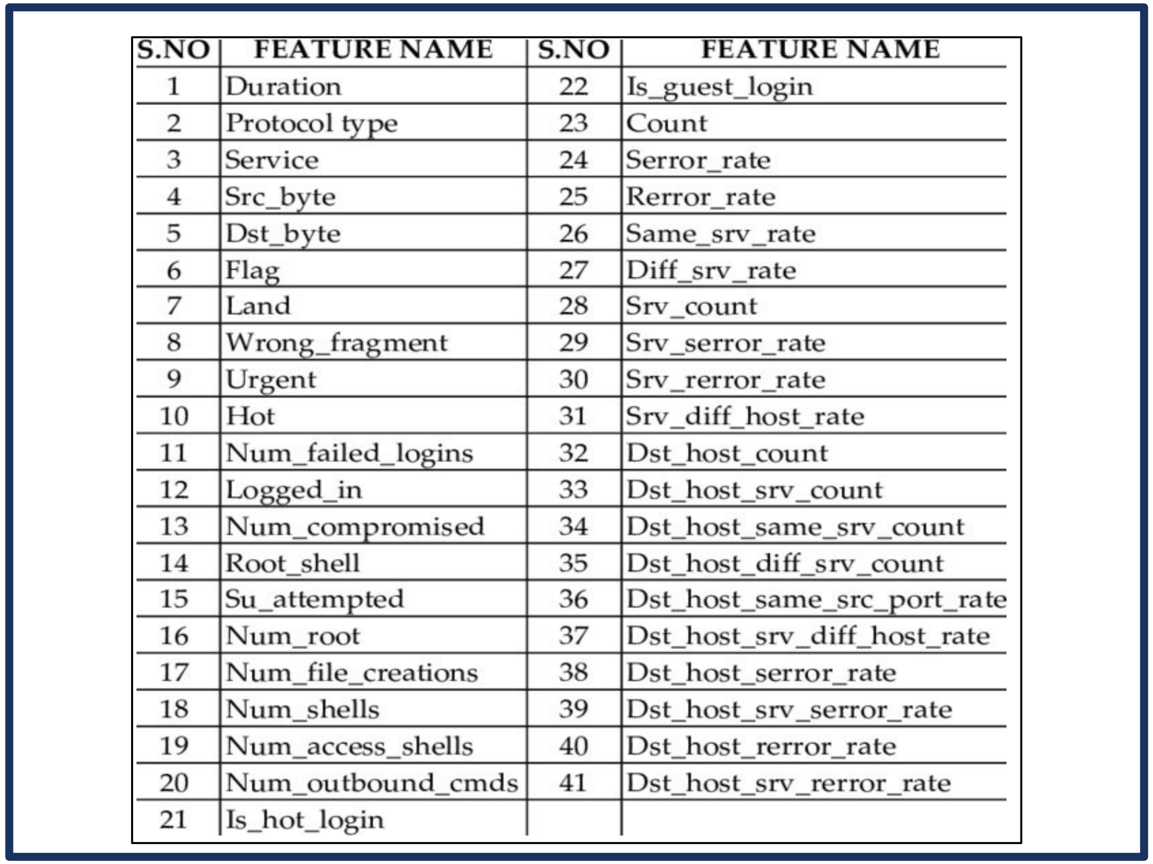


- The second group of categories distinguishes systems based on the type of information they monitor. These systems can be divided into:
- HIDS collect information from the local host system, such as system logs, file systems, user behavior, CPU and memory usage. They generally protect the system in which they reside.
- NIDS monitor network traffic and collect the original network packets with packet capture tools. Next, they analyze the packet header information and payloads to detect threats.
- NIDS perform real-time analytics and prevent online attacks, while HIDS conduct post-done analytics to prevent future attacks.

