BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF TELECOMMUNICATIONS

ÚSTAV TELEKOMUNIKACÍ

SELECTED TECHNIQUES INVOLVED IN NETWORK TRAFFIC ANALYSIS

VYBRANÉ TECHNIKY ANALÝZY SÍŤOVÉHO PROVOZU

HABILITATION THESIS

HABILITAČNÍ PRÁCE

AUTHOR AUTOR PRÁCE	Ing. Václav Oujezský, Ph.D.

BRNO 2023

ABSTRACT

This habilitation thesis presents an overview of research findings focused on traffic analysis in data networks. Network traffic analysis, in general, is an integral part of modern network security systems. This thesis presents selected techniques used in traffic data analysis, the authors' research findings in this area, and the goals achieved, which reflect the published results. The thesis is divided into seven main chapters. The opening section states the objectives of the thesis and summarises the current state of the art in the field and the available technologies. Subsequently, methods of network traffic data acquisition and data format are presented. Next, the given work addresses the question of data processing for further analysis. This is followed by a chapter dealing with selected network traffic analysis techniques and monitoring options. The final chapter summarizes the whole work. All the presented results have been tested or verified in a laboratory environment. The actual text is written to be both scientifically and pedagogically valuable. The thesis is mainly based on the author's original research and development in the years following the defense of his Ph.D. thesis. All presented solutions have been published in impact factor journals, indexed journals, or presented at international conferences.

KEYWORDS

Analysis, Data, Network, Traffic, Techniques

ABSTRAKT

Tato habilitační práce prezentuje přehled poznatků z výzkumu zaměřeného na analýzu provozu v datových sítích. Analýza síťového provozu obecně, je nedílnou součástí soudobých síťových bezpečnostních systémů. V této práci jsou uvedeny vybrané techniky používané v analýze provozu, vlastní výzkumné závěry v této oblasti a dosažené cíle, které reflektují publikované výsledky. Práce je rozdělena do sedmi hlavních kapitol. Úvodní část stanovuje cíle práce a shrnuje současný stav vědeckého poznání problematiky a dostupných technologií. Následně je představen způsob získávání dat síťového provozu a jejich formát. Dále je v předložené práci řešena problematika zpracování dat pro další analýzu. Následuje kapitola zabývající se vybranými technikami analýzy síťového provozu a možnostmi jeho monitoringu. Závěrečná kapitola shrnuje celou práci. Všechny uváděné výsledky byly testovány nebo ověřeny v laboratorním prostředí. Vlastní text je psán tak, aby byl jak vědecky, tak pedagogicky přínosný. Práce je založena především na původním výzkumu a vývoji autora v letech po obhajobě jeho doktorské práce. Všechna prezentovaná řešení byla publikována v časopisech s impakt faktorem, indexovaných časopisech nebo prezentována na mezinárodních konferencích.

KLÍČOVÁ SLOVA

Analýza, Data, Provoz, Síť, Techniky

OUJEZSKÝ, Václav. *Selected techniques involved in network traffic analysis*. Brno, 2023, 85 p. Habilitation thesis. Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications.

Prepared using `thesis` package 3.03; `http://latex.feec.vutbr.cz`
Grammar checked by Writefull; `https://www.writefull.com/`

DECLARATION

I declare that I have written the Habilitation Thesis titled "Selected techniques involved in network traffic analysis" independentlyand using exclusively the technical references and other sources of information cited in the thesis and listed in the comprehensive bibliography at the end of the thesis.

As the author I furthermore declare that, with respect to the creation of this Habilitation Thesis, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll., Section 2, Head VI, Part 4.

Brno ....

author's signature

Václav Oujezský

Signed by: Václav Oujezský

Date: 2023-01-10 13:10:15 CET

Acknowledgement

There are many people I must thank for their help in preparing and developing my habilitation thesis. I owe a debt of gratitude to the following coauthors of the works that are included: Vladislav Škorpil, Tomáš Horváth, Michal Jurčík, Adrian Tomašov, Martin Holík, Pavel Novák. I would also like to express my gratitude to Michal Polívka, who directed me to the subject at the beginning of my research. I also warmly recognize the funds for helping to fund the research I have presented here. I want to thank the staff of the Institute of Telecommunications, who provided me with the necessary background for developing this thesis, as well as my colleagues and my students at the Faculty of Informatics, Masaryk University, for the same. Furthermore last but not least, love to my family.

Contents

List of Figures

Introduction

In recent years, the study of large-scale networks has become widespread. Especially with the Internet boom, the first opportunity to study large-scale networks has emerged. Today's wide area networks are increasingly a part of society, and society is more dependent on them than ever before – their functioning results from a combination of many technologies and many principles. In the search for efficiencies in their combined functioning and the search for ways to provide protection, researchers have begun to systematically analyze and characterize the dynamic patterns and evolution of these complex and highly heterogeneous structures. Their properties are usually manifested in connectivity patterns characterized by large fluctuations, scale-free properties, and non-trivial correlations such as high clustering and hierarchical ordering. The large size and dynamic structure of complex networks are closely related to graph theory and the ability to characterize the dynamic evolution of the basic elements of such a system. Advances in research on complex networks have sparked interest in the possible implications and consequences of their operation for the most important questions concerning various dynamic processes and functions. Typically, questions such as what is the impact of a major network outage, how can a critical error or attack affect their functioning, how can we effectively search for data in large information structures, how data traffic can be efficiently transferred, and many other questions.

The issue of computer network security is not fading in importance; on the contrary, interest in addressing security issues in modern communication networks is growing. Unsolicited traffic nowadays significantly affects the security of society. The current trend is to use a hybrid approach, where attackers typically mix several operations together to create many attack vectors. During these operations, attack vectors and protocol signatures are altered to deceive automated mitigation devices. Increasingly, mobile clients and smart devices are at the forefront of attackers' minds. The need to develop security devices that can respond in real-time to attacks and network traffic anomalies is undoubtedly a priority.

On the other hand, providing a general account of the study of traffic networks is impossible since each research domain is very specific. The research I have conducted after I obtained my Ph.D. degree in Telecommunications has mainly had in view two areas that involve traffic analysis in passive optical networks and methods used for traffic behaviour analysis. The research I have done is related to projects in which I have personally participated or proposed and I am a co-investigator; details of some of the projects are published in [1]. The research itself focused on two fundamental questions. The design of a system to enable the detection and analysis of traffic in communication networks is partially and recently focused on passive gigabit networks. The second question concerns the possibility of analyzing and verifying the data obtained from such network traffic. As a result of the research conducted, the habilitation thesis entitled *„Selected techniques involved in network traffic analysis“* unfolds in the following directions: research pertaining to methods for network traffic analysis in gigabit passive optical networks and the use of subparts of artificial intelligence, such as genetic algorithms, machine learning or data processing. One part of the research also includes the mentioned system design for data capture and analysis.

A specific area of research related to passive optical networks presents complex networks different from other types of networks based on Ethernet technology. Its complexity lies in its architecture and interference with other types of networks and its use has historically been for metropolitan area networks, but it has already started to be used more in the development of *5th Generation Mobile Networks* (5G) [2] due to the lack of fibre backhaul (and not only for that reason) with the very rapid development of new mobile networks requiring high-speed, low-latency connectivity. The popularity and use of these networks are constantly growing and, therefore, need to be addressed from all aspects.

A question I have often encountered is why any traffic analysis is actually needed in optical networks if everything that happened is visible in Ethernet networks, i.e., at the interface of two technologies. That statement needs to be corrected. The fundamental shortcomings of traffic analysis in optical networks were presented in [3]. Several key aspects that are directly related to the security of passive optical networks are mentioned.

From this, my research work has resulted in algorithms and methods applicable to traffic analysis in several types of networks. Another result is a concept of a comprehensive model of a system for traffic analysis. Both results of which are the content of this habilitation thesis. None of the results or proposed solutions presented in this thesis were published in my Ph.D. thesis or any previous theses. It contains, of necessity, only some references or insights that support the continuous progress of scientific work in the related area.

1 Thesis Overview

This chapter contains an overview of this thesis. Section 1.1 provides a summary of its primary objectives. In Section 1.2, the author's publishing activities and contribution to the scientific subject of the thesis are briefly discussed. Finally, Section 1.3 introduces the organizational structure of the thesis.

1.1 Goals

In this habilitation thesis, the main topic is traffic and data analysis methods in complex and heterogeneous networks and techniques that provide and support data analysis. The main idea of the work is laid out in several parts. This reflects the distinct focus of each of the sections that follow and identifies the relevant in-depth publications. I also provide a quick summary of the most pertinent connected works in each area here, and details are then given in the particular included publications. The aim of the thesis is to present the latest new knowledge in the field of network analysis techniques obtained from other experts and especially from the author's research activities. Among the objectives presented in this thesis are the following:

- To provide readers with resources that cover an overview of the basics of the techniques and methods used to analyze network traffic, both from a research and from a pedagogical point of view.
- To demonstrate the implementation of selected traffic analysis techniques in specific cases.
- To introduce and demonstrate traffic analysis capabilities from several different perspectives, such as traffic behavior analysis and traffic appearance, and introduce techniques used to increase the efficiency of data analysis and processing.
- To propose, design, and verify a new method for traffic behavior analysis and traffic analysis in passive optical networks.

1.2 Contribution and Relation to Author's Publications

The thesis content is written considering both pedagogical and scientific contributions. In order to provide a foundational understanding, an overview of the state of the art, and familiarity with the most recent scientific results, it should be helpful not only to specialists in the field of data traffic analysis but also to students interested in this topic. The scientific contributions of the author to the topic are listed at the beginning of each chapter for better transparency.

The present habilitation thesis manuscript covers the scientific progress I have achieved in the years 2017 to 2022, by exploring different topics in traffic analysis, techniques, and methods of detection of malicious traffic or detection of abnormal traffic, especially in passive optical networks, but not limited to this type of network. The main results and proposals presented in the thesis have been published in various international journals with impact factors, such as Sensors *International Standard Serial Number* (ISSN): 1424-8220, Electronics ISSN: 2079-9292, or are indexed by Web of Science and Scopus. In addition, I am also the primary author or coauthor of various publications of other topics related to telecommunication networks.

The research results described in the thesis are partially the subject of recent research projects *Ministry of the Interior of the Czech Republic* (MV ČR) 2017-2019 – "Detection of security threats on the active components of critical infrastructures" (setting up, co-investigator, researcher, programmer), MV ČR 2017-2020 – "Reduction of security threats at optical networks" (setting up, co-investigator, researcher, programmer), MV ČR 2019-2022 – "Deep hardware detection of network traffic of next-generation passive optical network in critical infrastructures" (setting up, co-investigator, researcher, programmer), *Technology Agency of the Czech Republic* (TA ČR) 2019-2024 – "Decentralized Control of Distribution System" (research team member), and from the activities related to the foreign collaboration, as is COST Action CA20120 2021-2025 (working group member) and last but not least, from cooperation with Assoc. Prof. Vladislav Škorpil, Dr. Tomáš Horváth, Brno University of Technology, or from cooperation with industrial partners. Current projects in the role of the lead investigator are MV ČR VK01030030 2023-2026 – "Data backup and storage system with integrated active protection against cyber threats", and MV ČR VK01030152 2023-2024 – "Android federated learning framework for emergency management applications".

1.3 The Organisation of the Thesis

The thesis is divided into seven basic chapters. The thesis is organised as follows:

- In Chapter 1, the thesis overview is given, including the goals and contributions in related research, projects, and publications.
- Chapter 2 presents the state-of-the-art in the field of network traffic analysis, giving a general overview of the techniques used for network analysis.
- Chapter 3 presents the topic related to the source of data for network traffic analysis and the data sources currently used as a source for network traffic analysis and processing techniques.
- Chapter 4 discusses the data preparation and selected process streamlining techniques and algorithms used.
- Chapter 5 deals with data traffic analysis techniques such as network traffic behaviour analysis, deep packet inspection, network frame structure analysis, or the use of artificial intelligence in traffic analysis.
- Chapter 6 summarizes current data monitoring concerns and discusses the possible monitoring of passive optical networks.
- Chapter 7 proposes a template for the verification and interpretation of traffic analysis results.
- The conclusion provides a review of the thesis and planned short and long-term future works.

❏ For better readability, *research articles are cited according to the citation standard.* Other references are given as a footnote in a simple format.

2 The State of the Art in Network Traffic Analysis

In computer science, the analysis[1] can be object-oriented or syntactic. In networking, *Network Traffic Analysis* (NTA) refers to the discovery and understanding of events that occur in the operation of network elements for the purpose of operation and protection. From research and experience[2], it can be concluded that network analysis generally, from a high-level point of view, consists of behavioral analysis or appearance analysis. These two can be analyses of an individual element or neighborhood (the surrounding environment or elements) and its connections (network traffic flows, network traffic connections), and these two again in real or relative time and space, Figure 2.1.

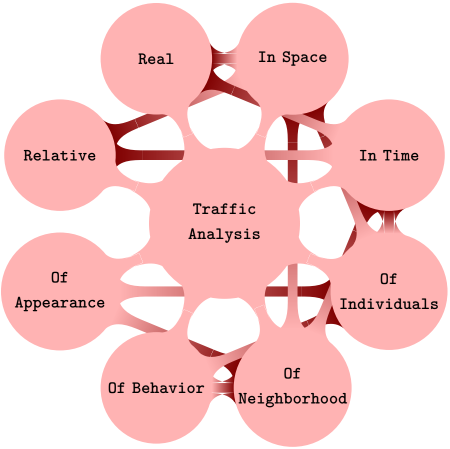


Fig. 2.1: The Functional Connections of Network Traffic Analysis.

The traffic analysis is performed for the purpose of finding an anomaly, to detect unsolicited traffic, or to detect malicious code content, etc (an analysis is, therefore, subject to subsequent detection). Therefore, anomaly detection is the process by which anomalies are detected against the relevant data. In contrast, an abnormality is a behavioral dysfunction, that is, a change in its behavioral characteristics that causes deviations.

Although academics are looking for other uses for the data in traffic flows, the fundamental purpose of these data is to monitor the network's performance. For instance, analyzing each device's network usage patterns to spot any unusual behavior

[1] The meaning of analysis (from the Greek *ana-lysis*, Analysis, disassembly, dissolution, unbinding) is the act of studying or examining something in detail in order to discover or understand more about it.

[2] Oujezsky, V. 2017-2022

based on network activities. However, these two approaches combine to offer the use of academic approaches to address issues applied in practice. For example, using descriptive statistics to determine network consumption by day and dividing the network into several segments based on traffic information. K-Means, two-step clustering, and other unsupervised learning techniques can be applied in this way. Future peak consumption periods can be predicted using *Generalized Linear Model* (GLM), *Classification and Regression Tree* (CART), neural networks, and other techniques applied in practice.

Thus, in practice, traffic analysis is used by security teams to find security vulnerabilities, as well as by teams that take care of day-to-day network operations. Therefore, traffic analysis is not limited to network visibility functions and is also used, for example, for forensic investment or strategic planning. Recent research from *Enterprise Strategy Group* (ESG) [4] shows, a shortened version shown in Figure 2.2, that the most important attributes of a network analytics solution are: detection of malicious endpoint behavior, the ability to monitor *Internet of Things* (IoT)[3] traffic, protocols and devices, or automated reporting capabilities of security events.

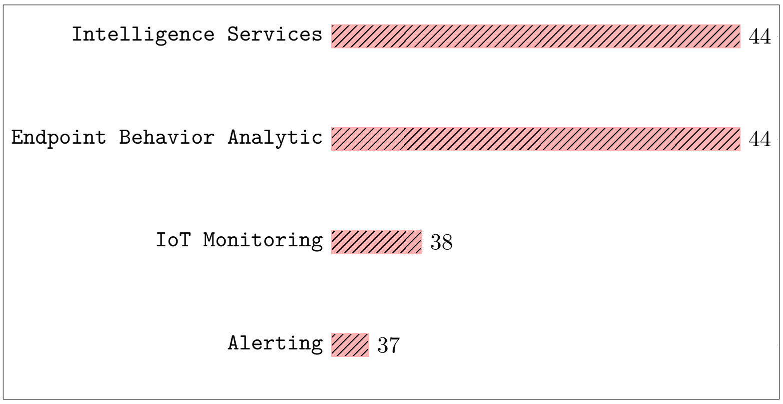


Fig. 2.2: The Most Important Attributes of a Network Traffic Analysis [4]

.

It is therefore important to continuously develop new techniques and algorithms for traffic detection and analysis to follow the latest network protocols for all of these high-demand and other segments. Moreover, this includes not only the development of algorithms for the analysis itself but also the development of techniques and data processing or techniques and algorithms for evaluating the results. First, the data must be obtained in some way in the form of some datasets or other records, such as data flow protocols, and then processed into a suitable form for further processing. Second, the data must be prepared for the analysis itself, and techniques for data preprocessing or re-sorting must be applied. The actual data analysis or extraction

[3] Connected smart “garbage” to the Internet.

follows this step. The last step has to include the evaluation of the analysis results, which acts as feedback for the analysis algorithms and processes themselves. The basic data analysis process is shown in Figure 2.3.

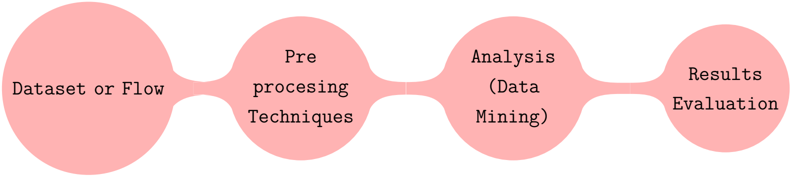


Fig. 2.3: The Generic Structure of Network Traffic Data Analysis.

Currently, there are many approaches to the analysis itself. Among the general approaches discussed above, the following types are defined, but not limited to:

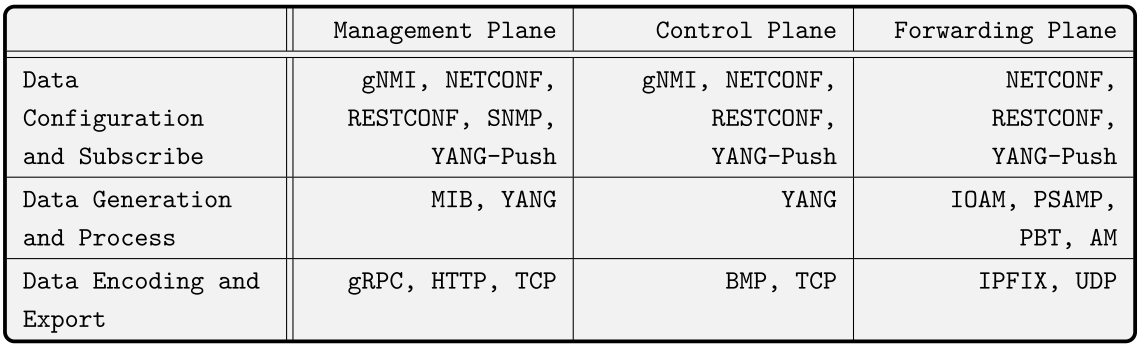
- **Host-based analysis** – refers to the collection and examination of information from a host, including but not limited to live memory collection and analysis, traffic analysis, file carving and recovery, and data analysis.
- **Network-level-based analysis** – refers to the process of examining a network's availability and activity. Tracking the data flow across various network segments includes determining what data are being sent when, when, and how. It can also be performed by:
    - **Flow-based traffic analysis** – in the sense of using any method to examine traffic flows (streams) between a source and a destination. These techniques frequently focus on looking at network flows, typically described as a series of packets sent over a certain amount of time and organized by protocol, source address, source port, destination address, and destination port.
        - **Statistical analysis** – for traffic analysis, either multivariate models or models based on available statistics such as data entropy, compression, or measurements of the mean and variance of predefined profiles are used. Cluster analysis is also often used in statistical methods. In this method, patterns of normal network traffic are also used. The basic principles of traffic measurement and analysis are simple average, data entropy measurement, and similarity comparison.
        - **Graph-theory-based analysis** – in the sense of using applied graph theory for analysis to calculate network measures.
        - **Signature-based analysis** – typically involves payload or deep packet inspection techniques and looking for specific patterns. Patterns can be extracted manually or automatically using an automatic signature extraction mechanism.

