APPLICATION OF CLUSTERING METHODS FOR DETECTING NETWORK THREATS

The objective of this article is to explore the possibilities of identifying network attacks using cluster analysis methods. Density-based spatial clustering of applications with noise (DBSCAN) and the ordering points to identify the clustering structure (OPTICS) algorithm have been implemented on data representing a simulated network. The findings reveal the most effective algorithmic combinations for distinguishing between regular connections and data associated with network intrusions.

Keywords: DBSCAN, OPTICS, clustering, principal component analysis, Calinski-Harabasz index, confusion matrix.

ЗАСТОСУВАННЯ МЕТОДІВ КЛАСТЕРИЗАЦІЇ ДЛЯ ВИЯВЛЕННЯ МЕРЕЖЕВИХ АТАК

Вторгнення у мережі стають суттєвою проблемою, що стимулює до проведення досліджень для виявлення таких несанкціонованих спроб та розробки відповідних методів для їх своєчасного усунення. Метою даної роботи є дослідження можливостей ідентифікації мережевих атак методами машинного навчання. Нещодавні дослідження показали, що методи машинного навчання, зокрема, методи кластерного аналізу, є дієвими у виявленні аномалій у мережевому трафіку, що можуть бути індикаторами атак або відмов у системах. Алгоритми кластеризації, на відміну від інших методів машинного навчання, не вимагають наявності навчальних наборів даних, які репрезентують усі можливі варіанти атак, що, у свою чергу, дозволяє виявляти аномалії, які можуть бути новими та раніше невідомими.

У роботі розглянуто та реалізовано алгоритм просторової кластеризації з припиненням шому (DBSCAN) та алгоритм виявлення кластерної структури упорядкованих точок (OPTICS). Дані, які використовуються під час аналізу представляють імітовану мережу, в якій наявні як звичайні з'єднання, так і дані, які відповідають мережевим вторгненням. Для обчислення відстаней між об'єктами застосовано Евклідову, Манхеттенську та відстань Канберри. З метою візуалізації реалізовано метод головних компонент для зменшення розмірності даних. За результатами побудовано графік об'єктів з набору, перетворенных методом головних компонент. Оцінено якість кластеризації та оптимальні значення кількість кластерів на основі індексу Calinski-Harabazs, матриці невідповідностей, яка буде завершена порівнянням отриманих результатів з вихідними даними імітованої мережі.

© Kiseleva O.M., Prytomanova O.M., Filat O.A., 2023
Introduction. Network traffic analysis is a complex task that encompasses various aspects. Typically, network data analysis refers to a set of technologies and methods that enable the real-time collection, processing, classification, monitoring, and modification of network packets. Despite network traffic classification being a fairly specific research area, the goals of works in this field are not identical. The purpose of this work is to explore the possibilities of identifying network attacks using machine learning methods, which are considered the most favorable method for network traffic classification [1]. Recent research [2][3] has shown that machine learning methods, particularly clustering analysis methods, are effective in detecting anomalous patterns in network traffic that may be indicators of attacks or system failures. Unlike other machine learning methods, clustering algorithms do not require the presence of a training dataset representing all possible attack scenarios, allowing the detection of anomalies that may be new and previously unknown.

Problem Statement. This study aims to examine the effectiveness of clustering analysis methods in identifying network attacks. The problem formulation, taking into account the subject area, involves implementing the following clustering algorithms: density-based spatial clustering of applications with noise (DBSCAN) and the algorithm of ordering points to identify the clustering structure (OPTICS). The data used for analysis is sourced from a machine learning repository [4] and represents a simulated network, encompassing both regular connections and data corresponding to network intrusions.

To calculate the distances between the points, it is required to apply the Euclidean, Manhattan, and Canberra metrics. For visualization purposes, implement the principal component analysis method to reduce the dimensionality of the data [5]. Based on the results, construct a plot of objects from the set transformed by the principal component method. Evaluate the clustering quality and determine the optimal number of clusters (or other input parameters provided to the algorithms) based on the Calinski-Harabasz index and a confusion matrix, which is constructed by comparing the obtained results with the original data of the simulated network (whether a certain data row is an attack or 'normal' traffic).