TYPES OF ATTACKS IN THE NETWORK

- Most of the research done using SOMs is based on the detection of intrusions into the network. There are four main categories of attacks in the network. Each attack on a network can be classified into one of these groups:
- Denial of Service (DoS): An attack in which the hacker makes memory resources occupied to serve legitimate network requests, thus denying users access to a machine, such as apache, smurf, Neptune, ping of death, mail bomb, UDP storm, etc.
- Remote to User attacks (R2L): A remote-to-user attack is an attack in which a user sends packets to a machine over the Internet, without having access, in order to expose vulnerabilities in the machine and exploit privileges that a local user would have on the computer, e.g. xlock, guest, XNSNOOP, PHF, Sendmail Dictionary, etc.
- User to Root attacks (U2R): These attacks are exploitations in which the hacker starts in the system with a normal user account and attempts to abuse vulnerabilities in the system in order to gain super user privileges, such as perl, xterm.
- Probing: An attack in which the attacker scans a machine or network device to determine weaknesses or vulnerabilities that could later be exploited to compromise the system. This technique is commonly used in data mining, e.g., satan, saint, portsweep, mscan, nmap, etc.
- The purpose of the classifiers in the IDS is to identify attacks from all four groups as accurately as possible.

DATASET KDD CUP’99



S.NO	FEATURE NAME	S.NO	FEATURE NAME
1	Duration	22	Is_guest_login
2	Protocol type	23	Count
3	Service	24	Serror_rate
4	Src_byte	25	Rerror_rate
5	Dst_byte	26	Same_srv_rate
6	Flag	27	Diff_srv_rate
7	Land	28	Srv_count
8	Wrong_fragment	29	Srv_serror_rate
9	Urgent	30	Srv_rerror_rate
10	Hot	31	Srv_diff_host_rate
11	Num_failed_logins	32	Dst_host_count
12	Logged_in	33	Dst_host_srv_count
13	Num_compromised	34	Dst_host_same_srv_count
14	Root_shell	35	Dst_host_diff_srv_count
15	Su_attempted	36	Dst_host_same_src_port_rate
16	Num_root	37	Dst_host_srv_diff_host_rate
17	Num_file_creations	38	Dst_host_serror_rate
18	Num_shells	39	Dst_host_srv_serror_rate
19	Num_access_shells	40	Dst_host_rerror_rate
20	Num_outbound_cmds	41	Dst_host_srv_rerror_rate
21	Is_hot_login		

- Most intrusion detection research uses the KDD Cup'99 dataset. Although KDD99 has some shortcomings, this dataset is the first and reliable reference dataset.
- The KDD99 contains four types of attacks, which are DOS,R2L,U2R,PROB with five million records. KDD99 contains 42 elements. The last element is a label, which indicates whether the data was normal or represented an attack.
- KDD99 does not reflect actual traffic statistics due to its simulated origin.