The early stages of the Internet and network development made it very simple to identify every aspect of network traffic. The key reason for this simplicity was because, at the time being, most of the traffic was not encrypted at all. Communication has started to be encrypted before being sent over the network, as privacy has grown in importance. The use of encryption is nowadays a two-edged sword. On the one hand, it helps everyday users maintain their privacy, and we cannot envision a safe Internet in the present day without using encryption technologies. On the other side, it also aids in concealing, for example, communication produced by malware. The analysis of encrypted communications is a growingly important issue. As a result, there has been a lot of research on the subject of classified encrypted traffic for a long time.

There are now three main research areas, as noted in [5], and hence three traffic inspection techniques. The first is the *Deep Packet Inspection* (DPI) approach, which separately decrypts and inspects each packet. This approach cannot be used in public networks or large networks due to privacy concerns and computational complexity. It is basically used at the network boundary using *Secure Socket Layer* (SSL) termination techniques. Behavioral analysis is the second option. This method uses several techniques. For example, it counts the number of packets transmitted or measures flow parameters such as the interval between packets, etc., for the purpose of discovering a pattern of behavior without needing to know the content of the data. Traffic behavior patterns are then either evaluated against known patterns or, conversely, deviations from known behavior are observed. Moore et al. [6] proposed a sizable collection of characteristics appropriate for behavior-based analysis. In addition to the methods mentioned, the fingerprinting technique is also an option. This technique takes advantage of data that can be seen during the early stages of an encrypted connection.

Other techniques beyond these three can include various methods to gather more information in combination to make traffic analysis effective. Such as the use of *Simple Network Management Protocol* (SNMP) and device statistics, or other new approaches such as *Data Model-Driven Management* (DMDM) using the *Yet Another Next Generation* (YANG) modeling language for the network configuration protocol, defined by *Request for Comments* (RFC) 7950 [7] . This can be part of network telemetry, which is also an option to gather useful information for network analysis. Although the SNMP protocol has historically been very effective at monitoring, it has some drawbacks. Telemetry provides an alternative way of addressing many of the shortcomings of older monitoring tools.

The topic of network telemetry is best described by the Network Telemetry Framework RFC 9232 [8]. In a generic term, in the networking world, telemetry is an automated communication process by which data and measurements are collected at remote devices and transmitted to a monitoring device. Today, many types of telemetry implementations and protocols are used, for example, the *In-band Network Telemetry* (INT) dataplane specification [9] or *In Situ Operations, Administration, and Maintenance* (IOAM) defined by RFC 9197 [10]. The purpose of RFC 9232 is to provide an entire telemetry framework, shown in Figure 2.4.



	Management Plane	Control Plane	Forwarding Plane
Data Configuration and Subscribe	gNMI, NETCONF, RESTCONF, SNMP, YANG-Push	gNMI, NETCONF, RESTCONF, YANG-Push	NETCONF, RESTCONF, YANG-Push
Data Generation and Process	MIB, YANG	YANG	IOAM, PSAMP, PBT, AM
Data Encoding and Export	gRPC, HTTP, TCP	BMP, TCP	IPFIX, UDP

Fig. 2.4: The work mapping of network telemetry, as is in RFC 9232 [8].

However, the topic of network automation and network telemetry is beyond the scope of this thesis. Basically, INT may be used for monitoring from the hardware level by data plane programming for service quality monitoring or microburst detection for latency demand application[4]. Therefore, the data analysis options can be summarized in the following techniques.

- Techniques using **traffic flow protocols**, discussed in Section 3.1.
- Techniques using **deep packet inspection** and **full packet capture** (PCAP), discussed in Sections 3.2.
- Techniques using **statistic collections** or specific data collections, discussed in Section 3.3.
- Techniques using a **mix of approaches**.

There are many approaches to obtaining traffic characteristics, in other words, a kind of basic traffic analysis. The basic characteristic is related to the performance of the network layer. Characteristics such as delay, throughput, or packet loss are measured and analyzed. One of the simplest ways is to calculate a moving average, which can be applied, for example, to flow analysis or to delay analysis, respectively. In the case of *Simple Moving Average* (SMA), it is an unweighted average of n numbers in a time series. Traffic analysis usually works with discrete data samples. When using data samples, the simple moving average can be written as Expression 2.1.

[4] GÉANT: `https://wiki.geant.org/display/NETDEV/DPP`

$$\mathrm{SMA}_k = \frac{\sum\limits_{i=n-k+1}^{n} p_i}{k} = \frac{1}{k} \sum_{i=n-k+1}^{n} p_i, \tag{2.1}$$

where k represents the window size, n the total of observed values, p_i a single observed value.

A simple moving average captures the average change in values in individual time windows or slots, but loses information about their fluctuations. There are also other variants based on the SMA. These are cumulative average, weighted moving average, exponential moving average, and other modified ones. The *Exponential Moving Average* (EMA) is used in the *Round-Trip Time* (RTT) calculation[5] of bidirectional network traffic delay by the PING (*Packet InterNet Groper*) program, which was written in 1983 by Michael John Muuss. It uses the *Internet Control Message Protocol* (ICMP) and the ICMP echo message for network measurement and analysis. In RFC 1122 [11] it is described that any network host must process an ICMP echo and respond back with an ICMP reply message. The protocol itself is defined by RFC 0792 [12]. For a series Y, the EMA value S_t at any period t can be calculated as[13]:

$$S_t = \begin{cases} Y_0 & \text{for } t = 0 \\ \alpha Y_t + (1-\alpha) \cdot S_{t-1} & \text{for } t > 0 \end{cases}, \tag{2.2}$$

where the coefficient α is the representation of the degree of weighting decrease, a constant smoothing factor between 0 and 1.

The S_t can also be written as a weighted sum of the Y_t point. The RTT was originally estimated by $RTT = \alpha \cdot$ *old* $RTT + (1-\alpha) \cdot$ *new round trip sample* in the *Transmission Control Protocol* (TCP).

The *Network Throughput* (NT) can be calculated using the TCP receive window W and the RTT related to latency, Expression 2.3.

$$TP \leqslant \frac{W}{RTT} \tag{2.3}$$

The use of only a moving average is not sufficient enough to reflect the various aspects of traffic analysis. For example, to measure the “burst” level of traffic streams. Entropy in information technology is also referred to as *Shannon Entropy* (H) after the author Claude Elwood Shannon [14]. However, the term itself was introduced

[5] RTT – the round-trip time is the amount of time it takes for a signal to transmit from one station to another, until the return of the acknowledgment of this transmission to the first station. The original algorithm has been replaced by the Jacobson/Karels algorithm, which takes the standard deviation into account.

by the physicist Rudolf Julius Emanuel Clausius in 1865. It is a calculation of the quantity of information for some whole phenomenon. If the phenomenon S under study is assumed to have n realizations with probabilities P, then the mean of the eigeninformation of all realizations of the phenomenon is defined by Equation 2.4. If this phenomenon is an occurrence of a signal element, then its unit is [Sh/element].

$$H(S) = -\sum_{i=1}^{n} P(s_i) \log_2 P(s_i) \qquad [Sh/bit]\,, \tag{2.4}$$

where S represents a system with a finite number of possible states $S \in \{s_1, s_2, \ldots, s_n\}$ and $P(s_i)$ is the probability distribution of the state S [15].

Many traffic analysis techniques are based on the concept of distance. The most well-known metric is the Euclidean distance. As such, it is widely used to measure continuity or similarity. If taking two vectors, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ as two-dimensional value observations. Taking into account an inner product space $(V, \langle ., . \rangle)$, then $d(\mathbf{x}, \mathbf{y}) := \|\mathbf{x} - \mathbf{y}\| = \sqrt{\langle \mathbf{x} - \mathbf{y}, \mathbf{x} - \mathbf{y} \rangle}$ is called the distance between $\mathbf{x}, \mathbf{y}$ for $\mathbf{x}, \mathbf{y} \in V$. If the dot product as the inner product is used, then the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$ is defined by the relation 2.5 [16].

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})} \tag{2.5}$$

Since each value of the eigenvector contributes to the calculation of the Euclidean distance, the results can vary significantly even with a small change in these values. The dominance of one set of values relative to another set of values also plays a role here. Therefore, the variability can be directly introduced into the calculation. One of the most well-known metrics with introduced variability is the Mahalanobis distance 2.6, where $\mathbf{V}$ represents the weighted variance matrix [17].

$$d^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \mathbf{V}^{-1} (\mathbf{x} - \mathbf{y}) \tag{2.6}$$

The parameter covariance matrix is a generalization of the notion of variance for random vectors. If this matrix is an identity matrix, the Mahalanobis distance is reduced to the Euclidean distance. If this matrix is diagonal, then it is a normalized Euclidean distance.

In addition to network delay, throughput, latency, or distance analysis, other techniques are being developed on top of it, some of which are presented in the following sections of this thesis.

In the field of traffic analysis, algorithms based on biologically inspired methods, commonly known as *Artificial Intelligence* (AI), are also used. These include, in particular, neural networks, computations using evolutionary *Artificial Immune System* (AIS) algorithms, and others.

For example, evolutionary algorithms have been used to improve existing traffic analysis methods. The authors in [18] used *Genetic Algorithm* (GA) for the extraction of network traffic components. The author in [19] used the Genetic Algorithm to calculate the Euclidean distance matrices. AIS was used by the author in [20] to detect anomalous traffic and by the author in [21] to analyze *Gigabit Passive Optical Network* (GPON) frames.

The new paradigms for the use of AI are not only for their use in traffic analysis, but also for the creation of a new communication model concerning the use of a semantic learning model to specify the values of the communication channel applied to the Shannon theorem [22].

Network simulation is also used for traffic analysis, followed by the application of selected methods to the resulting network model. Each model has its advantages and disadvantages. However, there is no model that is suitable for use with all types of networks [23].

It is advantageous to represent the results of the analysis in a human-friendly form, i.e., to use techniques suitable for visualizing the results and also with some form of the event highlighting. Such a representation must satisfy the condition that the results are presented in real-time. Examples of such systems are Grafana[6] or Dynatrace[7].

Network traffic analysis is also a matter of hardware, which goes hand in hand with the increasing speed of computer networks. Today we are already commonly facing a network speed of 100 Gb/s. This raises the problem of how to obtain all the necessary data without losing them, how to transfer the data for examination (e.g., via various peripherals in the computer), and how to store and read the data back. Therefore, it is necessary to be interested in this issue as well as in other topics. Commercial equipment, such as a high-speed analyzer capable of storing large amounts of data or high-speed traffic generators, can be advantageously used. In addition, specialized hardware elements such as *Field-programmable Gate Array* (FPGA) cards capable of handling large amounts of data, or devices equipped with *Graphics Processing Units* (GPU)s or *Tensor Processing Unit* (TPU)[8]s, or cloud environments, can be used in combination for the analysis itself or the development of new techniques.

[6] Grafana: `https://grafana.com/`

[7] Dynatrace platform: `https://www.dynatrace.com/`

[8] The Tensor Processing Unit is a specialized processing unit (hardware) developed by Google LLC primarily for their project as is, for example, AlphaGo by providing better machine learning results and they released it into the cloud so other people can use it in their machine learning projects.

Figure 2.5 shows the typical topology of a research polygon. The central element for the network connection is a programmable switch, which is connected to the network elements of the entire system, as well as the laboratory network and the external network. For data analysis capabilities, an optical *Test Access Point* (TAP)[9] is included in the optical section to allow data to be mirrored towards the network FPGA card interface. Another part consists of a development server in which the network FPGA card is embedded. For example, ntop software[10], in particular, n2disk and disk2n, can be used for data processing purposes. This software is provided free of charge for scientific and educational purposes. Custom algorithms can be developed on top of this software for data processing.

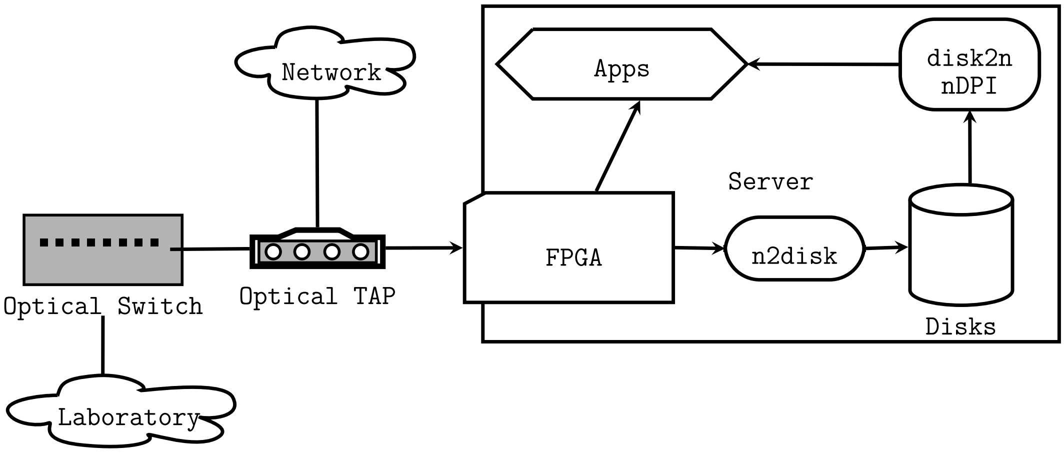


Fig. 2.5: Schematic diagram of a research polygon.

❏ *This chapter has outlined the topic and provided a simple insight, and the following chapters discuss the various aspects of the field of traffic analysis, such as the source of information and data, its processing, and evaluation.*

[9] A network tap is a device that keeps track on activity on a local network. A tap is often a specialized hardware item that offers access to the data being transmitted through a computer network.

[10] `https://www.ntop.org/`

3 Data Sources for Traffic Analysis

This chapter is composed of the author's own results and supplemented with theoretical knowledge on the subject from other sources. It deals with the problem of data acquisition for traffic analysis. The first Section 3.1 discusses network flows and their versions and outlines the topic of cloud solutions. Section 3.2 provides an overview of the techniques used to extract full traffic data from network traffic. Specific data sources and the author's published results are discussed in Section 3.3, and data sets as data sources are discussed in Section 3.4.

3.1 Network Traffic Flows

The author's contribution: Software: [11] Network Analyzer – detects and analyzes incoming NetFlow messages (versions 1, 5, and 9 in the latest version) of network devices that support them. It works in *Command Line Interface* (CLI). The output file is a database of information and analysis of the overall UNIX time duration of each reported traffic. The software has been developed to work with Python version 3 and greater, designed for the Windows operating system. Another contribution in this area is the leading of student theses, for example, on: Design and implementation of the network collector[12].

The data structures in computer networks depend on the network protocols used. The form in which they are stored and processed depends on the type of problem being addressed, and the approach (data science algorithms) used to solve the research question. These can be binary data or objects. The most commonly used data format for traffic analysis is, undoubtedly, network monitoring traffic flows. Network monitoring traffic flows are used to monitor performance, security, and other factors. They reflect the link between a source and a destination using a particular protocol. Network flows differ significantly from packet captures in that they provide only details about the transmission event, such as source and destination addresses, the network protocol used, the quantity of data transported, and other details, rather than information on the content of sent (user) data (payload).

The widely adopted NetFlow network flow monitoring standard was invented by Cisco Systems, Inc. Several multiple versions of NetFlow are now in use, most notably NetFlow 5, NetFlow 9, defined in RFC 7011 [24]. The network flow protocol in some modifications is used by a number of other vendors, such as Juniper

[11] Network Research Group – `https://nsr.utko.feec.vutbr.cz/software.php`

[12] Jaroslav Bošeľa: `http://bit.ly/3OkGc8i`

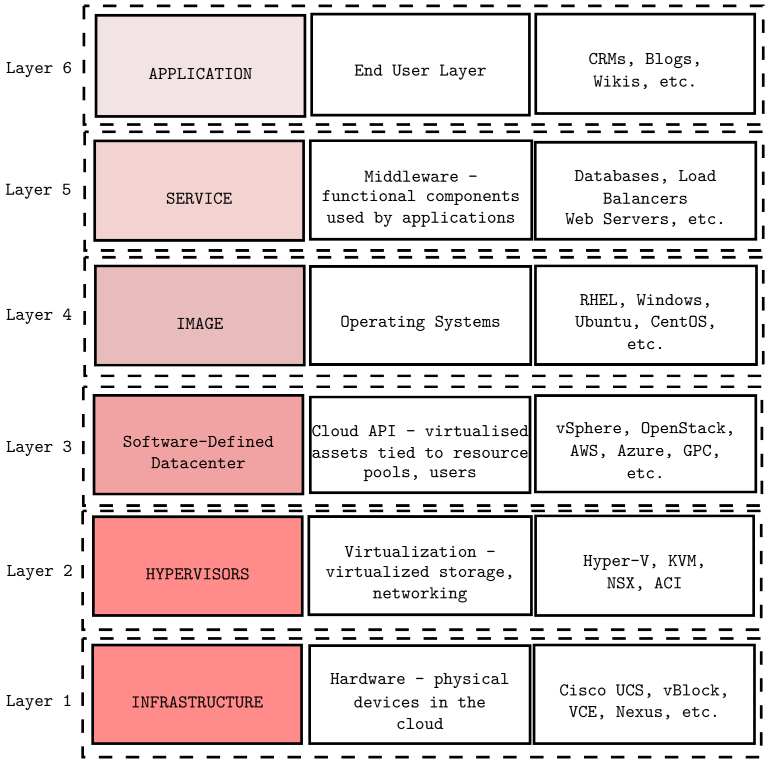
(Jflow); Hewlett-Packard, Dell, and Netgear (s-flow); Alcatel-Lucent (Cflow); Ericsson (Rflow). Another protocol is *IP Flow Information Export* (IPFIX), which is an *Internet Engineering Task Force* (IETF) protocol, defined in RFC 5101 [25], later extended in RFC 7011 [24]. *Sample Flow Protocol* (sFlow) is an alternative to commonly used flow standards. The sFlow protocol can sample non-*Internet Protocol* (IP) flows at network layer 3. It basically samples packets; it does not analyze the entire flow. It is also supported by several vendors and is defined in RFC 3176 [26]. Another protocol is the NetStream protocol[13] – it is equivalent to the NetFlow protocol but from the manufacturer Huawei. It is based on a very similar principle to the NetFlow standard. It differs, in particular, in the naming of the flow capture and analysis components. All of the aforementioned standards are unidirectional, which means that each flow record represents a connection in only one traffic direction. A bidirectional variant, commonly referred to as BinetFlow, is also available, where each flow reflects the link in both communication channels (directions). Another bidirectional flow protocol is Argus, which is an open-source project developed by Carter Bullard in the early 1980s at Georgia Tech and adopted for network anomaly detection. There are also other specific flow protocols for new types of networks, specifically for *Software Defined Network* (SDN) or cloud environment.

3.1.1 Network Traffic Flows and Cloud Environment

The previous chapter introduced network protocols for networks based on the classical hardware concept (legacy). On the contrary, cloud networks are different in the sense of virtualization, and the situation for flow analysis is not as straightforward here. Cloud systems are made up of a complex set of different technologies. A datacenter can be thought of as a multilayered system, shown in Figure 3.1, similar to the conceptual *Open Systems Interconnection* (OSI) model.

Each of these layers has specific ways to extract data for further analysis. This can be illustrated by looking at specific cloud service providers, such as AWS or AZURE. Typically, the cloud service provider has the authority and power of attorney over Layer 1 to Layer 3 devices. From the following upper Layer 3 to Layer 6 it is then a combination where funds are provided to individual lessees and customers. Customers can then build a virtualized network as an overlay layer on the physical layer. Or they can run virtualized operating systems over which they have management. But customers have no visibility into the provider-managed layer. So those are the two basic views of the data source. So in cloud services, data must be obtained at higher layers in different ways than in legacy networks. The data come from the overlay layer if it is data from the customer's perspective.

[13] Huawei – Overview of NetStream: `http://bit.ly/3EKdSZW`



Layer 6	APPLICATION	End User Layer	CRMs, Blogs, Wikis, etc.
Layer 5	SERVICE	Middleware - functional components used by applications	Databases, Load Balancers Web Servers, etc.
Layer 4	IMAGE	Operating Systems	RHEL, Windows, Ubuntu, CentOS, etc.
Layer 3	Software-Defined Datacenter	Cloud API - virtualised assets tied to resource pools, users	vSphere, OpenStack, AWS, Azure, GPC, etc.
Layer 2	HYPERVISORS	Virtualization - virtualized storage, networking	Hyper-V, KVM, NSX, ACI
Layer 1	INFRASTRUCTURE	Hardware - physical devices in the cloud	Cisco UCS, vBlock, VCE, Nexus, etc.

Fig. 3.1: A generic view on layers of cloud networks.