Research Methods and Algorithms. The general formulation of the clustering analysis task is that the dataset \( X = \{x_1, \ldots, x_n\} \) should consist of \( n \) elements (considering the subject area, it represents the number of traffic sessions). Each traffic session should be described as an \( m \)-dimensional point in Euclidean space \( x_i : x_i = (x_{i1}, \ldots, x_{im}) \). The goal is to partition the traffic sessions, i.e., the dataset \( X = \{x_1, \ldots, x_n\} \) into \( K \) clusters: \( C = \{C_1, \ldots, C_k\} \). At the same time:

1. Each cluster should have at least one point;
2. Two different clusters should have no common elements;  
3. Each point should belong to some cluster.  

In DBSCAN, clustering is understood as the distribution of a given set of data points into subgroups, each of which, as much as possible, is homogeneous. The DBSCAN clustering method is fundamentally based on the grouping of certain objects according to their intra-group 'connectivity'.  

The density of points for a point $X$ is determined by two parameters. The first one is $\alpha$ – the radius of the "neighborhood" of point $X$. Then, the set $M_\alpha(X)$ includes such points as $f_i, (i = 1, ..., n)$, for which the following inequality holds:  
$$ \text{dist}(X, f_i) \leq \alpha, \quad (i = 1, ..., n). $$  

The $\text{dist}(X, f_i)$ function defines the distance between objects in the dataset $D$. This distance, within the scope of this work, is computed using Euclidean, Manhattan, or Canberra distance.  

The second parameter defining point density is $\text{MNP}$ – the minimum number of points located closest to a given point within a certain radius $\alpha$. A point $f_i, (i = 1, ..., n)$ is considered as a surrounded point (according to $\alpha$ and $\text{MNP}$) if  
$$ M_\alpha(X) \geq \text{MNP}. $$  

This means that a point $f_i, (i = 1, ..., n)$, is surrounded if the number of "neighboring" points in the data sample $D$ is greater than or equal to the value of the $\text{MNP}$ parameter.  

A point $X$ is directly reachable by density from a point $f$ (with appropriate $\alpha$ and $\text{MNP}$) if the point $X \in M(X)$, i.e., the point $X$ is one of the points $f$ in another environment (neighborhood) where $f$ is a surrounded point.  

Reachability by density is a transitive closure of density reachable points.  

Point $X$ is density connected to point $f$ if there exists a point $e$ such that both points $X$ and $f$ are density reachable from point $e$.  

A cluster formed based on the density distribution of objects should satisfy the properties of maximality and connectedness. In this case, a cluster is understood as a non-empty subset of points $G$ from the data set $D$ that satisfies the aforementioned properties, where maximality is interpreted as follows: if $X \in G$ and $f$ is density-reachable from point $X$, then $f \in G$. This means that both points belong to the same cluster.  

The connectedness property indicates that each object in the subset $G$ is density connected to all objects in the cluster (with given $\alpha$ and $\text{MNP}$). All objects in the data set $D$ form a collection of subsets:  
$$ D = \{G_1, G_2, ..., G_n, N\}, $$  
where $G_1, G_2, ..., G_n$ are clusters formed based on density, and $N$ is a subset whose objects do not belong to any of the subsets $G_1, G_2, ..., G_n$.  

The implementation of the DBSCAN algorithm [6] can be divided into two stages. Firstly, from the entire data set $D$ it is necessary to identify the points that are density reachable. Then, perform the following procedure for each object $X$ from the data set $D$:  

1. Two different clusters should have no common elements;  
2. Each point should belong to some cluster.
1. Check if the current object belongs to any of the clusters.
2. Check if the current object is a density reachable point.

If the current object is a density reachable point, then all objects that can be density reached from the current object are connected to form a new cluster. Otherwise, if the object is not a density reachable point and is not reachable by density to any other object, then the current object is considered an outlier.