DATASET NSL-KDD

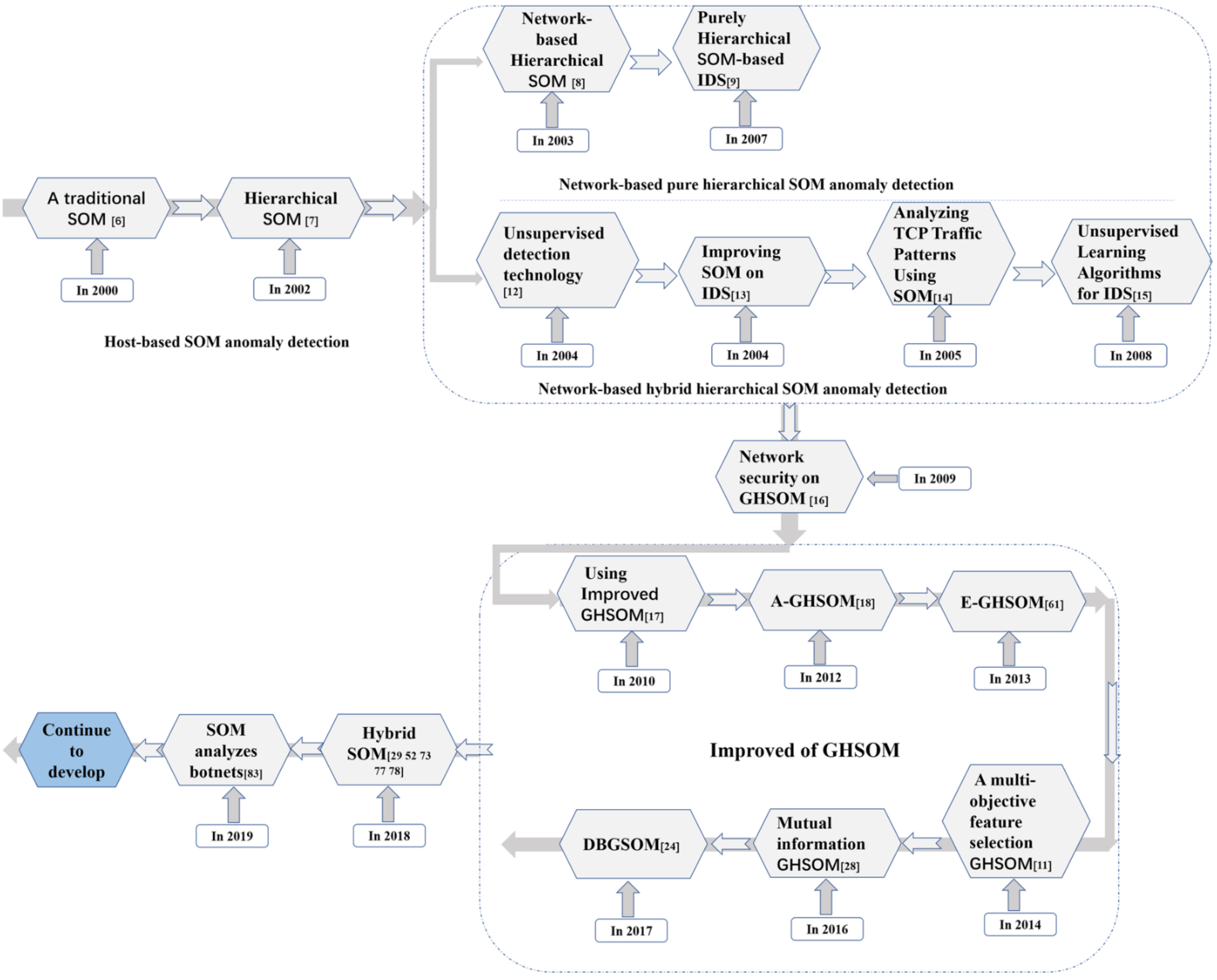
- KDD Cup '99 features a large number of redundant entries that make it difficult to categorize the remaining records.
- A new NSL-KDD dataset has been suggested to address these concerns. The NSL-KDD dataset consists of a small number of features from the KDD Cup '99 dataset that are not redundant in the training set or duplicated in the test set. The compelling reasons to use the dataset in experiments are:
- Removing duplicate data from the training set allows classifiers to be more unbiased when dealing with increasingly frequent records.
- Training and test sets contain enough instances to allow testing on the entire set without the need to select a small piece at random.
- The number of records in this dataset is 150,000 records

MORE MODERN DATASETS

- Although KDD99 and NSL-KDD are often used in research, these datasets have some limitations, such as a lack of representation of the most recent attacks.
- For this reason, newer datasets such as CICIDS2017 and CSE-CIC-IDS2018 have caught on, as they provide a more current and detailed view of security threats.
- Both have millions of instances and a rich variety of features, allowing for deeper modeling and analysis.
- Although they appear to be better, they have some disadvantages:
- They require more computational resources for processing due to the vast amount of data and variety of attacks
- Require significant pre-processing before you can use the raw data effectively.

RISULTATI OTTENUTI DALLE SOM E VARIANTI IN IDS

- ReferringA Survey on the Development of Self-Organizing Maps for Unsupervised Intrusion Detection SOMs have some decent results in this area, but variants of SOMs have been developed that are much more formatted for this use case, such as:
- HSOMs introduce a hierarchical structure that improves computational efficiency and represents data in a more structured way, allowing for easier identification of patterns and anomalies in network traffic.
- GHSOMs, on the other hand, extend HSOMs by allowing maps to grow dynamically depending on the complexity of the data, offering an even more adaptable and scalable model for real-time intrusion detection. GHSOM, and its optimizations, offer significantly better performance at detecting intrusions than the original SOM architecture, underscoring the importance of continuous innovation in cybersecurity to adapt to evolving threats.