The situation around flows is different, and specialized tools such as CloudWatch flow log AWS Lambda[14] or AZURE FlowLogs[15] operating at network layer 4 have been developed. These techniques have some limitations in terms of updating flows by seconds or in their total amount. The other option is to use JSON blobs and visibility data (similar to SPLUNK[16]) generated by intelligent applications.

3.2 Full Data Packets

Whole data is needed when deeper analysis of traffic patterns is performed, such as DPI, data pattern matching, or signature search. The process of obtaining the data is called packet acquisition. The most well-known is the PCAP format. It is the acronym for packet capture API. The Windows *Operating System* (OS) implementation is known as WinPcap, while the Unix OS implementation is known as libpcap. On Windows OS, it is also possible to use the native Netsh trace utility. However, to export to pcap, it is necessary to use the etl2pcapng[17] conversion utility.

[14] CloudWatch: https://go.aws/3XcTsA5
[15] FlowLogs Azure: http://bit.ly/3hZS97r
[16] Splunk: https://www.splunk.com/
[17] https://github.com/microsoft/etl2pcapng

Network traffic, or packets traveling through or arriving at a specific device via a computer network, may be captured by application programs using libpcap and WinPcap. The network card has to be in promiscuous mode in order to capture packets from a network segment that are not addressed to the device. Today, there are manufacturers of FPGA network cards that support the libcap library and can be used in high-speed networks. There are wrappers for various programming languages available, as is python-libpcap[18].

The next option is to use network TAP or *Switch Port Analyzer* (SPAN). The SPAN is a specific tool included in a network switch that copies Ethernet frames passing through switch ports and sends these frames out to a specific port to the network analyzer. It should be noted that these technologies require a large storage capacity if they are to be retained for further analysis. However, the captured data must be post-processed back, for example, using libcap.

3.2.1 Full Packet Data and Cloud Network

As in a legacy network, a virtual network TAP can be used. Azure virtual network TAP allows to continuously stream virtual machine network traffic to a network packet collector or analytic tool. The other possible way is to use cloning of network traffic by using specialized network devices as is traffic manager, for example, BIG-IP traffic manager[19]. *Encapsulated Remote Switch Port Analyzer* (ERSPAN) is also another way how to collect traffic from virtualized routers to network probes in an AWS cloud-based network.

3.3 Specific Data Sources

The author's contribution: Design and implementation of a system for analysis of GPON and *10 Gigabit-capable PON* (XGPON) frames from downstream and upstream traffic. The design includes parsing and processing of the bit stream obtained from the communication between *Optical Line Terminator* (OLT) and *Optical Network Unit* (ONU) using a splitter and an FPGA network card. The result is a custom data parser software that parses the data and sends it in *JavaScript Object Notation* (JSON) format to Apache Kafka for further processing. In addition, hosting a *Réseaux IP Européens* (RIPE) probe[20] to investigate the possibilities of anomaly detection using the network probe and developing an application as part of the thesis supervision.

[18]]https://wiki.wireshark.org/Development/LibpcapFileFormat

[19] https://www.f5.com/

[20] https://www.fi.muni.cz/~oujezsky/probe.html

Another way to obtain data and information about network traffic besides using full packet capture or using network flows is to use proprietary approaches and data formats. One representative is the JSON used by the RIPE community. The Resource Request *Application Programming Interface* (API) is used to submit requests for Internet number resources to the RIPE NCC and the supported requests at the time being are *Autonomous System* (AS) Number Assignment, IPv4 First Allocation, IPv6 First Allocation, or IPv6 *Provider Independent* (PI) Assignment.

The information is retrieved from the RIPE databases and sent back to the user, who can then deserialize it and work with it as part of an analysis application to detect anomalies in networks. The traffic analysis can be trace-route based, such as route changes in neighboring connections, delay in the entire neighboring network, RTT increase, or ping based, such as anchor[21] delay per country or anchor down, etc.

3.3.1 Sensors, Wireless, and Internet of Things Networks

The author's contribution: Research in IoT security and securing end-device communications and cloud applications[27].

A particular case, not necessarily separate from classical networks, are sensors, specifically *Wireless Sensor Networks* (WSN) and IoT networks. When determining performance indicators and making subsequent decisions to enhance network performance, modeling the behavior of the networks becomes necessary. This topic is very extensive, so this thesis is limited to the overview of data acquisition. The possibilities of acquiring data directly from sensors and the principle of transmission are more related to information acquisition than data acquisition for the purpose of traffic analysis as such. In terms of analysis of the network traffic itself, parameters such as path loss, delay, throughput, or sensor network lifetime can be monitored. Basic mathematics for IoT system networks is tied to the rules of classical networks.

3.3.2 Dedicated Hardware

The author's contribution: a project focused on developing a programmable FPGA network card for gigabit-capable passive optical networks. The network card is constructed to analyze GPON frames and check the correctness of communication between an optical line terminator and an optical network unit directly in the optical domain. The GPON networks use a specific encapsulation method for Ethernet

[21] `https://atlas.ripe.net/about/anchors/`

frames and control messages. Because the network card operates directly in the optical domain, it is possible to provide real-time analysis of headers of GPON transmission convergence layer [28].

As mentioned in the introductory chapter, analytical tools can be supplemented with specific programmable hardware that then provides network data that meets the exact requirements of the research and analysis. Figure 3.2 shows the second generation of FPGA development card that provides direct access to data from the GPON network that was developed as part of a specific research project [3][28][29].

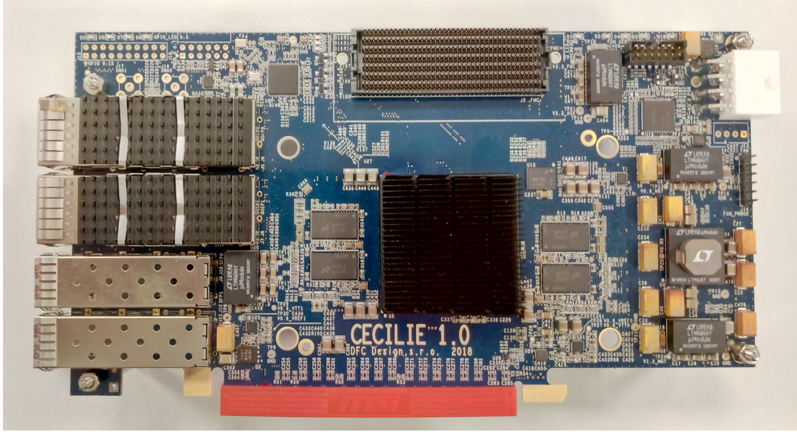


Fig. 3.2: FPGA Network Card [29].

Different technologies and different protocols require additional data processing. Most of the available solutions are for Ethernet-based networks. Available tools that enable real-time analysis of PON networks, *International Telecommunication Union – Telecommunication* (ITU-T) G.984 [30] or ITU-T G.987 [31] are limited. Such a network card can be connected to the laboratory PON network, as is shown in Figure 3.3.

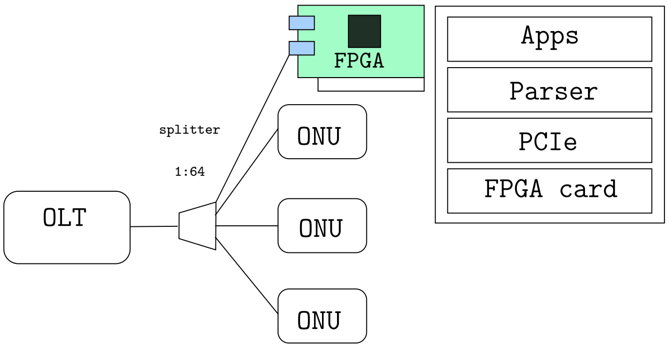


Fig. 3.3: The basic components of the PON analysis system.

This card enables the forwarding of traffic (PON frames) from the splitter to the development server via *Peripheral Component Interconnect Express* (PCIe). The development server then contains a software frame parser and applications (modules) for traffic analysis. To obtain outputs from this system in terms of detected security problems in the optical network, reported incidents in Apache Kafka can be stored using a specific message format. Then the events can be processed using custom-defined templates for optical networks.

3.4 Datasets

The author's contribution: A specific map dataset created in JSON format for geolocating IP addresses [32] [33]. Definition of GPON and XGPON frames in JSON format [3] as a source for frame and traffic analysis.

Datasets are used, among others, in the development of new methods and algorithms. A dataset, in terms of network traffic dataset, is a set of measurements that have been obtained using data acquisition tools. Therefore, they can contain the data in flow or binary form. Nowadays, many datasets that have already been created can be searched, for example, using Google Dataset Search[22].

The simplest dataset format is the *Comma-Separated Values* (CSV) for tabular data. The CSV is basically used for "flat" data. For data creating a "tree", which has multiple layers, is the JSON file format used as the most common file format. The following Figure 3.4 shows a possible JSON representation describing a map in GeoJSON format defined by RFC 7946 [34].

In Figure 3.5, another format of JSON is presented. In this format, the information about the XGPON frame is sent to Apache Kafka for further processing by

```
"type": "FeatureCollection",
"features": [{
    "type": "Feature",
    "properties": {},
    "geometry": {
        "type": "Polygon",
        "coordinates": [[[14.809375, 50.858984
            375],
        [14.895800781250015, 50.86137695312499
            6],
        . . .
```

Fig. 3.4: Polygon JSON format of GeoJSON map.

[22] https://datasetsearch.research.google.com/

```
{
   "Psync": 14259830462546033000,
   "SFC": {
      "SuperframeCounter": 2245625728463605,
      "HEC": 7108
   },
   "PON": {
      "ID": 1268321022378337,
      "HEC": 1051
   },
   "HLend": {
      "BWMapLength": 3,
      "PLOAMcount": 1,
      "HEC": 5364
   },
   "PLOAM": [
      {
         "ONUid": 1023,
         "MessageID": 1,
         "SequenceNumber": 216,
         "MIC": 15128669414371017000
      }
   ]
}
```

Fig. 3.5: JSON format of XPON frame.

the analysis tools [3]. The format defined by JSON can then be deserialized and processed into individual classes in the programming code.

Another representation of data is the lightweight SQLite format to deliver database files, which is the most used relational database today. Multiple tables make up SQLite databases, and each table holds data in a tabular style. These tables are comparable in practice to CSV files, except they accommodate huge datasets better. In practice, there are several types of databases ranging from hierarchical databases, which are not much used in practice, to document, relational or non-relational databases. Google also created the “big data” *Structured Query Language* (SQL) storage known as BigQuery. The Google BigQuery Public Datasets program makes a number of sizable public datasets, including all the code on GitHub, publicly accessible. They are multi-terabyte datasets housed on Google’s servers. Instead of loading the files from disk, the interaction with the dataset is done by creating SQL fetch queries within the Google BigQuery Python library.

4 Data Preparation and Process Streamlining Techniques

This chapter focuses on data preparation related to increasing the efficiency of data processing in traffic analysis techniques. It is composed of the author's own results and supplemented with theoretical knowledge on the subject from other sources. The first Section 4.2 provides a brief theory of data mining and clustering topic. Section 4.2.2 presents the authors' approach for streamlining analysis processing by using genetic algorithms.

4.1 Data Preparation

In the following section, clustering algorithms will be discussed. Therefore, it is necessary to mention at least a few words on the topic of data preparation. In basic, clustering determines how similar two examples are in the way of merging all of the feature data from two instances into a numerical value, so the size of data must match for the feature data when they are combined. The normalization, transformation, and creation of quantiles techniques are used besides of sampling, scaling, or randomization.

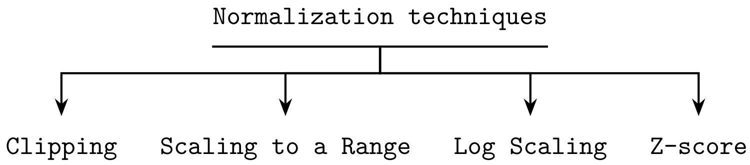


Fig. 4.1: The normalization techniques.

Normalization is used when the data are numeric. The basic normalization techniques are shown in Figure 4.1. When the data are categorical (when features have a specific set of possible values), hashing techniques may be used to process the different length of features. In machine learning, known as feature hashing techniques or hashing trick. This hashing technique is used very frequently in network analysis or network pattern analysis. The idea is straightforward, to convert data into a vector of features. The authors in [35] presented a mathematical representation of the basic feature hashing algorithm[23].

[23] Feature hashing coding example: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.FeatureHasher.html

4.2 Data Classification and Clustering Techniques

The author's contribution: The aforementioned subject of clustering techniques was used in the author's several projects for the preparation and classification of network traffic. Parallelization of genetic algorithms was proposed [36] [37] and used to increase the data processing efficiency of machine pre-processing. The following text and results are provided from the author's research papers.

Clustering is one type of unsupervised learning method used in machine learning. The objective of clustering is to group comparable groupings of the incoming data. Clustering techniques are used to classify the data obtained from multidimensional observations. The data are sorted so that the difference in the data values of the group members is close to zero. Cluster analysis is concerned with the formation of just such homogeneous units. The number of data dimensions is reduced, and one variable expresses the membership of a data unit in the cluster.

The basic clustering problem can be described in "*general terms*" by providing a data matrix $\mathbf{X}_{(m,n)}$, where m is the number of objects and n is the number of variables. The number of clusters is denoted by k. It is a decomposition of the set of m objects in dependence on the values of n into k clusters. Only decompositions with disjunctive clusters are considered. An object must belong to only one cluster C_k. The distance for all objects is calculated. This calculation yields a square-symmetric matrix, called the association matrix. Clustering itself may be referred to as hard and soft clustering. In hard clustering, each object is allocated to one cluster C_k as is valid for the general description. In soft clustering, a probability is assigned to each object and cluster that the provided object belongs to the cluster. Therefore, a point has the potential to belong to numerous groups [38].

Clustering methods may also be classified depending on whether a fixed number of clusters is defined or not as hierarchical or non-hierarchical. Several approaches to clustering exist and an exhaustive list is provided by authors in [39] in their article "*A Comprehensive Survey of Clustering Algorithms*". A basic overview of the clustering methods based on the approaches is shown in Figure 4.2.

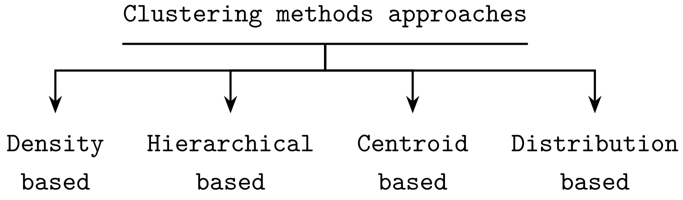


Fig. 4.2: Clustering algorithms according to their approach [39].

Non-hierarchical, centroid (partition) based clustering includes the K-Means method, the X-Means method, and the K-Medoids method. Next, the principle of the K-Means method is presented. This method is among the methods mentioned above as their basis.

The basic K-Means algorithms [40] randomly partitions data into k clusters. It is determined by the k centroids[24] c_k using the concept of the average distance in the cluster. Each cluster object and its distance to the centroid is evaluated using a distance metric. If it is closer to another one, it is relocated, and the centroids are recalculated so that a new average is computed over all the elements in the cluster. The step is repeated until none of the elements cannot be relocated anymore. Mathematically, the relation k of the C_k clusters and the k centroids can be expressed by minimizing C_k and c_k according to the following relation 4.1 can be expressed [40].

$$\sum_{k=1}^{k} \sum_{x_n \in C_k} \|x_n - c_k\|^2 \tag{4.1}$$

The minimization problem is a hard problem to solve. The best-known solution uses Lloyd's algorithm. Once the centroids are known, the elements are assigned according to the concept of distance according to the following relation 4.2.

$$\begin{aligned} C_k = \{x_n : \|x_n - c_k\| \leq \forall \|x_n - c_k\|\}\,, \\ c_k = \frac{1}{C_k} \sum_{x_n \in S_k} x_n \end{aligned} \tag{4.2}$$

There are several modifications to this algorithm. Disadvantages include the fixed definition of k-clusters and the use of the Euclidean distance calculation, which is prone to distant objects. The use of Euclidean distance has drawbacks. Particularly when using high-dimensional data, the phenomenon known as the "curse of dimensionality" is affecting the result. Data get scarce as dimensionality rises, which is a concern. The parameters issue does not just apply to the issue that is discussed; it also arises with every data mining work, and the parameters have a distinct impact on every method that is utilized. Again, methods exist to validate the number of k-clusters, such as the silhouette validation method or the *Davies-Bouldin validation index* (DB). The DB is based on the proportion of the sum of the intra-cluster distribution and the inter-cluster distribution. This index is obtained from Equation 4.3 [41].

$$DB = \frac{1}{n} \sum_{i=1}^{n} \max_{i \neq j} \left\{ \frac{C_n(Q_i) + C_n(Q_j)}{C_n(Q_i, Q_j)} \right\}\,, \tag{4.3}$$

[24] A centroid is the center of the cluster. It is a vector containing the averages of the variables observed in the cluster.

where n is the number of clusters, $C_n(Q_i)$ is the average distance within a cluster from its center, and $C_n(Q_i, Q_j)$ is the distance between clusters represented by centroids.

Other clustering methods that use different distance metrics can also be used in the clustering. It depends on the format of the data, as stated before. The distance between two samples is measured and quantified in some way in all clustering techniques. If the distance is a squared Euclidean distance, the K-Means clustering method can be used. If the distance is the Taxicab metric (Manhattan), where the distance of two points is defined as the absolute difference of their Cartesian coordinates, the K-Medians method is suitable for the computation.

In network analysis, it is often necessary to compare the similarity of data strings from pre-processing. The data string can be, for example, a message hash generated or a fingerprint [42]. For this, the theory of metric space and metrics is applied. A mapping of $\mathcal{M}^2 \rightarrow \mathbb{R}$, where $\mathcal{M}$ is any non-empty set, is called a metric ρ. For each $x, y, z \in \mathcal{M}$, the metric must meet the basic three axioms, as is identity $\rho(x, y) = 0 \iff x = y$, symmetry $\rho(x, y) = \rho(y, x)$ and triangle inequality $\rho(x, z) \geq \rho(x, y) + \rho(y, z)$.

A standardized real and discrete metrics for different sets $\mathcal{M}$ may be used. From the real metrics, it can be Euclidean and Manhattan. From the discrete metric, Levenshtein [43] (edit distance is another name for this metric) and Damerau-Levenshtein metric used in combination with K-Median or *Ordering Points to Identify the Clustering Structure* (OPTICS) come into consideration, or similarity metric suitable in combination with CD-HIT clustering method. Good results were obtained by authors [42][44] with the use of the Levenshtein metric.

When using the Levenshtein metric, the least number of operations required to convert one string to another serves as “deciding factor”. The following character-related actions are acceptable: removing, adding, and swapping out characters. The formula for *Levenshtein distance* (*lev*) between two strings of lengths $|a|$ and $|b|$ is shown in Equation 4.4 [43].

$$lev(a, b) = \begin{cases} |a| & \text{if } |b| = 0, \\ |b| & \text{if } |a| = 0, \\ lev(tail(a), tail(b)) & \text{if } a[0] = b[0], \\ 1 + min \begin{cases} lev(tail(a), B) \\ lev(a, tail(b)) \\ lev(tail(a), tail(b)) \end{cases} & \text{otherwise.} \end{cases} \tag{4.4}$$

where tail is a function that returns the string minus the initial letter, and $a[0]$ and $b[0]$ represent the first character of a and b, respectively.

Levenshtein metric has been updated by the Damerau-Levenshtein metric. The same actions are permitted, plus the option to swap two consecutive characters is added. This statistic may be normalized in relation to the string's length. The outcome is scaled to the range $[0, 1]$. The distance between an i-symbol prefix of string a and a j-symbol prefix of string b is known as the Damerau-Levenshtein distance between the two strings a and b. The definition is as follows, shown in Equation 4.5 [45].

$$d_{a,b}(i,j) = min \begin{cases} 0 & \text{if } i = j = 0, \\ d_{a,b}(i, j-1) + 1 & \text{if } i > j, \\ d_{a,b}(i-1, j-1) + 1 & \text{if } i, j \neq 0, \\ d_{a,b}(i-2, j-2) + 1 & \text{if } i, j > 1 \text{ and } a_i = \\ & b_{j-1} \text{ and } a_{i-1} = b_j \end{cases} \tag{4.5}$$

The use case for the presented algorithms can be finding non-usual traffic patterns. OPTICS, which is a generalization of *Density-based Spatial Clustering of Applications with Noise* (DBSCAN) in combination with the Damerau-Levenshtein metric, can be used to cluster similar and non-similar network traffic patterns before being converted to fingerprints to find any anomalies. The samples are ordered in such a way that the two nearest samples always follow each other in sequence. The resultant clusters are identified by dividing the sequence into locations where the relative distance exceeds a predetermined threshold. Separated outliers can be identified as unusual traffic detected.