Ordering Points to Identify the Clustering Structure (OPTICS) consists of the following steps:

1. The algorithm takes input data, as well as values for $\varepsilon$ (neighborhood radius) and the minimum number of points. For each point, the accessibility distance is initially set as undefined. The algorithm also includes an ordered list and a list of processed points, which is initially empty and gets filled as points are processed [7].
2. Then, for each unprocessed point, the following is done:
   - Compute the $\varepsilon$-neighborhood of the point.
   - Mark the point $p$ as processed.
   - Move the point $p$ to the ordered list.
3. On the third step, it is necessary to estimate the core distance of point $p$. If it is not undefined (i.e., if it is a core point), proceed to step 3. If it is not a core point, move to the next unprocessed point, i.e., go to step 2.

For the core points, we initialize an empty priority queue, which is a queue where the most important values are read first. Then we call the update function, which sorts the priority queue based on accessibility distance. In the ordered priority queue, for each point, we get its neighbors. Then we mark point as processed and output it to the ordered list. If it is also a core point, it is necessary to expand the priority queue because clusters are likely to be close to each other and may belong to one larger cluster. Expanding the priority queue through updating means that more points will be added to the list of initial loads, ordered by accessibility distance.

To avoid the influence of scale (as input data has different scales), data normalization is applied in the work. This transforms feature values by scaling them to a range from 0 to 1. Each normalized feature element is computed using the formula:

$$x_{i,\text{norm}} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}},$$

where $x_{i,\text{norm}}$ is the normalized feature element, $x_{\min}$ is the minimum element, $x_i$ is the $i$-th element, and $x_{\max}$ is the maximum element.

The distance between points is computed using the following metrics:

1. Euclidean distance between points $x$ and $y$ is the length of the line segment $xy$ in Cartesian coordinates. If $x = \{x_1, x_2, \ldots, x_n\}$ and $y = \{y_1, y_2, \ldots, y_n\}$ are
two points in Euclidean space, then the length of the segment $xy$ is given by:

$$p(x, y) = \|x - y\| = \sqrt{\sum_{p=1}^{n} (x_p - y_p)^2}.$$  

2. Manhattan distance is the distance that is the average of coordinate differences. In most cases, this method leads to the same results as the ordinary Euclidean distance. However, for this method, the impact of individual large differences (outliers) is reduced because they are not squared:

$$p(x, y) = \sum_{p=1}^{n} |x_p - y_p|.$$  

3. Canberra distance is a metric function used for data distributed around the origin. The generalized equation is given by:

$$d^{CAD}(i, j) = \sum_{k=0}^{n-1} \left| \frac{y_{i,k} - y_{j,k}}{y_{i,k} + y_{j,k}} \right|.$$  

Calinski-Harabasz Index [8].

The fundamental idea of this evaluation is as follows. Let $\overline{d^2}$ be the average squared distance between elements in the clustered set, and $\overline{d^2_{ci}}$ is the average squared distance between elements in cluster $c_i$. Then the within group sum of squares is given by:

$$WGSS = \frac{1}{2} \sum_{i=1}^{c} (n_{ci} - 1) \overline{d^2_{ci}}.$$  

And the between group sum of squares is given by:

$$BGSS = \frac{1}{2} \left( (c - 1) \overline{d^2} + (N - c) A_c \right),$$  

where $A_c = \frac{1}{N - c} \sum_{i=1}^{c} (n_{ci} - 1) \left( \overline{d^2} - \overline{d^2_{ci}} \right)$ is the weighted average difference between distances from cluster centers to the overall set center. Then, the index formula is defined as:

$$VRC = \frac{BGSS}{N - c} \frac{\overline{d^2} + \frac{N - c}{c - 1} A_c}{WGSS} = \frac{BGSS}{N - c} \frac{\overline{d^2} + \frac{N - c}{c - 1} A_c}{1 - a_c},$$  

where $a_c = \frac{A_c}{d^2}$. The maximum value of the $VRC$ index corresponds to the optimal cluster structure.