A traditional SOM [6]

In 2000

Hierarchical SOM [7]

In 2002

Host-based SOM anomaly detection

Network-based Hierarchical SOM [8]

In 2003

Purely Hierarchical SOM-based IDS[9]

In 2007

Network-based pure hierarchical SOM anomaly detection

Unsupervised detection technology [12]

In 2004

Improving SOM on IDS[13]

In 2004

Analyzing TCP Traffic Patterns Using SOM[14]

In 2005

Unsupervised Learning Algorithms for IDS[15]

In 2008

Network-based hybrid hierarchical SOM anomaly detection

Network security on GHSOM [16]

In 2009

Using Improved GHSOM[17]

In 2010

A-GHSOM[18]

In 2012

E-GHSOM[61]

In 2013

Improved of GHSOM

A multi-objective feature selection GHSOM[11]

In 2014

Mutual information GHSOM[28]

In 2016

DBGSOM[24]

In 2017

Hybrid SOM[29 52 73 77 78]

In 2018

SOM analyzes botnets[83]

In 2019

Continue to develop

SOM development process

Methods	DR	FR	Features number	Disadvantage	Advantages
A raw SOM architecture [7]	89%	4.6%	6	The hierarchical relationship of data is not fully mapped. The detection rate is not high and the false alarm rate is high.	Using only the six most basic features, it achieves a detection rate comparable to 41 features.
A 2 or 3 hierarchical SOM architecture [8]	90.4%	1.38%	6 and 41	It takes a lot of experiments to get the best results. Use static hierarchical SOMs architecture.	Different characteristic data correspond to different SOM. Achieve the ideal detection rate with a small number of features.
Growing Hierarchical Self-Organizing Map (GHSOM) [15]	99.99%	3.73%	41	It results in a high false alarm rate, which may be caused by the small amount of attacking sample data and the poor accuracy of SOM mapping.	High detection rate.
Improved GHSOM [16]	96.2%	–	21	The detection rate of U2R and R2L is not high.	Obtain higher detection rate and improve the stability of intrusion detection.
ine an adaptive GHSOM (A-GHSOM) [17]	99.63%	1.8%	41	The classification calculation has high complexity, and there are redundant and irrelevant data, which affects the performance of the classifier.	Adapt online to changing exception detection.
A multi-objective feature selection HSOM [10]	99.6%	4.32%	41	The false positive rate is still relatively high compared to other methods.	Reduce the computational complexity, higher detection rate.
	99.12%	2.24%	25		
I–NGSA–III + GHSOM-pr [26]	99.37%	–	20	The detection rate of U2R and R2L is still not high.	For classes with fewer instances, the classification accuracy is higher. It can achieve higher detection accuracy with lower computational complexity.
GHSOM + mutual information [27]	97.73%	–	41	This method is only applicable to the entire cluster, and there are still some abnormal attacks that cannot be effectively clustered.	The detection rate of U2R and R2L is greatly improved.
DR is Detection rate, FR is False positive rate.					

SOM NETWORKS AND VARIANTS VS OTHER ANNS

- In the field of IDS, SOM networks and its variants were compared with other ANNs that are not based on competitive learning.
- The non-competitive Anns were found to be more accurate than the best version of the Som by about 1-3%.
- Anns that are not based on competitive learning are considered black boxes, as it is difficult for humans to understand the model's choice.For this reason, in the following article:Explainable Intrusion Detection Systems Using Competitive Learning Techniques, SOMs and variants are considered as a valid choice to be able to implement XIDS (Explainable IDS), allowing experts in the field a better understanding of the choice made by the model, since SOMs are white boxes. They can become valuable consultation tools for experts and analyses.