To prove and evaluate the results, one of the basic methods is used to evaluate and compare univariate clustering algorithms, that is, to compare the change in cluster size for different clustering algorithms. That is, what clusters are produced by each clustering method and how they differ from each other. For this purpose, the normalized Shannon entropy H [14] can be used. The normalized Shannon entropy is defined as follows in Equation 4.6, given the proportions $p_1, \ldots, p_N$ of cells assigned to each of the N clusters [14].

$$\frac{H}{H_{max}} = -\sum_{1}^{N} p_i \frac{log_2 p_i}{log_2 N} \tag{4.6}$$

Since the actual degree of equality of cluster sizes may vary between data sets, it is helpful to subtract the normalized entropy calculated from the actual distribution to obtain a final performance index. To evaluate how well inferred clusters recover the true of sub-populations and to evaluate the stability of clusters, *Hubert-Arabie Adjusted Rand Index* (ARI) [46] is one of the useful methods. To evaluate whether points are clustered well and separated well, *Silhouette Coefficient* (SC) metric can be used [47]. The separation of the clusters based on the distances between and within the clusters is measured to obtain the silhouette score. For each sample, the mean intra-cluster distance A and the mean distance of the nearest cluster B are calculated. Then the silhouette coefficient for the sample is $(B-A)/\max(A, B)$ with a result between 0 and 1 looking for a higher value. When working mainly with clustering, visual examination of clusters is one of the good practices to evaluate them, mainly when they are unsupervised. To better visualize clusters, two very popular methods are used for dimensionality reduction, such as *Principal Component Analysis* (PCA) or *t-distributed Stochastic Neighbor Embedding* (t-SNE). A significantly understandable visual representation is provided by the t-SNE, shown in Figure 4.3.

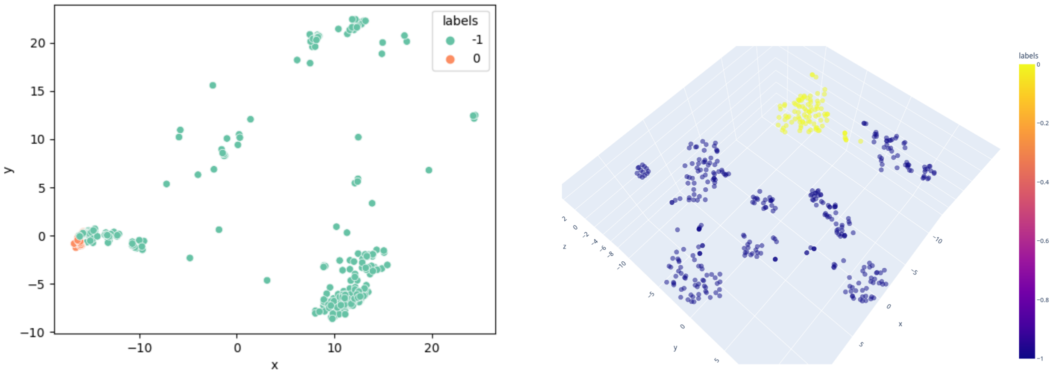


Fig. 4.3: The PCA and t-SNE representation of Optics used in [42]

.

Potly[25] was used to visualize botnet traffic and normal traffic with the t-SNE[26]. A precomputed similarity matrix generated by the Damerau-Levenshtein distance and computed OPTICS labels has been used in both cases. Only the vector of labels and the two first matrix values are selected in this visualization. However, it still gives a good view of the cluster layout, more readable than the basic reachability graph.

❏ *A part of the author's research was presented here. A combination of a clustering method and metric is used to analyze and cluster traffic patterns. There is no simple answer to which clustering algorithm to use. It always depends on the type of data. Data sets can have many entries, but not all clustering algorithms scale well. For example, if computing the similarity between all pairs, the runtime increases as the square of the number of an entry n, thus it has a complexity of $\mathcal{O}(n^2)$, so such an algorithm is not practical when there are entries in millions.*

[25] Plotly: `https://plotly.com/`

[26] `https://www.fi.muni.cz/~oujezsky/t-SNE_example.html`

4.2.1 The Role of Evolutionary Algorithms

The motivation to use the principles of evolutionary algorithms in the problem of data clustering is mainly their application to practical problems that cannot be solved with other methods. Evolutionary algorithms represent suitable techniques for solving complex optimization problems and can achieve better results than linear methods. The most commonly used evolutionary algorithms are genetic algorithms and combined evolutionary strategies. Evolutionary algorithms are used for both single-criteria optimization and multi-criteria optimization, also called multi-objective or multi-objective optimization [48].

4.2.2 Process Streamlining by Using Genetic Algorithms

Modern genetic algorithms are evolutionary algorithms that are derived from natural laws and phenomena. By their very nature, genetic algorithms lend themselves to parallel processing, which boosts performance and promotes optimization. Compared to a serial architecture, parallelization increases the speed of the algorithm and the load distribution.

Algorithm parallelization is a valuable technique for making an algorithm be more efficient and faster. GAs are very suitable for a large set of problems, but some of them require a more significant amount of time, and therefore, GAs became unusable for them. The most time-consuming operation within GAs is the fitness function[27] evaluation. This function is performed for each individual (solution) and is independent of the others. This makes it suitable for parallel processing. The basic idea of parallelization is shown in Figure 4.4. Genetic operators: mutation and

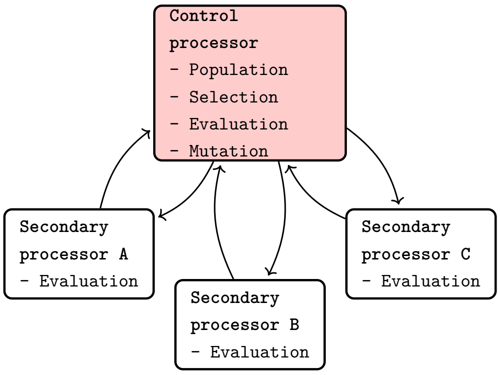


Fig. 4.4: The parallel processing of GA.

[27] In general, evolutionary techniques use an objective function f, also called fitness function, to evaluate the best solution (individual)

crossover can work in isolation as they act on one or two individuals. These operators are usually much more straightforward than fitness functions but can consume more time than calculating a fitness function depending on the crossover/mutation operation. Communication is also a problem for another genetic operator, selection, which often needs information about the entire population. Therefore, the following is concerned only with parallelizing the fitness function [37].

For parallel processing, there are several basic models. They are the Hierarchical Model, the Multi-Population Coarse-Grained Model, the Global One-Population Primary-Secondary (Master-Slave[28]) Model, and the architecture One-Population Fine-Grained Model. The Primary-Secondary model works with only one population. Further processing, if the data of the last not yet evaluated individual in the population has already been sent, divides the Primary-Secondary GA into two types:

- **Synchronous** – in which the primary node will start sending data from the new population only after the previous one has been processed on all secondary nodes.
- **Asynchronous** – when the data of all individuals in the population have already been sent to the secondary nodes, the primary node starts sending the data from the new population to the free nodes.

While functioning similarly to standard GA, synchronous architecture Primary-Secondary GA is faster. When converting conventional GAs to asynchronous ones, asynchronous systems operate differently. The number of nodes may stay the same or fluctuate during the course of the GA computation. Proper load-balancing across *Central Processing Unit* (CPU)s in a multi-user environment would boost system and GA effectiveness. When we have more processors and a population that is the same size, this method of GA is more effective [49].

In the case of fewer processors, the algorithm loses its efficiency, and this is due to the existence of the primary node and the related communication between it and the secondary nodes. However, the deployment of a load-balancing system can restore the algorithm to efficiency. The most significant advantage of this GA is that it does not change the functionality of the regular GA, and it is therefore effortless for users to modify their regular GA and apply the Primary-Secondary architecture to it.

The Fine-Grained model works with one global population spatially dispersed into nodes, thus creating a topology with neighborhoods. The Fine-Grained model is a stochastic process based on Markov chain[29] model [50], ergodic, converting to

[28] Master-Slave indication is replaced by Primary-Secondary.

[29] Definition of Markov Chain: Let $X = \{X_i : i \in S\}$ be a discrete time stochastic process with finite state space S. If $Pr\{X_{k+1} = j|X_0 = i_0 \dots X_k = i\} = Pr\{X_{k+1} = j|X_k = i\}$ then X is a Markov chain [15] [50].

stationary distribution. The closest environment of the node gives the neighborhood, and the neighborhoods may overlap. On the other hand, due to the relative isolation of a neighborhood, the best individuals do not spread as fast as in other types of GA, which increases population diversity. Neighborhoods cover the entire node topology and can have different shapes. Different node topologies and their neighborhoods imply different GA behaviors [36].

Neighborhoods enclose the space where selection can take place. The selection is thus local and parallel (by neighborhoods) compared to other types of GA. Each individual participates in the selection process only within the neighborhoods of which it is a part. Only one central element is modified by crossover and mutation within one neighborhood.

Frequently used topologies are one and two-dimensional grids, which are also often used to deploy computational elements of parallel computers, which is also why they use GA [51]. However, the use of the bus topology is excluded [52]. When designing a topology, borders are often a problem. Since we want all nodes to be linked to the same number of nodes, we need to ensure that the neighboring nodes are connected. The curvature of space most often solves this. The line becomes a circle, and the two-dimensional grid becomes a toroid.

The type of topology also affects the neighborhood schedule. All GA operators are performed in parallel, but differently from ordinary GA. The selection is applied only locally. On the other hand, other types of GA utilize global, centralized, and sequential selection, which requires collecting large amounts of data, leading to communication problems known as the “bottleneck”. The mutation is independent of other individuals, so it can be performed on each node separately without communication between them. As an operator over two individuals, the crossover will already require communication, the rate of which will depend on the population density and the selection algorithm.

However, different behavior of genetic operators affects the algorithm’s behavior. Therefore, this type of GA may not work on a particular problem in the same way as the ordinary GA. The advantage of efficiency outweighs this disadvantage, flexible and scalable implementation on hardware [52]. The nodes of the system are simple and uniform, and their communication is local and at regular intervals. The number of individuals per node must be considered when designing the system. A higher number of individuals would speed up the algorithm, but, on the other hand, it would increase the system’s complexity. When adding individuals to nodes, increased requirements for the memory, processor performance, and communication infrastructure need to be considered.

The Coarse-Grained model is described in [49]. The main difference between Fine-Grained and Coarse-Grained genetic algorithms is that the Fine-Grained genetic algorithm works with one global population that is spatially dispersed into a large number of nodes, so it is fine-grained. All nodes are identical and contain only one or two individuals. The number of nodes is much larger than for other parallel genetic algorithms. Coarse-Grained genetic algorithms are often referred to as "distributed" and work on multiple populations or "deme". The evolution process takes place over each "deme" asynchronously and relatively in isolation. This relative isolation, which may be partially disrupted by migration, is characteristic of the Coarse-Grained genetic algorithm. Determining whether it is a Fine-Grained or Coarse-Grained model is possible by comparing the number of nodes and the number of individuals in one of them. The Coarse-Grained model is when the number of nodes is less than the number of individuals in one, and vice versa.

Comparing their space search and speed is a difficult task. A summary of various studies in [51] gives different comparisons of these models while noting that the comparison cannot be absolute but must be done concerning the particular optimization problem since some studies indicate that the Fine-Grained model is better, while some indicate the opposite. However, the presented theoretical study concludes that with a sufficient number of processors, the Fine-Grained model is faster (regardless of the population size). However, communication and memory requirements were not taken into account.

Each model offers different implementation options. Topology, population size (as well as the neighborhood in Fine-Grained models), implementation of genetic operators, and migration may vary. Usually, synchronous migration is applied that occurs at the same predetermined intervals. Asynchronous migration is triggered by some event. An example is an implementation where the algorithm stops migration if it is already close to convergence. Another possible example may be the opposite, where migration begins only when the population is wholly converged (thus restoring population diversity). This raises the critical question of when is it right to migrate, how many individuals should be migrated, and what happens to individuals who will suddenly become redundant in the new population (if a constant number of individuals in the population is used).

However, in general, too high or too low migration frequency can have a negative impact on the quality of the solution. However, it is clear from the overviews of the various models that models with migration generally perform better than those without it. Different topologies are optimal for different optimization problems. In the Coarse-Grained model, some studies even suggest that the shape of the topology is not very important if the topology is densely interconnected and not too large, which would cause insufficient "mixing" among individuals. However, in general, smaller topologies converge faster, but at the risk of lower-quality solutions. Therefore, finding the right size is an essential step in implementation.

The parallelization of evolutionary processes is more related to the real processes of evolution that can be observed in nature. The maximum utilization of parallelization is conditioned by its implementation on parallelized hardware. Although implementing the Fine-Grained or Coarse-Grained models on sequential computing units does not bring much acceleration, it is recommended to use them at the expense instead of ordinary GA.

The factors SpeedUp, Efficiency, and Scaling are frequently used to evaluate the advantages of parallelization [49]. The SpeedUp parameter S is given by Equation 4.7.

$$S = \frac{T_S}{T_P}, \tag{4.7}$$

where T_S is the computation time for the serial algorithm and the Tp is the computation time for the parallel algorithm. The efficiency parameter E is given by Equation 4.8.

$$E = \frac{S}{p}, \tag{4.8}$$

where p is the number of processing units and corresponds to the number of processes. The scaling parameter detects the loss of algorithm performance as the number of processing units and the difficulty of the calculation increase.

The contribution of parallelization can generally also be evaluated by comparing the time taken by each compared model to find the best solution. The efficiency of the algorithm itself can be evaluated by comparing the number of iterations.

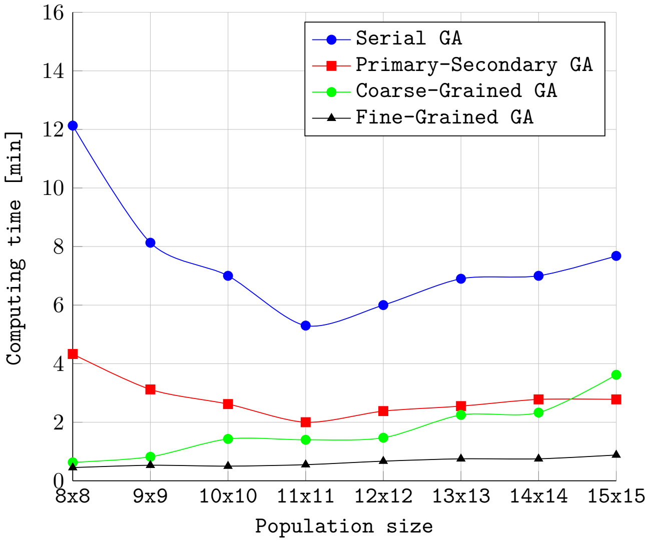


Fig. 4.5: The comparison of computation time of GA models.

As an example, in the author's research [37], test results of the measurements of the time required to find the correct solution have been provided and their illustrations are shown in Figure 4.5.

As with the progression of the number of iterations, the similarity between the progressions of the Serial and Primary-Secondary models can be noticed, as well as the similarity between the Fine-Grained and Coarse-Grained models. The wave forms of the Serial and Primary-Secondary models are similar to those for the number of iterations. However, the trend is reduced for the population dimension 11×11 by decreasing time changes. Additionally, computational time starts to increase with increasing population size.

In the case of the Fine-Grained and Coarse-Grained models, the time increases with increasing population size from the beginning of the run. This phenomenon can be explained by the fact that the algorithm has to process a larger and larger population, and the increase in this time exceeds the time saved by fewer iterations. This is true for all GA models, including the Serial one. It is assumed that the Fine-Grained model is faster for a sufficient number of processors and independent of population size.

❏ *A part of the author's research was presented here. Parallelization applied to data clustering processes using genetic algorithms is practically applicable in network data processing and leads to an increase in the computational speed of algorithms.*

5 Selected Traffic Analysis Techniques

The previous chapters covered data processing. This chapter presents selected techniques that combine the various approaches and techniques in practice. It is divided into three main topics. Section 5.1 discuss the network localization techniques and provides an overview of the technique proposed by the author, Section 5.2 provides an overview of the author's contribution to this topic which is followed by Caption 6, related to the network data analysis and reporting.

The author's contribution: The sub-objective of the research was directed toward the early detection of attacks and analysis of traffic behavior. In particular, the research started by focusing first on the possibilities of improving the localization analysis of stations or computer nodes in the network based on the knowledge of the general boundary of their occurrence and on the possibilities of botnet traffic detection and analysis. Second, the research continued by inventing a method and algorithms combination for traffic similarity observation using a genetic algorithm and clustering. The articles related to this research were published in [32, 33, 53, 54, 55] and the following statements are from the results of the research carried out. The sub-part of the research is focused on the potential use of artificial intelligence as anomaly detection tools and integration into tools used for event logging. Ongoing current research focuses on the use and implementation of algorithms to enhance security in smart grids and the specific use of federated learning.

5.1 Location Accuracy Analysis

Obtaining findings about the behavior and location of network nodes in real-time is crucial for modern network security solutions. For example for *Web Application Firewall* (WAF) services and *IP Intelligence* (IPI) when using DPI.

The well-known regional Internet registries, which provide IP addresses to businesses in their respective service regions and are the primary sources of data on IP addresses, are the following:

- African Network Information Centre (AfriNIC).
- American Registry for Internet Numbers (ARIN).
- Asia-Pacific Network Information Centre (APNIC).
- Latin American and Caribbean Internet Address Registry (LACNIC).
- RIPE Network Coordination Centre (RIPE NCC)[30].

[30] RIPE NCC Database: `https://www.ripe.net/manage-ips-and-asns/db/tools/geolocation-in-the-ripe-database`

There are many data mining techniques in this field, using registry entries or specialized databases that maintain information related to IP latitude and longitude positions. The problem is with the accuracy techniques, which are not many. Accuracy is improved mainly by statistical analysis or data scrubbing[31]. The problem in the following is to verify the correctness of an IP address location, shown in Figure 5.1.

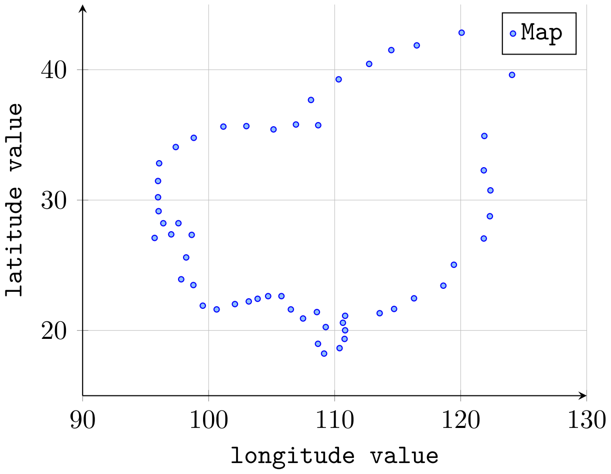


Fig. 5.1: The coordinate map example.

Suppose the address is claimed to be in an area (an area defined by the points formed by latitude and longitude values). In that case, it also needs to be verified against latitude and longitude values in contrast to the database entries. It is not uncommon for an IP address nominally located in one area to be shown using location values in another area. Clustering can be performed to eliminate this error.

The computation of positioning requires fast and efficient processing. Clustering techniques, such DBSCAN or K-Means, are also used to achieve this, but they are not always appropriate or accurate. Back to the clustering topic, instead of using the Euclidean distance as a distance function for location clustering (to calculate distance between two latitude-longitude points), the Haversine formula[32] can be used. Again, it has advantages and downsides. It may be applied when there are discrete, non-overlapping clusters, but it is unsuitable for K-Means. The coordinates can move around in an arithmetic average. If using the DBSCAN, there is an issue with the selection of the radius variable and the minimal number of points needed to construct a dense region, which might result in incorrect point assignment.

[31] Techopedia definition of data scrubbing – "*Data scrubbing refers to the procedure of modifying or removing incomplete, incorrect, inaccurately formatted, or repeated data in a database. The key objective of data scrubbing is to make the data more accurate and consistent.* – the end of the citation".