**Analysis of the results.** Python was chosen as the programming language for implementing the software, and Jupyter Notebook was used as the interactive development environment with the cloud service Google Colaboratory. Various third-party libraries were employed during implementation, including:
- Scikit-learn (for cluster modeling and analysis).
- Pandas library for loading, preparing, analyzing, and manipulating indexed data.
- NumPy for processing multi-dimensional arrays and matrices, as well as large datasets for high-level mathematical functions.
- Matplotlib and Seaborn for data visualization.

The input data, as described in the problem statement, underwent normalization. During clustering with the DBSCAN algorithm, the optimal value for \( \varepsilon \) was found to be 3, and the minimum number of points was set to 10. The best interpretable results were obtained from clustering using the Manhattan metric and Canberra distance. The data was divided into 5 clusters using this algorithm.

![DBSCAN using Manhattan distance](image1)

![DBSCAN using Canberra distance](image2)

**Fig. 1.** Plot of data clustered using the DBSCAN method with the Manhattan distance and Canberra distance, along with the application of the principal component analysis method

Confusion matrix as an interpretation of clustering results. Two clusters correspond to attacks, while three clusters represent "normal" traffic.

![Confusion matrix](image3)

**Fig. 2.** Confusion matrix for data clustered using the DBSCAN method with the Manhattan distance (left) and using Canberra distance (right)

A similar interpretative outcome was achieved using \( \varepsilon = 3.4 \) and the Manhattan metric. However, in this case, the algorithm divided the data into 4 clusters.

Additionally, let's present the clustering quality results based on the Calinski-Harabasz index. Table 1 displays the clustering quality evaluation results depending on the value of \( \varepsilon \).
Despite the equally high index results for $\varepsilon = 3$ and $\varepsilon = 2.6$, the best interpretative outcome was achieved with $\varepsilon = 3$. It is also worth noting that when using Euclidean distance and the aforementioned input data, clustering is not possible (as the algorithm does not form data into at least 2 clusters).

During clustering with the OPTICS algorithm, $\varepsilon$ was set to 0.5. The best results were obtained using the Canberra distance, although quite accurate results were also obtained with Euclidean and Manhattan metrics. In all three cases, the algorithm clustered the data into 2 clusters. The results of these experiments are provided below.

![Confusion matrix for data clustered using the OPTICS method with Euclidean distance (left) and Canberra distance (right)](image)

**Fig. 3.** Plot of data clustered using the OPTICS method with Euclidean distance and Canberra distance, along with the application of the principal component analysis method

**Fig. 4.** Confusion matrix for data clustered using the OPTICS method with Euclidean distance (left) and Canberra distance (right)
Cluster quality assessments with the Calinski-Harabasz index depending on the parameter $\varepsilon$ are provided in the table below.

<table>
<thead>
<tr>
<th>OPTICS</th>
<th>Manhattan distance</th>
<th>Euclidean distance</th>
<th>Canberra distance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of formed clusters</td>
<td>Calinski-Harabasz index value</td>
<td>Number of formed clusters</td>
</tr>
<tr>
<td>$\varepsilon=0.5$</td>
<td>2</td>
<td>1065.08</td>
<td>2</td>
</tr>
<tr>
<td>$\varepsilon=0.2$</td>
<td>3</td>
<td>367.9</td>
<td>3</td>
</tr>
<tr>
<td>$\varepsilon=0.35$</td>
<td>3</td>
<td>1403.47</td>
<td>3</td>
</tr>
</tbody>
</table>

**Conclusions.** The tasks set for this study have been successfully accomplished. Specifically, Python software has been developed to conduct density-based spatial clustering of applications with noise and the algorithm of ordering points to identify the clustering structure in the context of network traffic data. Euclidean, Manhattan, and Canberra distances were employed for calculating distances between points. Additionally, the principal component analysis method was applied to reduce the dimensionality of the data. The evaluation of clustering quality involved the use of the Calinski-Harabasz index and a confusion matrix. The results identified the most effective algorithm combinations, enabling the most accurate differentiation between 'normal' traffic and network attack traffic.

**References**


Received 18.07.2023.