[32] https://mathworld.wolfram.com/Haversine.html

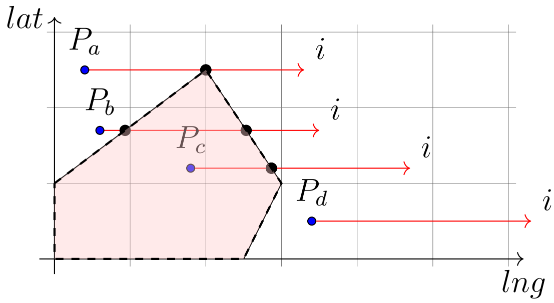


Fig. 5.2: The principle of polygon intersection.

Continuing research compared different algorithms and a new approach using an intersection method, the inclusion of a point in a polygon has been proposed by the author. Such an algorithm based on the crossing number method proved to be accurate also for points lying close on a cluster boundary, but with a specific issue with very close lying points. To avoid the so-called "degradation point", starting point verification method should be used in combination with the algorithm.

Determining the inclusion of a point P in a $2D$ planar polygon is a geometric problem. In the research, the author used the crossing number method. Within this method, the number of times a ray starting from the point P crosses the polygon boundary edges is counted. The point is outside when the number is even; otherwise, when it is odd, the point is inside [56]. The algorithm is composed of the following steps, shown in Fig. 5.2:

- In the first step, it creates a horizontal line on the right side of every point P and extends it to a defined value expressing infinity i.
- In the second step, it counts the number of times the line intersects with polygon edges.
- The conditions are determined in the following steps; a point P is inside a polygon if either count of intersections is odd or the point P lies on the edge of the polygon. If none of the conditions is true, then the point P lies outside.

The algorithm returns the Boolean value *true* if a point P lies on the border, or if the point P has the same value as one of the vertices of the given polygon. To do so, after the algorithm checks if the line from P to the extreme intersects, it continues to check whether the P is colinear and if the point P lies on the current side of the polygon. If it does not lie, it returns the value *false*, else *true*. The algorithm based on the crossing number method proved to also be accurate for points lying close to the map boundary. To avoid the degradation point, a verification of the starting point needs to be taken into account. This verification occurs because the boundary is also considered part of the map. The complexity of it in the worst case includes the possibility that a point P may intersect all N edges of the polygon, and $\Omega(N)$ time is necessary in the worst case [57]. Using this algorithm and approach, it is possible to unambiguously confirm or deny the affiliation of an IP address with the map base.

5.2 Network Traffic Behavior Analysis

The second aspect of network traffic behavioral analysis is to get an understanding of the specific behavior of individual network nodes, not just their specific relative or real location. There have been several methods employed up until now for identifying malicious traffic (some kind of anomaly behavior).

Host-based detection mixed with the Network-level based was chosen in paper [58], where Behavioral Classification is performed. A comprehensive view on the issue of botnet networks has been presented in [59]. Graph theory and cyber-thread infrastructure is covered by an article [60]. The Network-level-based approach has been presented in [61], where the authors used flow data collected from a backbone network to detect e-mail spammers. Detailed research on this topic is presented by Stebastián García et al. Their work is concerned in the time-based behavioral characteristics. In article [62] the identification of the *User Datagram Protocol* (UDP), TCP and *Hypertext Transfer Protocol* (HTTP) *Command and Control* (C&C) channels and its analysis is presented.

From this point of view, Host-based detection, Network-level-based detection, and Graph-theory-based detection category may be used to group existing approaches. The author of this thesis and his research has been generally concerned with Network-level and Graph-theory combined-based detection. The behavioral representation of a specific network traffic connection takes into account aspects such as the lifetime of the traffic connection, the size of the flow, and the duration of the traffic flow. The data source can be a network flow protocol or, for example, a pcap file.

The related author's research and technique presented here focused on the possibility of comparing data flows and detecting, for example, the source of ransomware [33] propagation from several sources in different time sequences without using DPI, but only NetFlow protocol. The proposed technique is based on the hypothesis that each specific traffic has some unmistakable property such as periodicity in time, shape, or error (traffic similarity observation) and, recently, a statistical

[33] Ransomware is a type of malware that threatens to expose a victim's personal data or permanently block access to it unless a *ransom* is paid.

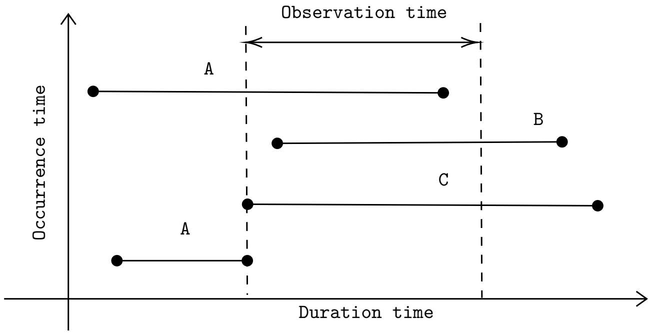


Fig. 5.3: The time of observed epochs.

method based on survival analysis[34] has been used in combination with NetFlow information.

Survival analysis was initially developed to measure the lifespan of individuals. This analysis can be applied to any process duration. For example, it can be related to an HTTP service, where the start of the duration is when users connect, and the end of the duration is when users leave the web service, shown in Figure 5.3. The survival function is defined by Equation 5.1 [63].

$$S(t) = Pr(T > t) \tag{5.1}$$

where T represents random lifetime taken from a set of population and function $S(t)$ is defined as the probability of surviving until at least time t [63], equivalently, it defines the probability, that the death event of subject has not occurred yet at time t. The survival function with the above statement has the following properties: $0 \leq S(t) \leq 1$, $F_T(t) = 1 - S(t)$, where $F_T(t)$ is the cumulative distribution function of T and $S(t)$ is a non-increasing function of t.

Further, right and left censorship is defined. With the right-censored individuals, we only have information about their current lifelines' duration. On the other hand, with the left-censored individuals, we do not know information about their birth (formation, start). The last type of censoring is the interval-censored. In this case, the exact time without the event is not known. We have partially observed the events. The truncation happens when the subjects have been in the study even before entering the study. Survival analysis is a very helpful tool for understanding duration.

[34] Time to event information is often the subject of survival analysis. It includes methods for positive-valued random variables in the broadest sense. Survival data are typically censored rather than fully observed.

The survival curves, shown in Figure 5.4 in which the traffic is transformed using the survival function based on the traffic properties, can then be compared for similarity using one of the clustering techniques [55][53][32].

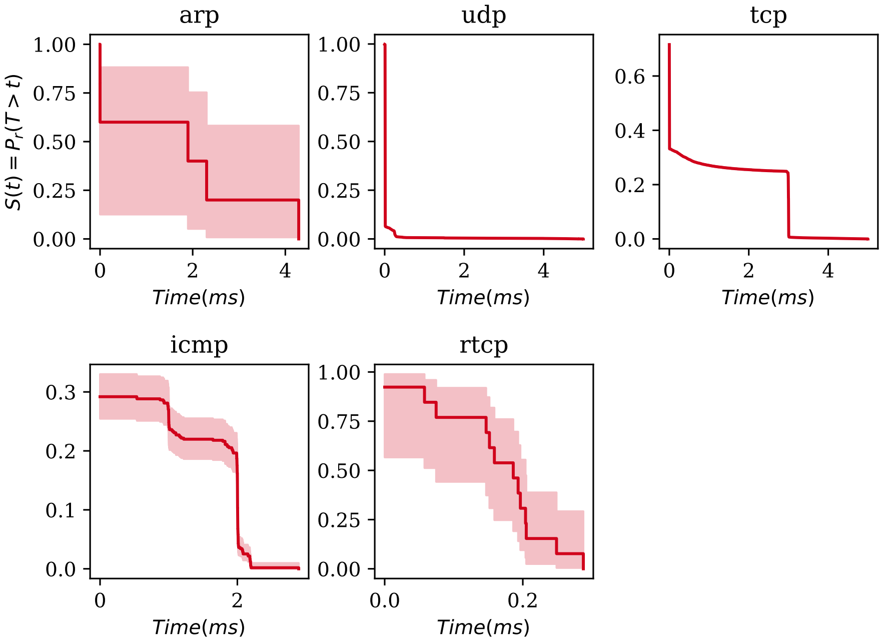


Fig. 5.4: The lifelines of different traffic.

In the case of finding some unwanted traffic, for example, ransomware, it does not decide what the curve looks like. Ransomware generates specific traffic to C&C servers in time sequences and with a particular type of traffic. If traffic capture is done on multiple probes and traffic is converted to curves, the most numerous curves with the most similar waveform can be filtered out using clustering or other method comparing similarity[35].

To estimate the survival function, the Kaplan-Meier estimator is used. It is a non-parametric method; therefore, it does not require knowledge of the probability distribution that governs the survival of individual subjects. It is also called the product-limit method [63]. Kaplan-Meier method gives an estimate of the survival function at every moment, in which can be the monitored event. The Kaplan-Meier Estimate is defined as presented in Equation 5.2 [63]:

$$\hat{S}(t) = \prod_{t_i<t} \frac{n_i - d_i}{n_i} \tag{5.2}$$

where d_i is the number of events in time t_i, eventually the number of completed events in time i and n_i are related to the number of objects still observed in time i.

[35] `https://nsr.utko.feec.vutbr.cz/VI2VS428.php`

The compliance tests of the survival functions are used to compare two survival curves. There are many types of these tests, each of which has optimal properties for different situations. The Breslow test or the Tarone-Ware test can be mentioned. The most famous asymptotically valid tests include the non-parametric Mantel-Cox test, named after Nathan Mantel and David Cox. Sometimes it is also called the "log-rank test".

The censoring process is here independent of the process that leads to the event. Mantel-Cox Chi-Squared is defined [64] by Equation 5.3:

$$\chi^2_{MC} = \frac{(O_1 - E_1)^2}{E_1} + \frac{(O_2 - E_2)^2}{E_2} \tag{5.3}$$

where O_1 is the sum of the events that occur in the experimental group and O_2 is the sum of the events that occur in the control group. The E means the expected sum for each group.

The Mantel-Cox test is also generalized to test more than two test groups. Then, the equation 5.3 is just extended by $k-1$ definitions $\chi^2_{MC} = k_1 + k_2 + k_n$ and each traffic event can be compared with this test and observe its conformity, if two or more flows are similar or not. If there are many curves and groups to compare, the asymptomatic complexity increases by $\mathcal{O}(k_i)$. In this case, a different method, as a clustering or genetic algorithm using the Euclidean distance and the Davies–Bouldin validity index to evaluate an individual (fitness function), may provide better results[55] in terms of scalability. The proposed GA algorithm to cluster the lifelines works only with centroids as individuals.

It does not hold clustered data, as in the case of K-Means, when a population is created using the random choice method, and a paradox of choosing the same centroid occurs with a small number of chromosomes. The genetic algorithm cannot then correct the errors. This paradox can be influenced by increasing the number of chromosomes in the population, possibly increasing the value of the mutation. The convergence time (run time) of the genetic algorithm should be compared to see whether this solution is more successful [65].

❏ *A part of the author's research was presented here. Two methods were compared with each other. In further research, a parallel genetic algorithm was used to increase the efficiency of curve similarity calculation.*

5.3 Machine Learning Analysis

An area of computer science known as AI is concerned with the development of systems that can solve complex problems such as recognition or classification, for

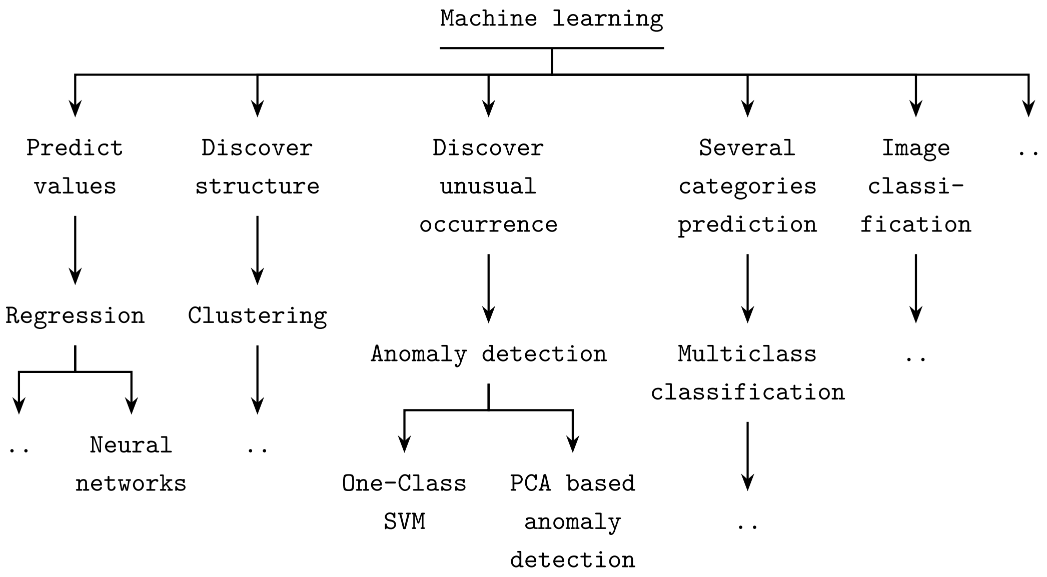


Fig. 5.5: Machine learning algorithms according to their approach [67].

example, in the fields of image processing or the processing of written or spoken language, or planning or control based on the processing of large volumes of data. The term Artificial Intelligence, therefore, covers a set of techniques and approaches such as Machine Learning, Neural Networks, Bayesian Networks, Evolutionary Algorithms, and others. Machine learning has already been used for a wide range of tasks and is particularly crucial for any application that requires the collection, analysis, and act on large data sets [66].

The Figure 5.5 shows the possible division of machine learning algorithms based on the goal of what to achieve with data. There are many more algorithms that are used depending on what the intent is. For anomaly detection analysis, i.e., anomalies in traffic, algorithms like One-Class *Support Vector Machine* (SVM) or PCA Based algorithms can be used. Further, also prediction algorithms such as some type of neural network. There are three main categories of machine learning: supervised learning, unsupervised learning, and reinforcement learning [67].

Each data point in supervised learning is labeled or linked to an interesting category or value. The purpose of supervised learning is to learn from a large number of examples and subsequently be able to predict future data points. For instance, to recognize new pictures. This kind of machine learning is well-liked and practical.

Unsupervised learning uses data points without labeling. The objective of an unsupervised learning algorithm is to explain the structure of the data or arrange them in some way. Unsupervised learning divides data into clusters, much like K-Means does, or searches for novel methods to present complex data in order to make it seem more straightforward.

In reinforcement learning, the algorithm gets a choice of action in response to each data point. The learning algorithm receives a reward signal a short time later, indicating how good the decision was. Based on this signal, the algorithm adjusts its strategy to achieve the highest reward. This is a typical strategy in robotics, where an algorithm must determine the robot's next move based on a collection of sensor measurements at one moment in time that forms a data point.

The following research has focused on the method to evaluate downstream traffic behavior in the GPON ITU-T G.984 [30] transmission protocol on the sequence of *Physical Layer Operation Administration and Maintenance* (PLOAM) message ID[36]s during the activation process [68, 69]. The processes that an inactive ONU takes to connect or reconnect to a PON are described by the activation procedure. Three stages comprise the activation process: parameter learning, serial number acquisition, and ranging. The ONU obtains the operational parameters required for the upstream transmission during the learning parameter phase. The OLT finds a new ONU (by serial number) and gives it an ONU identifier (ONU-ID) during the serial number acquisition phase. There are several states and messages used for negotiation during the activation process defined by the Recommendation ITU-T G.984.3 [70].

The objective of the solution was to use and test a machine learning-based solution for the verification of the activation phase to determine if the device under test complies with the defined recommendations and the standard, specifically, the PLOAM messages. Whether the concurrence of PLOAM messages is in accordance with the standard and whether their content is in accordance with the standard [71]. Traffic data from the network was collected using the FPGA network card, and GPON frames, shown in Figure 5.6, were parsed using a dedicated software parser [72]. PLOAMd fields were extracted from each frame.

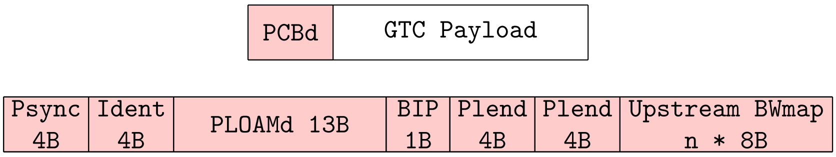


Fig. 5.6: GPON frame in downstream direction.

[36] Identification number, abbreviation for identification

The analysis of the frames takes into account two areas of the audit:

- **Syntax verification** – examining the message to see if it complies with the standard, verification of each field content in GPON header, whether it is similar to patterns from baseline traffic or not. For the testing, One-Class SVM and AutoEncoder[37] have been selected. This is because there are typically few outcomes, and more significantly, these outcomes usually do not have any further structure, and binary classification is in use.
- **Sequence verification** – controls the continuity of individual messages and the content of the respective fields between messages. The analysis of patterns verifies whether the analyzed protocol uses the same message in the same order and with similar content. The *Long Short Term Memory* (LSTM) network and AutoEncoder are used for testing. This is because in the sequence verification is required to retain the information gained from previous data, and the LSTM can store processed information about the longer data sequence.

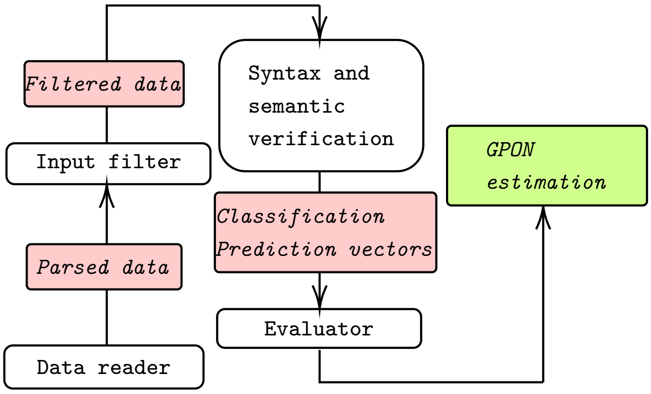


Fig. 5.7: The system diagram of the solution.

The system diagram of the analysis model is shown in Figure 5.7. The primary responsibilities of the data reader component include reading data in a certain format, loading it into memory, preprocessing RAW data, and storing them in a particular file format. The following component receives an n-dimensional array of read data.

The input filter component is responsible for filtering or normalizing the input data, which may contain unanticipated errors or differ in length. It also offers to filter or minimize lengthy sequences of the same messages to aid in the learning process and prevent incorrect learning.

The content of each message field is subject to unique criteria that must be considered during the analysis. Such an evaluation is carried out via the syntax

[37] AutoEncoder is not a classifier, but it can be used as a layer before classification layers. The reason to use AutoEncoder is to get a better representation of inputs, it is a dimensionality reduction technique like PCA, but it is a nonlinear dimensionality reduction.

verification component. The two implementation choices considered various learning data. The first uses a supervised learning algorithm in a deep neural network that needs both positive and negative input data to learn. The benefit of this strategy is that the model can pick up on more intricate relationships.

The second choice is to employ one of the outlier detection models that can distinguish between data that is similar to and distinct from the training dataset. Unsupervised learning is used in this method; hence no new data is required for learning. But in contrast to the deep neural network, it might not find intricate relations.

The semantic verification model evaluates message relations in time and LSTM cells to perform a time-based sequence check. Finding patterns in time is an issue that is solved by the system's initial component. This section can be utilized for message sequence analysis under the condition that the state is not used for language translation but rather a forecast of whether it fulfills the GPON recommendation.

The categorization or prediction of earlier models is statistically evaluated by the evaluator module. It generates aggregated output based on these data, reflecting similarity to baseline traffic.

For practical implementation, TensorFlow and Keras can be used. The TensorFlow library is being developed by the Google AI research community. TensorFlow model or program creation involves two steps: defining a static computation graph and conducting a computation session on this graph. Based on this model representation, data evaluation in pipelines is simple, especially on GPUs, enabling the development and learning of deep neural networks. The core idea behind TensorFlow computations is the creation of directed networks that are used for calculation and where a node may, for instance, be a value or math function, and an edge could be a tensor. There are zero to N inputs and outputs for each node. When a node represents a math function in TensorFlow, the implementation of that node on a specific hardware component, such as a GPU, is where the kernels are assigned [73].

Project Keras [74] contains two main model definitions. The sequential is the initial model definition. This approach can build a fully operational NN model by layering abstract *Neural Network* (NN) layers. The second is a basic model that offers an API for individualized model definition via functional description or inheritance. The definition of two models with various levels, but one of these layers is shared by both models, is one of the use cases. The initialization, training, and prediction of the two model classes listed above all use the same fundamental API.

When testing PON traffic, it requires artificially (manually) creating modified traffic sets. For example, in the case of PLOAM message sequences, it is necessary to create non-conforming standard sequences. It is hard to get a corrupted one from a long-time measurement.

Outlier detection using One-Class SVM classifies normal and abnormal GPON frames sufficiently, but it uses an approximate function that is unable to learn the importance of frame field usage. The LSTM model can distinguish time sequences. The disadvantage is that the model needs to be learned with corrupted or improper communication samples, which are not available in all cases. In combination with the AutoEncoders, the AutoEncoders prove their outlier detection capabilities and make a great alternative to the LSTM model, especially due to unsupervised learning [71].

❏ *A part of the author's research was presented here. A personal contribution was in the design of a method to analyze GPON frames for their verification. In practical terms, as determined during testing, the use of machine learning in the analysis of frames requires re-learning the models when the network topology changes. Otherwise, as required, an unidentified unit is detected.*

5.4 Analysis using Artificial Immune Systems

AIS are adaptive systems inspired by the biological immune system. Cells, such as neutrophils, macrophages, and dendritic cells, are present in innate immunity. The T and B lymphocytes play a significant role in adaptive immunity. In recognition, the presence of receptors that have the ability to recognize and capture a pattern is important to us. This part of the cell is called the antibody. Each antibody is capable of identifying a particular pattern. This pattern is called the antigen. The principle of antigen-based pattern recognition could be compared to a key and a lock, shown in Figure 5.8.

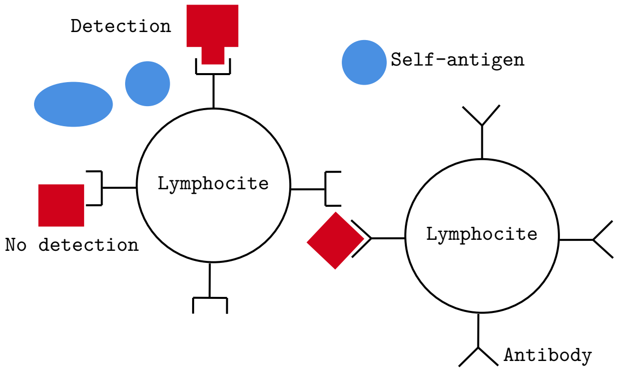


Fig. 5.8: The model of AIS system.

There are also other antigens called self-antigens or self-items. These are proteins that are naturally occurring. The antibody is only capable of recognizing and binding to one particular antigen. The adaptive part of the immune system has memory. In artificial immune systems, this memory represents a trade-off between memory requirements and the rate of an immune response. The use of AIS is extensive; a basic overview is shown in Figure 5.9, from classification and robotics to bioinformatics.

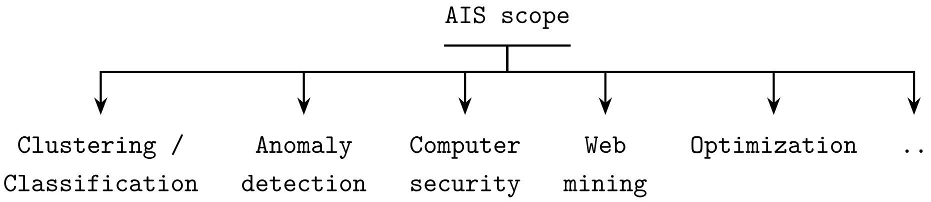


Fig. 5.9: The artificial immune system use.

There are basic algorithms used in artificial immune systems [75]:

- Negative selection algorithm (Forrest, Gasgputa, Kim, -...).
- Clonal selection algorithm (Clonalg – De Castro, B-cell – Kelsley).
- Immune Network algorithm (models) – continuous models (Farmel, Jerne, discrete models – RAIN (Timmis), AiNET (De Castro).

The principles of the respective algorithms are well-described by [75][38]. The techniques and algorithms used within the AIS are affinity[39] functions related to distance or similarity (Hamming, Euclidean, Manhattan), the Bone-marrow algorithms, or somatic hyper-mutation. The operators used include affinity evaluation operator, operator assessment of individual levels, incentives meter calculate son, immune selection operator, clone (individual multiplication) operator, mutation operator, operator, and population suppression cloning Refresh operator [76]. The *Clonalg selection* algorithm has the following layers:

- Initialization – creates a random population P.
- For each antigenic pattern in a data set S do:
    - Affinity evaluation – present it to the population P and determination of affinity with each element in the P.
    - Clonal selection and expansion – n highest affinity elements of P are selected and clones are generated proportional to their affinity with the antigen, a higher affinity creates more clones.
    - Affinity maturation:
        - Each clone mutation – when high affinity, low mutation rate, and vice versa.
        - Mutated individuals are added to the population P.
        - Best individuals re-selection, kept as memory m of the antigen.
    - Metadynamics – n individuals with low affinity are replaced by randomly generated new n.
- Cycle – the second step is repeated until a certain stopping criterion is met during the cycle.

[38] Artificial Immune Systems: Part I-Basic Theory and Applications – `bit.ly/3B6JxCD`

[39] Affinity – Strength of ligand (molecule) binding to its receptor.

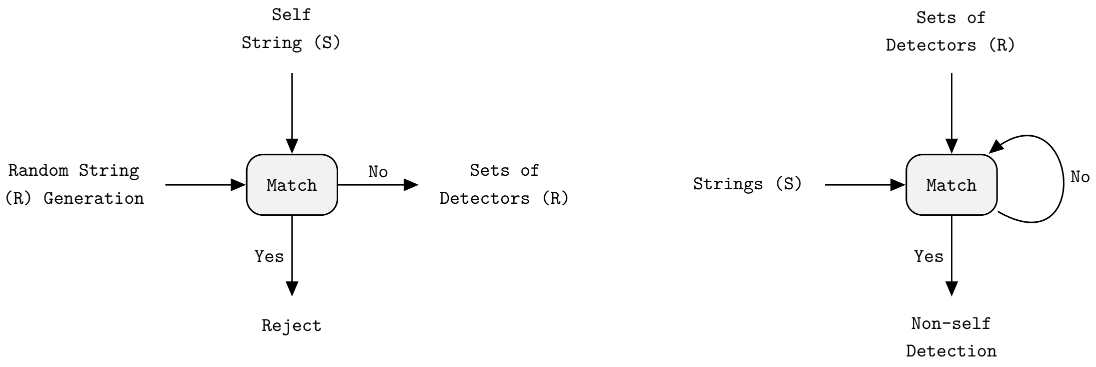


Fig. 5.10: The negative selection algorithm.

The *negative selection* algorithm, shown in Figure 5.10, has the following layers:

- Self-definition as normal pattern:
  - A set of equal size of the pattern sequence.
  - The set is presented as a multi-set S of string of length l over a finite alphabet.
- A set of R detectors is generated, each of which fails to match any string in S.
- The changes of S are observed by testing the R matching the S, and if any R matches, the non-self is detected.

When working with AIS, the optimization of the high density of individuals is constrained to ensure the diversity of individuals in a given solution. The antibody-antigen affinity density is defined by Equation 5.4 as the SMA of antibody-antigen affinity [76].

$$den(Ab_i) = \frac{1}{m}\sum_{i=1}^{m} aff(Ab_i, Ag_i), \tag{5.4}$$

where Ab_i is the antibody of the i-species, the size of the population is defined by m and aff is the antibody-antigen affinity for i.

The antibody-antigen affinity aff is mainly based on the method of calculating the Euclidean distance, Hamming distance or the information entropy. If the aff is smaller, a stronger affinity is encountered; thus, the stronger ability of the antibody to capture and kill an antigen, in the case of network analysis, does the detection. Problems like function approximation and optimization are being solved using a variety of machine-learning techniques.

Among these are NN for non-linear function approximation and the use of genetic algorithms to find a function's optimum (maximum or minimum). The starting collection of candidate solutions for the GA or initial weight vectors for the ANN must be defined for both procedures [75].

Antibodies can be modeled as a string of bits of length l. As a binding of the antibody to an unknown element, a string match of bits on the antibody and a string of bits on the antigen (unknown element) is considered. In order for the antibody to bind to the antigen (the affinity), it is necessary that the chains equalize each other. But this is hardly a feasible requirement, so the principles of string similarity are used. These include the Hamming distance, the Euclidean distance, or the Manhattan distance, as mentioned above. It can be a binary representation, continuous as numeric, or categorical. Assume the general case $Ab = \langle Ab_1, Ab_2, \ldots Ab_i \rangle$ and $Ag = \langle Ag_1, Ag_2, \ldots Ag_i \rangle$. The binary representation typically employs the Hamming rule, and the numeric (real or integer) representation is typically Euclidean. The wast of the AIS includes Euclidean, but also Manhattan. But these two can differ in results. The Euclidean is more sensitive to noisy data, and the Manhattan is the opposite, robust to noisy data, but the computational complexity will be different.

When using the principle of AIS in clustering base algorithms, data items are seen as antigens and clusters as antibodies. The process by which the immune system repeatedly creates antibodies to detect the antigen and finally creates the best antibody that can capture the antigen is analogous to the clustering of the data items [77] and clustering K-Means, K-Nearest based algorithms are used.

The data representation that is best suited to the issue should be selected, and the selection rule is subsequently selected or adapted to represent the data by the other operators used. Detectors can be made without knowledge of the problem domain other than the known data sets. The algorithm can be configured to balance detector convergence, that is, the quality of the match, and spatial complexity, that is, the number of detectors.

5.4.1 Mapping selected problem into the AIS

From the author's research [21], applying the same problem as in Caption 5.3 for the activation process, in the case of a recognition system, antigens correspond to the GPON frames whose contents may contain some unidentified field, for example, unspecified PLOAM fields and its content by the standard. Self-antigens match the known parts of the GPON frame. An antibody presents a bit pattern that matches a potentially unknown part of the specified GPON frame. A lymphocyte represents two or more detectors. The cell apoptosis[40] is modeled using the negative selection algorithm. The steps of the algorithm are as follows [21]:

- Input: self-set (known patterns). Output: New set P of detectors.
- Initialization of the empty set P of detectors (memory cells) and determination of the affinity boundary (maximum similarity between the detector and some element of self-set).
- Creating a random detector.
- Determining the affinity of the detector step by step for all elements of the self-set and as the detector affinity will be considered the highest one.
- If the solution has less affinity than the determined affinity limit, add the detector to the P set.
- If the P set is large enough, terminate the algorithm. Continue with the detection.
- Input: Set P, Data D.
- Count affinity of Set P_i, Data D_i.
- If affinity and matches D_i, P_i – detection.

The operators used are, in this case, hard-coded; thus, they are not included in the algorithms' steps. The optimization function is done using GA; therefore, this is a mix of genetic and K-Nearest algorithms. When comparing the ML Syntax model and AIS algorithms, the AIS algorithm showed a higher success rate when recognizing modified strings of data, in this case, PLOAM messages, but only when the ratio of modified strings was lower than the correct messages. However, the performance aspect and the processing time must be considered. It fails when the set of valid messages is lower than the possible set of non-self. The negative selection algorithm serves to detect detectors that can only recognize foreign elements. Thus, this algorithm removes the elements that recognize the known element. It is used where the known set is much larger than its complement.

❑ *A part of the author's research was presented here. The use of AIS instead of neural network to detect the contents of specific fields of GPON frames was investigated. The author's contribution was the design, programming and testing of the method used.*

[40] The death of cells that occurs as a normal and controlled part of an organism's growth or development.

6 Network Data Analysis and Reporting

This chapter presents the current data monitoring situation from a legislative perspective and from the standpoint of active elements. Then the principle of the possibility of analysis and monitoring of passive optical networks is explained, which is one of the author's contributions to this topic.

The author's contribution: The research was directed toward the design and development of an active network element and the implementation of algorithms such that would enable efficient analysis of transmitted data structures in a real-time optical access and distribution network on the GPON ITU-T G.984 [30] transmission protocol, and the continuous research on the XGPON ITU-T G.987.1 [31] [28] [3]. This research also involved the upgrade of existing data processing algorithms and algorithms used in the first sub-objective research. During the research, it was also necessary to resolve issues related to a large amount of data processing for the analysis itself and the preparation of data for the individual processes [78]. The obvious solution was to introduce the parallelization of computational processes and algorithms [37]. Part of the research and development involved designing a system that would be able to analyze traffic in passive optical networks in real-time. Such a system needed to be improved on the market at the time the research was initiated. Real-time traffic analysis is provided mostly at the higher Ethernet layers. However, the intent of the research conducted was to analyze real-time management traffic to detect deviations from the standard, errors, or unsolicited traffic directly at the passive optical network layer.

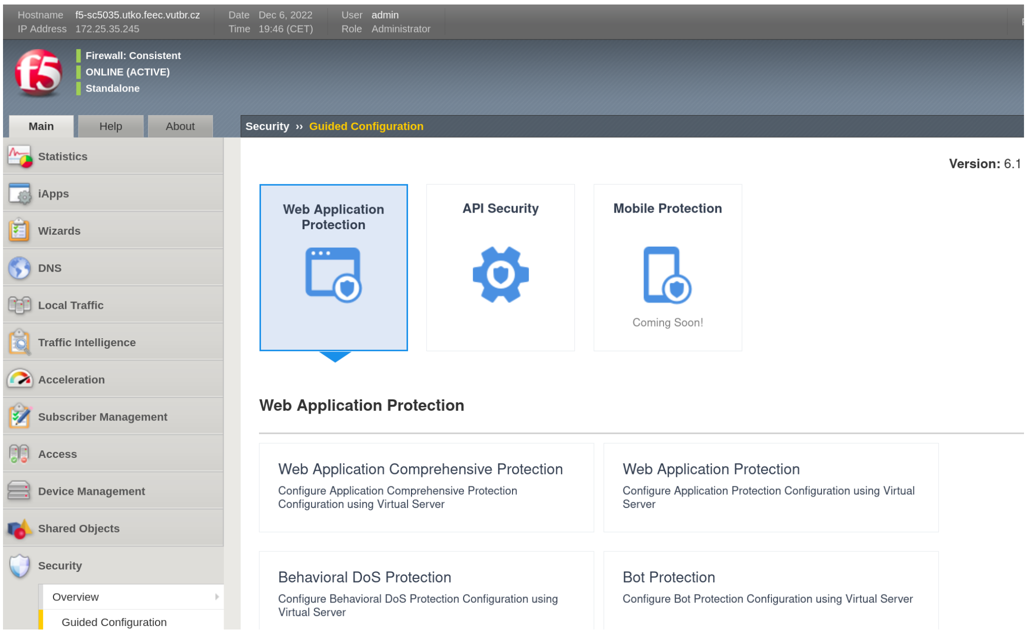
Data structure analysis is nowadays equally important. In order to further strengthen the resilience and incident response capabilities of the public and commercial sectors as well as the *European Union* (EU) as a whole, the Council and the European Parliament agreed on steps for a high common level of cybersecurity across the Union. The present *Network and Information Systems* (NIS) directive regulation on the security of network and information systems will be replaced once the new directive, known as NIS2 [79], is implemented. The baseline for cybersecurity risk management practices and reporting requirements will be established by NIS2 for all of the sectors covered by the directive, including energy, transportation, health, and digital infrastructure.

For example, in the Czech Republic, the NIS2 Directive should be implemented into national law by 2024. One of the goals is that everyone who falls under NIS2 must secure their systems. The regulation will also apply to small and medium enterprises with more than 50 employees or with an annual turnover of at least

€10 million. European Commission Recommendation 2003/361/EC of 6 May 2003 defines the parameters of companies. Security requirements are also imposed by the trustworthiness of the device manufacturer. The question is how to verify such trustworthiness. One answer may be to verify the functionality and communication of the devices to see if they adhere to the recommended standard. If they are proven to comply, such devices can be considered trustworthy in terms of the fact that, for example, traffic management frameworks do not contain undefined fields, thus not hiding potential backdoor risks. This is also important from a supply chain perspective.

A range of tools is used by security teams to investigate and mitigate breaches. The selection of specific tools is left up to the discretion of the many separate teams; there is no uniform standard that dictates its use. Teams are given particular advice and guidelines by organizations like *European Union Agency for Cybersecurity* (ENISA) and *National Institute of Standards and Technology* (NIST). *Intrusion Detection Systems* (IDS), *Intrusion Prevention Systems* (IPS), *Security Information and Event Management* (SIEM), and *Security Orchestration, Automation and Response* (SOAR) are a few broad categories of the technologies used.

Network traffic is analyzed and tracked by IDS systems using known signatures or abnormalities. With the ability to actively block and filter traffic, IPS systems expand the capabilities of IDS. Then, notifications are generated by both IDS and IPS systems for additional examination. One such example of an integrated solution


Hostname f5-sc5035.utko.feec.vutbr.cz
IP Address 172.25.35.245
Date Dec 6, 2022
Time 19:46 (CET)
User admin
Role Administrator

Firewall: Consistent
ONLINE (ACTIVE)
Standalone

Security ›› Guided Configuration

Version: 6.1

Web Application Protection

Web Application Comprehensive Protection
Configure Application Comprehensive Protection Configuration using Virtual Server

Web Application Protection
Configure Application Protection Configuration using Virtual Server

Behavioral DoS Protection
Configure Behavioral DoS Protection Configuration using Virtual Server

Bot Protection
Configure Bot Protection Configuration using Virtual Server

Fig. 6.1: BIG-IP traffic manager.

is shown in Figure 6.1[41]. This is a comprehensive traffic manager solution from F5, Inc. that integrates both IDS and IPS services called BIG-IP and can also send continuous reports to logging systems.

SIEM systems aggregate and analyze data obtained from various network or system logs. This system also uses ML for the analysis. The IDS builds a predictive model (i.e., a classifier) to differentiate between intrusion or attacks and regular connections. For example, the SIEM has integrated prebuilt machine learning to detect host and network anomalies automatically. Therefore, potential threats can be identified, and alarms can be automatically generated based on this data. Conversely, SOAR systems collect and centralize data and alarms from different resources for further use.

The following author's research and work have been motivated by the non existence of a solution for GPON and XG-PON networks related to the verification of standard and automatic reporting features. The security features specified by the standards for the PON network are based on the assumption that eavesdropping on the signal is not trivial. Uplink transmission in the two-direction communication of the PON network is therefore considered secure. However, this may only sometimes be true in a real environment. Splitters[42] are commonly installed, for example, in basements where they are not further secured. If some of the ports on the splitter are free, an attacker can easily connect to the network; otherwise, it is enough to disconnect a legitimate user from the network and connect an attacker's device known as Rogue ONU, shown in Figure 6.2. Several known types of security risks

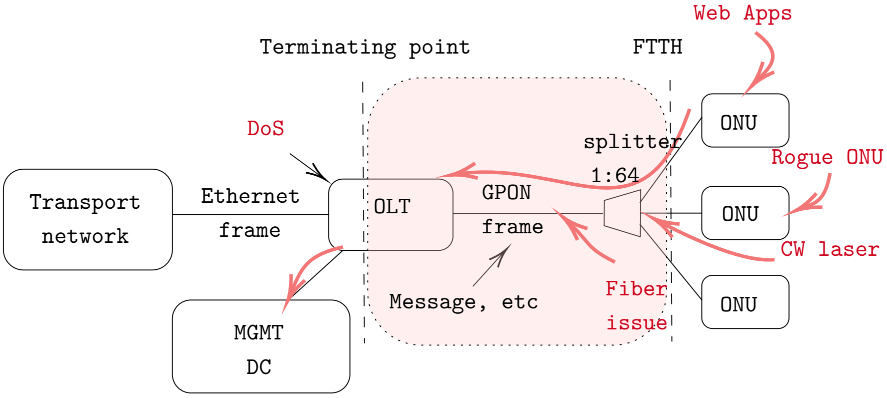


Fig. 6.2: Security risks in a GPON network [3].

[41] The author of the thesis is also the author of laboratory exercises, and this solution is implemented in the transport networks laboratory in the courses.

[42] The passive optical splitter can split, or separate, an incident light beam into several light beams at a certain ratio.

should not be ignored within PON. It also presents some other of the security weaknesses of GPON. Modified ONUs represent one of the most significant security risks in the PON network. For example, they can be used for attacks such as *Theft of Service* (TOS), Masquerade, or Reply Attack. *Denial of Service* (DoS) attacks make a network service unavailable to legitimate users. The blocking of upstream communication occurs when an ONU transmits outside of its allocated time slots. The root cause of a DoS attack also can be a hardware or software malfunction of an ONU. However, an attacker can deliberately modify an ONU to transmit continuously on a given wavelength and with sufficient transmit power to block the communication of other ONUs. The attack can be realized with a sufficiently powerful laser beam source [80]. Specific exploits are available in the Exploit Database [81].

As part of the research, the exploitation of exposed ONUs on the Internet, as well as their "Web App" API interfaces, and how it would be possible to paralyze an internal network [82] has been investigated. The research focused on PLOAM messages used in the management communication between OLT and ONU. These messages are used to transmit control and monitoring instructions between the OLT and the ONU, but in the theory, it would be possible the messages can carry unwanted instruction. It has been found, that some vendors do not follow the standard, and thus, undefined messages are present in these PLOAM messages.

The individual security tools work with *Indicators of Compromise* (IoC), forensic data discovered during network or system monitoring indicating potential intrusion or malicious activity. In general, these can be, for example, IP addresses, malware signatures, domain names, malware file hashes, and more [83]. Another example can be unusual activities such as data found in system log entries, unusual network traffic, bundles of data in the wrong place, etc. According to the research findings, most systems are designed for IP networks, but little attention is paid to the optical access network. The IoCs are *not defined for PON networks*. For example, based on the detection of non-standard PLOAM messages, these particular messages could be considered as IoCs. Several possible IoC to be used for PON system automation and monitoring have been proposed with this research.

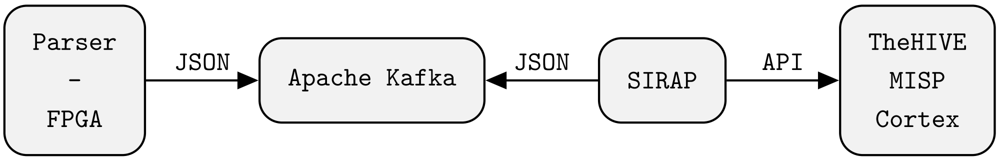


Fig. 6.3: The functional diagram of the reporting solution.

Security Incident Response Automation for xPON (SIRAP), shown in Figure 6.3, consists of an API and parts, published in [3], that provide a collection and analysis of reports from PON networks. The SIRAP also uses modules that connect it to existing incident reporting tools using specific IoC identifiers for PON. The SIRAP is a middleware between the rest of the modules processing data and tools used for reporting.

Specifically developed FPGA card [28] for data acquisition is connected to capture data on the PON layer and transfers frames from the optical network (downlink and uplink direction) to the server's *Direct Memory Access* (DMA). The frame parser [72] is a C# application parsing the traffic from the FPGA, creating a JSON with parsed frames and sending them for further processing to the Apache Kafka. The Apache Kafka is used to scale the solution when working with high-speed data, buffering the messages. The data analysis module included in the SIRAP is based on TensorFlow, meaning that it is a TensorFlow detector. Once the traffic is analyzed and if unspecified or unsolicited traffic is found, a report is created according to a template specific to TheHIVE[43] and sent via API to TheHIVE. So far, the following report types have been defined:

- **PLOAMd Anomaly** – notification of anomaly detection in a PLOAMd message, i.e., a deviation from the messages specified by the standard.
- **Activation process anomaly** – notification of anomaly detection in activation process.
- **Non-standard Frame structure** – notification of anomaly detection in PON Transmission Convergence Layer frames.
- ***ONU Management and Control Interface* (OMCI) Anomaly** – notification of anomaly detection in OMCI messages or the *ONU Management and Control Channel* (OMCC) activation process channel.
- **Non-specified error** – the empty report, used as a template for the specification of a new report type.

This solution allows both monitoring of the passive optical network and defining custom templates for incident reporting. Reports can be created using the specified API. These reports must contain all the following mandatory attributes:

- `title`: the title of the report.
- `description`: a description of the report,
- `severity` (number) : hard-coded per PON report type.
- `date` : the date and time of an alarm triggered, the default value is date(now).
- `tags` (multi-string) : defined by the PON templates (XPON, GPON).
- `tlp` (number) : *Traffic Light Protocol* (TLP), default value = 2, when user data included = 4.

[43] TheHIVE `https://thehive-project.org/index.html`

- `status` (AlertStatus) : the status of the alert (new, updated, ignored, imported), the value is hard-coded for the PON = new.
- `source`: the source of the message (SIRAP, ML).
- `sourceRef`: report reference, e.g. system ID.
- `artifacts` (multi-artifact) : list of indicators carrying each attribute. General data type defined.

The combination of the `type`, `source` and `sourceRef` attributes is unique for each message. If a report with the same combination of these three attributes already exists in the system, the creation of a new report is rejected. The defined reports carry an external type, since they come from an external source, i.e. the PON analyzer. A combination of values describing the origin of the message received from the Kafka server is used as a reference.

❏ *A part of the author's research was presented here. The example above provided a look at a custom solution for interpreting data analysis results when a reporting solution is not available for a particular technology or protocol.*

7 Verification and Interpretation of Results

The previous chapters have partially presented methods for verifying results related to each selected traffic analysis technique, such as for clustering algorithms. This chapter provides a general summary overview dealing with the verification and interpretation of results obtained from network analysis. The purpose of this chapter is to summarize in general terms the requirements for the last part of the traffic analysis presented in Chapter 2, Figure 2.2, namely the general procedure for verifying the results. The following pattern of approach is more a general conception of the author of the thesis than an established paradigm, and it is open for discussion.

7.1 General approach

The general verification model can serve well as a pattern. A pattern [84] is a way of regulating a group of different accesses using a mechanism that is defined in a particular environment. The goal of the pattern is not to eliminate the weaknesses of the approach; rather, it is to minimize or mitigate the error of the approach. When working on interpreting the results of traffic analysis, it is useful to have basic access parameters set. The following Figure 7.1 shows in general the first step of the verification of results. First and foremost, it is necessary to ask whether the analysis performed is correct. If possible, it is advisable to compare and quantify the results with some known data. Accuracy and precision are two measures of observational error that can be observed. Precision in meaning means how close a given set of measurements or observations is to its true value, while accuracy means how close the measurements are to each other.

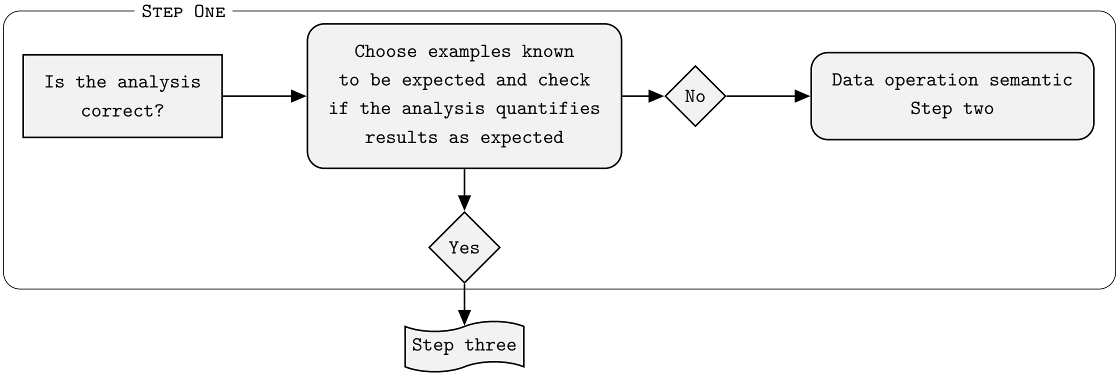

STEP ONE
Is the analysis correct?
Choose examples known to be expected and check if the analysis quantifies results as expected
No
Data operation semantic Step two
Yes
Step three

Fig. 7.1: The pattern of the approach, step one.

The accuracy is defined by ISO 5725-1:1994/Cor 1:1998[44]. In previous chapters, some techniques for such measurement have been presented for specific purposes. For example, in multiclass classification, the accuracy is the fraction of correctly determined classifications. For clustering, different metrics are used. A good overview is given in scikit learn [45]. The metrics used can be, as already mentioned, silhouette, ARI, or DB. If the measurements do not conform, it is advisable to review the procedure, the values used, and the semantics of the code. An example is the silhouette measurement for the research paper in Section 4.2. The following Figure 7.2 shows the measurement scores using the silhouette metric.

```
K-Means:        0.6384384757932036
K-Means Cosine: 0.8288671473155667
DBSCAN:         0.6055154609146888
OPTICS:         0.820598603207504
```

Fig. 7.2: Silhouette score for different methods.

The results between K-Means and normalized K-Means are expected, given how many features are in the dataset. Interestingly, however, DBSCAN also has good results. It is worth going back to the values used for each clustering algorithm, the number of minimum samples, or the type of metric used. Another example is if a GA was required to be rated, the situation is again perceived differently. In practice, empirical tuning has been widely used so far. The best tuning of the

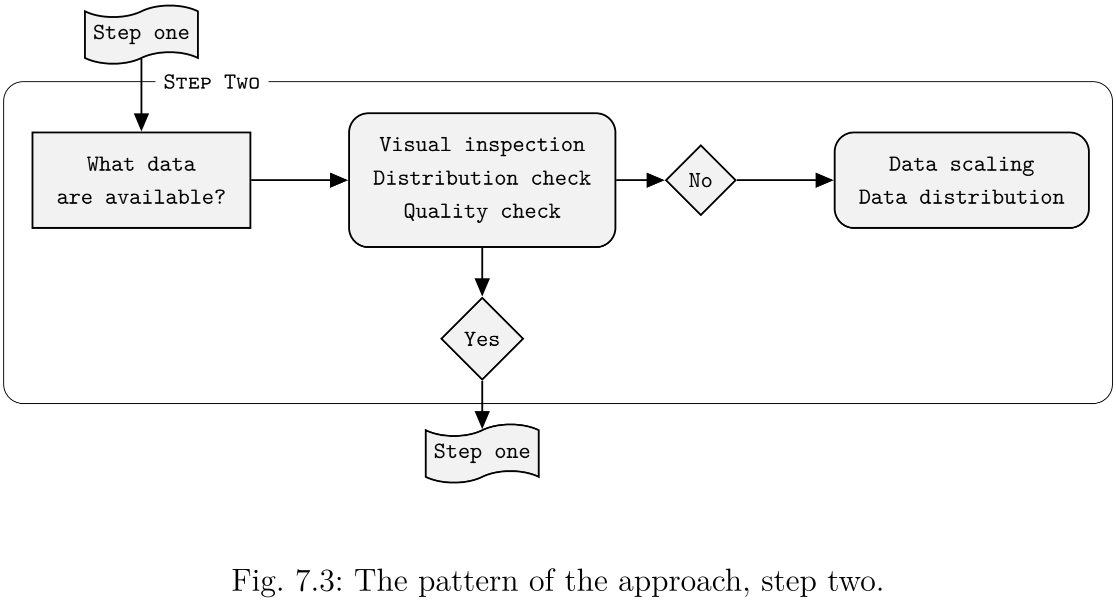


Fig. 7.3: The pattern of the approach, step two.

[44] https://www.iso.org/standard/29779.html

[45] https://scikit-learn.org/stable/modules/clustering.html#clustering-evaluation

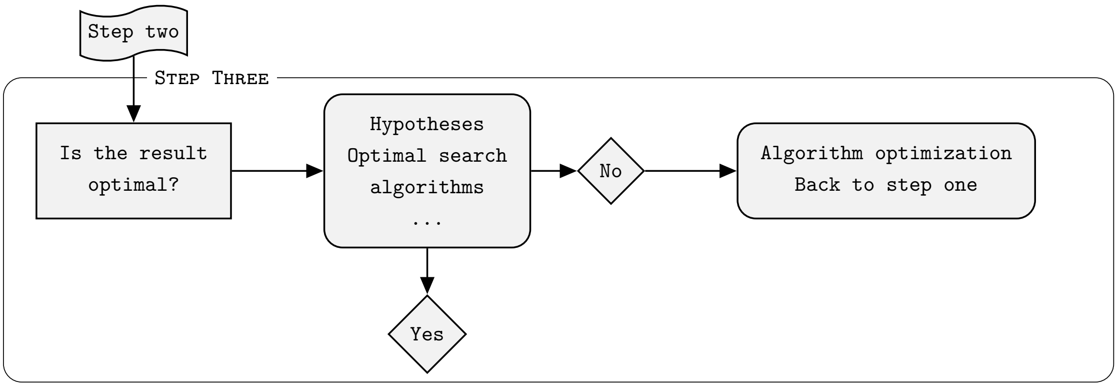


Fig. 7.4: The pattern of the approach, step three.

actual GA parameters is determined by extensive experiments on their performance using simulation. Either self-tuning or hierarchical tuning is used, where another GA manipulates the parameters of the measured GA and tracks the best performance.

Therefore, if the results are satisfactory, it is possible to proceed to step three, shown in Figure 7.4. If they are still not, it is advisable to go to step two, shown in Figure 7.3, and be interested in the data itself. It can be helpful to visually inspect the results by displaying them using subsidiary graphs. For now, there is nothing that completely replaces the experience acquired, or at least some general idea of the result. The quality control should look at what data is available and whether it contains unknown values. For example, whether they contain *Not a Number* (NaN) or null values.

The possible outcomes and values are limitless, based on the tool and type of measurement used. It's also good to find out what the data layout is. Whether the data is from a normal (Gaussian), power law, or different distribution. The $Q - Q$ (Quantiles Quantiles) [85][46] plots, for example, can be used to determine the distribution. The data needs to be normalized and scaled based on their distribution. The data preparation techniques are out of the scope of this thesis; some of them were presented in Chapter 4. The choice of technique depends on the type of data. Analysis algorithms will calculate more precisely when the data are normalized and scaled. Once the data analysis correction has been performed, it is advisable, if possible, to subject the results of the analysis to validation and interpret the results appropriately. In the case of evaluating the optimal solution or some other optimality, hypotheses are used. In terms of statistics, the optimality criterion provides some measure of the fit of the data to the hypothesis and helps in the selection of the

[46] From the author's seminar related to packet analysis: `https://is.muni.cz/predmety/predmet?pvysl=15366604;lang=en;kod=PB156CV;fakulta=1433;jazyk=en`

model or procedure used. For example, in traffic analysis practice, it is determined how optimal an algorithm is. In the case of machine learning, overfitting and underfitting are distinguished[47]. These processes should reflect the current situation, and the analysis results should be subject to repeated inspection. Repeated verification provides a means to ensure that traffic analysis is adequate and accurate.

❏ *This chapter provided a closing of the overall network traffic analysis topic. The proposal to create patterns is one of the topics for network security, not only for network traffic analysis. Another part of the author's research, not mentioned in this thesis, deals with the creation of security patterns for modern networks.*

[47] `https://www.ibm.com/cloud/learn/overfitting`

Conclusion

This habilitation thesis aimed to present a summary of the author's work and research in the field of network traffic data analysis. The field of networking technology is evolving at an unstoppable pace, and there is a need to respond to this evolution in a very flexible manner. Therefore, the different aspects of the thesis have been presented to reflect at least the current topics. The content of the thesis was divided into two main areas, namely theoretical and practical, which dealt with selected technologies and techniques for network traffic analysis that the author has worked with both in his projects and collaborative research activities. They also draw on experience in commercial and academic environments.

The thesis consists of seven chapters in total. The first chapter stated the objectives of the thesis and the author's own contribution and organization of the thesis. The next chapter dealt with the current knowledge in the field of traffic analysis. A generic view of the problem and of the various traffic analysis techniques was presented, with an indication of the basic mathematical relationships later in the text for each technique. Furthermore, the chapters were structured according to the generic view of traffic analysis. First, the principles of data sources for traffic analysis were explained, then how the data are processed, and then in which techniques they are used were presented. These were mainly selected techniques covering the author's contributions and publications in the field. Thus, it was not an exhaustive list of techniques in use today, but a view of some of them was offered. In particular, this included the author's proposed technique for making data localization more efficient, as well as the author's proposed technique and algorithm for analyzing traffic behavior based on the clustering algorithm and the use of the survival algorithm. Another technique discussed the use and aspects of artificial intelligence, of which the less-used traffic analysis technique using the artificial immune system is represented here. Subsequently, a practical example and the issue of monitoring and its possible solution for passive optical networks were presented. The whole work is designed to have a contribution to education as well. Thus, each chapter contained a part of the theory on the problem and a part dedicated to the author's work.

In general, there is no one-size-fits-all instruction on how to work with data in traffic analysis. The author's recommendation is to work with intuition and also use approaches used in other research fields, such as in the medical or nature field, and to use own imagination to the fullest. This aspect was reflected in the results of the author's work.

The thesis presents the results of the author's work achieved from the completion of his Ph.D. studies in 2017 to the present. Further, ongoing research in this area is mainly focused on improving existing methods of traffic behavioral detection and their other applications, especially within the project “Data backup and storage system with integrated active protection against cyber threats”, and “Android federated learning framework for emergency management applications”. The first project focuses on creating a solution for the early detection of ransomware before backing up virtual machines. The second project focuses on developing a framework based on federated learning for secure data transfer and decentralized learning for crisis management applications.

The author of this thesis is the author or co-author of 30 articles and 29 conference papers, most of them indexed in Web Of Science, Scopus or with an impact factor, with a total of over 160 citations at the time of writing this thesis.

Google Scholar ID: `https://bit.ly/3UBmgj7`
Scopus ID: 57160133400
Orcid ID: 0000-0001-7629-6299
Research ID (WoS): Q-9784-2017
SciProfile ID: 533908
Credly: `https://bit.ly/3BbDRHi`

Bibliography

[1] *Network Research Group.* [Online; accessed 2022-09-09]. URL: https://nsr.utko.feec.vutbr.cz/.

[2] Jun Shan Wey and Junwen Zhang. "Passive Optical Networks for 5G Transport: Technology and Standards". In: *Journal of Lightwave Technology* 37.12 (2019), pp. 2830–2837. DOI: 10.1109/JLT.2018.2856828.

[3] Vaclav Oujezsky, Tomas Horvath, and Martin Holik. "Security Incident Response Automation for xPON Networks". In: *Journal of Communications Software and Systems* 18.2 (2022), pp. 144–152.

[4] *ESG WHITE PAPER. Network Traffic Analysis (NTA): A Cybersecurity 'Quick Win'.* 2020. URL: https://www.cisco.com/c/dam/en/us/products/collateral/security/stealthwatch/stealthwatch-esg-wp.pdf (visited on 11/19/2022).

[5] Petr Velan et al. "A survey of methods for encrypted traffic classification and analysis". In: *International Journal of Network Management* (2015). URL: https://doi.org/10.1002/nem.1901.

[6] Andrew Moore, Denis Zuev, and Michael Crogan. "Discriminators for Use in Flow-based Classification". In: *Queen Mary and Westfield College, Department of Computer Science* (2005). ISSN: 1470-5559. URL: https://qmro.qmul.ac.uk/xmlui/bitstream/handle/123456789/5050/RR-05-13.pdf.

[9] P4.org Applications Working Group. *In-band Network Telemetry (INT) Dataplane Specification.* URL: https://github.com/p4lang/p4-applications/blob/master/docs/INT_v2_1.pdf (visited on 11/28/2022).

[13] NIST/SEMATECH. *e-Handbook of Statistical Methods.* URL: https://doi.org/10.18434/M32189 (visited on 11/25/2022).

[14] C. E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[15] T. M. Cover and Joy A. Thomas. *Elements of information theory.* 2nd ed. Hoboken: Wiley-Interscience, 2006. ISBN: 978-0471241959.

[16] D. Cohen. *Precalculus: A Problems-Oriented Approach.* Cengage Learning, 2004. ISBN: 9781111793685. URL: https://books.google.cz/books?id=JSI8AAAAQBAJ.

[17] Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. *Mathematics for machine learning*. Cambridge: Cambridge University Press, 2020. ISBN: 978-110-8455-145.

[18] Divya Somvanshi and R.D.S. Yadava. "Boosting Principal Component Analysis by Genetic Algorithm". In: *Defence Science Journal* 4.60 (2010), p. 7. DOI: 10.1.1.902.7675. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.902.7675%5C&rep=rep1%5C&type=pdf (visited on 04/12/2017).

[19] Wilson Rivera-Gallego. "A GENETIC ALGORITHM FOR SOLVING THE EUCLIDEAN DISTANCE MATRICES COMPLETION PROBLEM". In: *SAC* (1998), p. 5. URL: http://slapper.apam.columbia.edu/bib/papers/river_b_99.pdf (visited on 04/12/2017).

[20] Dipankar Dasgupta. "Advances in artificial immune systems". In: *IEEE Computational Intelligence Magazine* 1.4 (2006), pp. 40–49. ISSN: 1556-603X. DOI: 10.1109/MCI.2006.329705. URL: http://ieeexplore.ieee.org/document/4129847/ (visited on 06/24/2019).

[21] Vaclav Oujezsky, Vladislav Skorpil, and Tomas Horvath. "Gpon frame analysis with artificial immune system". In: *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE. 2019, pp. 1–4.

[22] Huawei. *6G: The next horizon white paper*. URL: https://www.huawei.com/en/technology-insights/future-technologies/6g-white-paper (visited on 11/27/2022).

[23] Balakrishnan Chandrasekaran. "Survey of Network Traffic Models". In: (), p. 8. URL: http://www.cse.wustl.edu/~jain/cse567-06/ftp/traffic_models3.pdf (visited on 04/12/2017).

[27] Vladislav Skorpil, Vaclav Oujezsky, and Ludek Palenik. "Internet of things security overview and practical demonstration". In: *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE. 2018, pp. 1–7.

[28] Vaclav Oujezsky et al. "Fpga network card and system for gpon frames analysis at optical layer". In: *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*. IEEE. 2019, pp. 19–23.

[29] DFC Design. *DFC we MAKE Electronics...* URL: https://www.dfcdesign.cz/cz.

[30] *Gigabit-capable passive optical networks (G-PON): Transmission convergence layer specification.* 2014. URL: https://www.itu.int/rec/T-REC-G.984.3 (visited on 10/02/2021).

[31] *G.987.1 : 10-Gigabit-capable passive optical networks (XG-PON): General requirements.* 1st ed. Geneva, Switzerland: ITU-T, 2016.

[32] V Oujezsky and T Horvath. "Aequor Tracer–Network Analysis Application". In: *2019 27th Telecommunications Forum (TELFOR).* IEEE. 2019, pp. 1–4.

[33] Vaclav Oujezsky, Tomas Horvath, and Petr Munster. "Application for Determining whether IP Addresses belong to a Map by Coordinates". In: *2019 42nd International Conference on Telecommunications and Signal Processing (TSP).* IEEE. 2019, pp. 14–18.

[35] Kilian Weinberger et al. *Feature Hashing for Large Scale Multitask Learning.* 2009. DOI: 10.48550/ARXIV.0902.2206. URL: https://arxiv.org/abs/0902.2206.

[36] V Skorpil et al. "Parallel processing of genetic algorithms in Python language". In: *2019 PhotonIcs & Electromagnetics Research Symposium-Spring (PIERS-Spring).* IEEE. 2019, pp. 3727–3731.

[37] Vladislav Skorpil and Vaclav Oujezsky. "Parallel Genetic Algorithms' Implementation Using a Scalable Concurrent Operation in Python". In: *Sensors* 22.6 (2022), p. 2389.

[38] *Scikit – Clustering.* 2022. URL: https://scikit-learn.org/stable/modules/clustering.html#clustering.

[39] Dongkuan Xu and Yingjie Tian. "A Comprehensive Survey of Clustering Algorithms". In: *Annals of Data Science* 2.2 (June 2015), pp. 165–193. ISSN: 2198-5812. DOI: 10.1007/s40745-015-0040-1. URL: https://doi.org/10.1007/s40745-015-0040-1.

[40] Simon Haykin. *Neural networks. a comprehesive foundation.* 2nd ed. Upper Saddle River, Prentice-Hall, 1999. ISBN: 978-0132733502.

[41] David L. Davies and Donald W. Bouldin. "A Cluster Separation Measure". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1.2 (1979), pp. 224–227. DOI: 10.1109/TPAMI.1979.4766909.

[42] Pavel Novak and Vaclav Oujezsky. “Detection of Malicious Network Traffic Behavior Using JA3 Fingerprints”. In: *Proceedings II of the 28th Conference STUDENT EEICT 2022.* Ed. by Assoc. Prof. Vítězslav Novák. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, 2022, pp. 194–197. ISBN: 978-80-214-6030-0. URL: `https://www.eeict.cz/eeict_download/archiv/sborniky/EEICT_2022_sbornik_2_v2.pdf`.

[43] Vladimir I. Levenshtein. “Binary codes capable of correcting deletions, insertions, and reversals”. In: *Soviet physics. Doklady* 10 (1965), pp. 707–710.

[44] Pavel Novák. “Detection of malicious network traffic behavior”. Master’s thesis. Brno: Masaryk University, Faculty of Informatics, 2022. URL: `https://is.muni.cz/th/dq2t1/`.

[45] Robert A. Wagner and Roy Lowrance. “An Extension of the String-to-String Correction Problem”. In: *J. ACM* 22.2 (Apr. 1975), pp. 177–183. ISSN: 0004-5411. DOI: `10.1145/321879.321880`. URL: `https://doi.org/10.1145/321879.321880`.

[46] Lawrence Hubert and Phipps Arabie. “Comparing partitions”. In: *Journal of Classification* 2.1 (Dec. 1985), pp. 193–218. ISSN: 1432-1343. DOI: `10.1007/BF01908075`. URL: `https://doi.org/10.1007/BF01908075`.

[47] R.O. Sinnott, H. Duan, and Y. Sun. “Chapter 15 - A Case Study in Big Data Analytics: Exploring Twitter Sentiment Analysis and the Weather”. In: *Big Data.* Ed. by Rajkumar Buyya, Rodrigo N. Calheiros, and Amir Vahid Dastjerdi. Morgan Kaufmann, 2016, pp. 357–388. ISBN: 978-0-12-805394-2. DOI: `https://doi.org/10.1016/B978-0-12-805394-2.00015-5`. URL: `https://www.sciencedirect.com/science/article/pii/B9780128053942000155`.

[48] Václav Oujezský. “Converged Networks and Traffic Tomography by Using Evolutionary Algorithms”. Disertation Thesis. Brno: Brno University of Technology. Faculty of Electrical Engineering and Communication. Department of Telecommunications, 2017. URL: `http://hdl.handle.net/11012/68296` (visited on 11/27/2022).

[49] Vladislav Skorpil, Vaclav Oujezsky, and Martin Tuleja. “Testing of Python models of parallelized genetic algorithms”. In: *2020 43rd International Conference on Telecommunications and Signal Processing (TSP).* IEEE. 2020, pp. 235–238.

[50] A. Muhammad, Andrzej Bargiela, and Graham King. “Fine-grained parallel genetic algorithm: A global convergence criterion”. In: *International Journal of Computer Mathematics - IJCM* 73 (Jan. 1999), pp. 139–155. DOI: 10.1080/00207169908804885.

[51] Enrique Alba and José M. Troya. “A survey of parallel distributed genetic algorithms”. In: *Complexity* 4.4 (1999), pp. 31–52. DOI: 10.1002/(SICI)1099-0526(199903/04)4:4<31::AID-CPLX5>3.0.CO;2-4.

[52] Sven E. Eklund. “A massively parallel architecture for distributed genetic algorithms”. In: *Parallel Computing* 30.5 (2004). Parallel and nature-inspired computational paradigms and applications, pp. 647–676. ISSN: 0167-8191. DOI: https://doi.org/10.1016/j.parco.2003.12.009. URL: https://www.sciencedirect.com/science/article/pii/S0167819104000365.

[53] Vaclav Oujezsky, Tomas Horvath, and Vladislav Skorpil. “Botnet C&C traffic and flow lifespans using survival analysis”. In: *International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems* 6.1 (2017), pp. 38–44.

[54] Vaclav Oujezsky and Tomas Horvath. “Traffic analysis using netflow and python”. In: *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska* 7.2 (2017), pp. 5–7.

[55] Vaclav Oujezsky and Tomas Horvath. “Traffic similarity observation using a genetic algorithm and clustering”. In: *Technologies* 6.4 (2018), p. 103.

[56] Rod Pierce. *Area of Irregular Polygons.* URL: https://www.mathsisfun.com/geometry/area-irregular-polygons.html (visited on 11/27/2022).

[57] Frank L. D evai. “On the Complexity of Some Geometric Intersection Problems”. In: 1995.

[58] John McHugh, Ron McLeod, and Vagishwari Nagaonkar. “Passive network forensics: behavioural classification of network hosts based on connection patterns”. In: *Operating Systems Review* 42 (2008), pp. 99–111.

[59] Sérgio S.C. Silva et al. “Botnets: A survey”. In: *Computer Networks* 57.2 (2013). Botnet Activity: Analysis, Detection and Shutdown, pp. 378–403. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2012.07.021. URL: https://www.sciencedirect.com/science/article/pii/S1389128612003568.

[60] Amine Boukhtouta et al. “Graph-theoretic characterization of cyber-threat infrastructures”. In: *Digit. Investig.* 14 Supplement 1 (2015), S3–S15.

[61] Willa K. Ehrlich et al. "Detection of Spam Hosts and Spam Bots Using Network Flow Traffic Modeling". In: *LEET*. 2010.

[62] Sebastián García, Vojtěch Uhlíř, and Martin Rehak. "Identifying and Modeling Botnet C&C Behaviors". In: *Proceedings of the 1st International Workshop on Agents and CyberSecurity*. ACySE '14. Paris, France: Association for Computing Machinery, 2014. ISBN: 9781450327282. DOI: `10.1145/2602945.2602949`. URL: `https://doi.org/10.1145/2602945.2602949`.

[63] Cameron Davidson-Pilon. "lifelines: survival analysis in Python". In: *Journal of Open Source Software* 4.40 (2019), p. 1317. DOI: `10.21105/joss.01317`. URL: `https://doi.org/10.21105/joss.01317`.

[64] Eric Vittinghoff. *Regression methods in biostatistics. linear, logistic, survival, and repeated measures models*. 2nd ed. New York: Springer, 2012. ISBN: 978-1-4614-1352-3.

[65] Pietro S. Oliveto, Jun He, and Xin Yao. "Time complexity of evolutionary algorithms for combinatorial optimization: A decade of results". In: *International Journal of Automation and Computing* 4.3 (June 2007), pp. 281–293. ISSN: 1751-8520. DOI: `10.1007/s11633-007-0281-3`. URL: `https://doi.org/10.1007/s11633-007-0281-3`.

[66] IBM. *Machine Learning*. URL: `https://www.ibm.com/cloud/learn/machine-learning` (visited on 11/27/2022).

[67] Microsoft. *An introduction to the mathematics and logic behind machine learning*. URL: `https://azure.microsoft.com/cs-cz/resources/cloud-computing-dictionary/what-are-machine-learning-algorithms` (visited on 11/27/2022).

[68] Tomas Horvath et al. "Activation Process of ONU in EPON/GPON Networks". In: *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*. IEEE. 2018, pp. 1–5.

[69] Tomas Horvath et al. "Activation Process of ONU in EPON/GPON/XG-PON/NG-PON2 Networks". In: *Applied Sciences* 8.10 (2018). ISSN: 2076-3417. DOI: `10.3390/app8101934`. URL: `https://www.mdpi.com/2076-3417/8/10/1934`.

[71] Vaclav Oujezsky et al. "Gpon traffic analysis with tensorflow". In: *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*. IEEE. 2020, pp. 69–72.

[72] Michal Jurcik et al. “GPON parser for database analysis”. In: *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*. IEEE. 2019, pp. 347–350.

[73] Martin Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2022. URL: `https://www.tensorflow.org/`.

[74] François Chollet et al. *Keras*. `https://keras.io`. 2022.

[75] Leandro Nunes de Castro and Von Zuben. “Artificial Immune Systems: Part I-Basic Theory and Applications”. In: 1999.

[76] Jing Zhang. “Artificial immune algorithm to function optimization problems”. In: *2011 IEEE 3rd International Conference on Communication Software and Networks*. 2011, pp. 667–670. DOI: `10.1109/ICCSN.2011.6014177`.

[77] Tao Liu et al. “A New Clustering Algorithm Based on Artificial Immune System”. In: *2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*. Vol. 2. 2008, pp. 347–351. DOI: `10.1109/FSKD.2008.67`.

[78] Martin Holik et al. “Storage for Traffic from xPON Networks”. In: *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*. IEEE. 2020, pp. 77–80.

[79] *The NIS 2 Directive*. 2022. URL: `https://www.nis-2-directive.com/` (visited on 11/19/2022).

[80] David Gutierrez, Jinwoo Cho, and Leonid G. Kazovsky. “TDM-PON Security Issues: Upstream Encryption is Needed”. In: *OFC/NFOEC 2007 - 2007 Conference on Optical Fiber Communication and the National Fiber Optic Engineers Conference*. 2007, pp. 1–3. DOI: `10.1109/OFC.2007.4348474`.

[81] *Exploit Database - Exploits for Penetration Testers, Researchers, and Ethical Hackers*. OffSec Services Limited. 2022. URL: `https://www.exploit-db.com/` (visited on 03/26/2022).

[82] Vaclav Oujezsky et al. “Security testing of active optical network devices”. In: *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*. IEEE. 2019, pp. 9–13.

[83] Onur Catakoglu, Marco Balduzzi, and Davide Balzarotti. “Automatic Extraction of Indicators of Compromise for Web Applications”. In: *Proceedings of the 25th International Conference on World Wide Web*. WWW ’16. Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, 2016, pp. 333–343. ISBN: 9781450341431. DOI: `10.1145/2872427.2883056`. URL: `https://doi.org/10.1145/2872427.2883056`.

[84] Eduardo B Fernandez. *Security patterns in practice. designing secure architectures using software patterns*. United Kingdom: John Wiley & Sons, Ltd., 2013. ISBN: 978-1-119-99894-5.

[85] Paras Varshney. *Q-Q Plots Explained*. Towards Data Science, Medium. 2022. URL: https://towardsdatascience.com/q-q-plots-explained-5aa8495426c0 (visited on 11/20/2022).

Bibliography – cited RFCs and ITUs

[7] M. Bjorklund. *The YANG 1.1 Data Modeling Language.* RFC 7950. RFC Editor, Aug. 2016.

[8] H. Song et al. *Network Telemetry Framework.* RFC 9232. RFC Editor, May 2022.

[10] F. Brockners, S. Bhandari, and T. Mizrahi. *Data Fields for In Situ Operations, Administration, and Maintenance (IOAM).* RFC 9197. RFC Editor, May 2022.

[11] Robert Braden. *Requirements for Internet Hosts - Communication Layers.* STD 3. `http://www.rfc-editor.org/rfc/rfc1122.txt`. RFC Editor, Oct. 1989. URL: `http://www.rfc-editor.org/rfc/rfc1122.txt`.

[12] J. Postel. *Internet Control Message Protocol.* STD 5. `http://www.rfc-editor.org/rfc/rfc792.txt`. RFC Editor, Sept. 1981. URL: `http://www.rfc-editor.org/rfc/rfc792.txt`.

[24] B. Claise, B. Trammell, and P. Aitken. *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information.* STD 77. RFC Editor, Sept. 2013. URL: `http://www.rfc-editor.org/rfc/rfc7011.txt`.

[25] B. Claise. *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information.* RFC 5101. RFC Editor, Jan. 2008. URL: `http://www.rfc-editor.org/rfc/rfc5101.txt`.

[26] P. Phaal, S. Panchen, and N. McKee. *InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks.* RFC 3176. RFC Editor, Sept. 2001. URL: `http://www.rfc-editor.org/rfc/rfc3176.txt`.

[34] H. Butler et al. *The GeoJSON Format.* RFC 7946. RFC Editor, Aug. 2016.

[70] International Telecommunication Union (ITU). *Recommendation ITU-T G.984.3 Gigabit-Capable Passive Optical Networks (G-PON): Transmission Convergence Layer Specification.* Tech. rep. Geneva, 2014. URL: `https://www.itu.int/rec/T-REC-G.984.3-201401-I/en`.

A Appendix – popular NTA techniques

Many different techniques used or proposed for anomaly detection are presented in the research literature. The following is a list of a variety of these techniques. This review is provided without references as the author's notes.

- AutoEncoders for anomaly detection.
- Bayesian networks for network traffic recognition.
- Clustering analysis for outlier detection.
- Graph-based anomaly detection.
- Fuzzy logic for outlier detection.
- Hidden Markov models for anomaly detection.
- Long short-term memory neural networks for time series anomaly detection.
- One Class SVM for the detection of data patterns.
- PCA based network traffic anomaly detection.
- Statistical tests – Z-score, normalized residual test for outliers data detection.
- Tensor-based anomaly (outlier) detection.

Acronyms

5G	5th Generation Mobile Networks
AI	Artificial Intelligence
API	Application Programming Interface
ARI	Hubert-Arabie Adjusted Rand Index
AS	Autonomous System
AIS	Artificial Immune System
CART	Classification and Regression Tree
C&C	Command and Control
CLI	Command Line Interface
CPU	Central Processing Unit
CSV	Comma-Separated Values
DBSCAN	Density-based Spatial Clustering of Applications with Noise
DMA	Direct Memory Access
DMDM	Data Model-Driven Management
DoS	Denial of Service
DPI	Deep Packet Inspection
EMA	Exponential Moving Average
ENISA	European Union Agency for Cybersecurity
ERSPAN	Encapsulated Remote Switch Port Analyzer
ESG	Enterprise Strategy Group
EU	European Union
FPGA	Field-programmable Gate Array
FTTH	Fiber to the Home
GA	Genetic Algorithm
GLM	Generalized Linear Model
GPON	Gigabit Passive Optical Network
gNMI	gRPC Network Management Interface
GPU	Graphics Processing Units
gRPC	Google Remote Procedure Call
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection Systems
IETF	Internet Engineering Task Force
INT	In-band Network Telemetry
IOAM	In Situ Operations, Administration, and Maintenance
IoC	Indicators of Compromise
IoT	Internet of Things

IP	Internet Protocol
IPFIX	IP Flow Information Export
IPI	IP Intelligence
IPS	Intrusion Prevention Systems
ISSN	International Standard Serial Number
ITU-T	International Telecommunication Union – Telecommunication
JSON	JavaScript Object Notation
LSTM	Long Short Term Memory
MV ČR	Ministry of the Interior of the Czech Republic
NaN	Not a Number
NIS	Network and Information Systems
NIST	National Institute of Standards and Technology
NN	Neural Network
NT	Network Throughput
NTA	Network Traffic Analysis
ONU	Optical Network Unit
OLT	Optical Line Terminator
OMCC	ONU Management and Control Channel
OMCI	ONU Management and Control Interface
OPTICS	Ordering Points to Identify the Clustering Structure
OS	Operating System
OSI	Open Systems Interconnection
PCA	Principal Component Analysis
PCAP	Packet Capture
PCIe	Peripheral Component Interconnect Express
PI	Provider Independent
PLOAM	Physical Layer Operation Administration and Maintenance
RFC	Request for Comments
RTT	Round-Trip Time
RIPE	Réseaux IP Européens
SC	Silhouette Coefficient
SDN	Software Defined Network
sFlow	Sample Flow Protocol
SIEM	Security Information and Event Management
SIRAP	Security Incident Response Automation for xPON
SMA	Simple Moving Average
SNMP	Simple Network Management Protocol
SOAR	Security Orchestration, Automation and Response
SPAN	Switch Port Analyzer

SSL	Secure Socket Layer
SQL	Structured Query Language
SVM	Support Vector Machine
TA ČR	Technology Agency of the Czech Republic
TAP	Test Access Point
TCP	Transmission Control Protocol
TLP	Traffic Light Protocol
TOS	Theft of Service
TPU	Tensor Processing Unit
t-SNE	t-distributed Stochastic Neighbor Embedding
UDP	User Datagram Protocol
WAF	Web Application Firewall
WSN	Wireless Sensor Networks
XGPON	10 Gigabit-capable PON
YANG	Yet Another Next Generation

Symbols

$\mathbf{b}$	bit
$\mathbf{B}$	byte
H	Shannon Entropy
s	second
$\mathcal{O}$	Omicron, asymptotic complexity
P	probability of the truth of the predicate formula [propositional forms]
ϕ	empty set
t, T	time value representative
$\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$	fat lowercase letter – vector
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots$	fat uppercase letter – matrix
$\forall$	for all, for any; generic [uppercase] quantifier, generic [uppercase] quantizer, generalizer
$\in$	<is> an element of <set>, belongs to <set> ; incidence [membership] element to the set
$\mathbf{a}^T$	column [transposed] vector
R_{-1}, R^{-1}	inverse relation to relation R
c_k	centroid with membership in *to*
DB	Davies-Bouldin validation index
GHz	10^9 Hz, Hertz, unit of frequency in the SI system
SVD	singular decomposition matrix
lev	Levenshtein distance