UNIVERSITY OF BELGRADE

SCHOOL OF ELECTRICAL ENGINEERING

Juma Ibrahim

AN ARCHITECTURE FOR NETWORK TRAFFIC ANOMALY DETECTION SYSTEM BASED ON ENTROPY ANALYSIS

Doctoral Dissertation

Belgrade, 2022.

UNIVERZITET U BEOGRADU

ELEKTROTEHNIČKI FAKULTET

Juma Ibrahim

ARHITEKTURA SISTEMA ZA PREPOZNAVANJE NEPRAVILNOSTI U MREŽNOM SAOBRAĆAJU ZASNOVANO NA ANALIZI ENTROPIJE

Doktorska disertacija

Beograd, 2022.

Podaci o mentoru i članovima komisije

Mentor:
dr Slavko Gajin, vanredni profesor,
Univerzitet u Beogradu - Elektrotehnički fakultet

Članovi komisije:

dr Milo Tomašević, redovni profesor,
Univerzitet u Beogradu - Elektrotehnički fakultet

dr Miroslav Marić, redovni profesor,
Univerzitet u Beogradu - Matematički fakultet

Datum usmene odbrane: _____

ACKNOWLEDGMENT

This research came late to my life, although our believe says "ask for science from born to death…" and as a normal life no one can do such work alone without help and encouragement from others.

I am particularly deep thanks and great praise extend to God and then to the soul of my parents, who devote a good part of their lives to making sure I am happy and in the right way, they're the root of who I am.

I would like to express my profound and sincere appreciation to Prof. Dr. Slavko Gajin, my supervisor. With his tremendous expertise of computer security, he generously trained me on how to be a good researcher. He led me in my investigations and directed me in the right direction. Gajin introduced me to the area of network anomaly detection and has been helping me since the days I started working in this field. For the last six years he has been mentoring me and helping me.

My heartfelt thanks go to all RCUB team work, the community was a source of fellowship, as well as successful and collaboration.

I remember with appreciation the sources of funding that made my Ph.D. research possible. I was funded by the Ministry of Education-Libya.

Last but not least; I want to thank my family for all the love and encouragement they offer. Most of all for my caring, compassionate, motivating and patient wife.

Belgrade,
01.06. 2022.

DEDICATION

I dedicate the dissertation to the souls of my mother and father,
to all my family members,
with many thanks
on help support and patience

ABSTRACT

With the steady increase in reliance on computer networks in all aspects of life, computers and other connected devices have become more vulnerable to attacks, which exposes them to many major threats, especially in recent years. There are different systems to protect networks from these threats such as firewalls, antivirus programs, and data encryption, but it is still hard to provide complete protection for networks and their systems from the attacks, which are increasingly sophisticated with time. That is why it is required to use intrusion detection systems (IDS) on a large scale to be the second line of defense for computer and network systems along with other network security techniques. The main objective of intrusion detection systems is used to monitor network traffic and detect internal and external attacks.

Intrusion detection systems represent an important focus of studies today, because most protection systems, no matter how good they are, can fail due to the emergence of new (unknown/predefined) types of intrusions. Most of the existing techniques detect network intrusions by collecting information about known types of attacks, so-called signature-based IDS, using them to recognize any attempt of attack on data or resources. The major problem of this approach is its inability to detect previously unknown attacks, even if these attacks are derived slightly from the known ones (the so-called zero-day attack). Also, it is powerless to detect encryption-related attacks. On the other hand, detecting abnormalities concerning conventional behavior (anomaly-based IDS) exceeds the abovementioned limitations. Many scientific studies have tended to build modern and smart systems to detect both known and unknown intrusions. In this research, an architecture that applies a new technique for IDS using an anomaly-based detection method based on entropy is introduced.

Network behavior analysis relies on the profiling of legitimate network behavior in order to efficiently detect anomalous traffic deviations that indicate security threats. Entropy-based detection techniques are attractive due to their simplicity and applicability in real-time network traffic, with no need to train the system with labelled data. Besides the fact that the NetFlow protocol provides only a basic set of information about network communications, it is very beneficial for identifying zero-day attacks and suspicious behavior in traffic structure. Nevertheless, the challenge associated with limited NetFlow information combined with the simplicity of the entropy-based approach is providing an efficient and sensitive mechanism to detect a wide range of anomalies, including those of small intensity.

However, a recent study found of generic entropy-based anomaly detection reports its vulnerability to deceit by introducing spoofed data to mask the abnormality. Furthermore, the majority of approaches for further classification of anomalies rely on machine learning, which brings additional complexity.

Previously highlighted shortcomings and limitations of these approaches open up a space for the exploration of new techniques and methodologies for the detection of anomalies in network traffic in order to isolate security threats, which will be the main subject of the research in this thesis.

This research addresses all these issues by providing a systematic methodology with the main novelty in anomaly detection and classification based on the entropy of flow count and behavior features extracted from the basic data obtained by the NetFlow protocol.

Two new approaches are proposed to solve these concerns. Firstly, an effective protection mechanism against entropy deception derived from the study of changes in several entropy types, such as Shannon, Rényi, and Tsallis entropies, as well as the measurement of the number of distinct elements in a feature distribution as a new detection metric. The suggested method improves the reliability of entropy approaches.

Secondly, an anomaly classification technique was introduced to the existing entropy-based anomaly detection system. Entropy-based anomaly classification methods were presented and effectively confirmed by tests based on a multivariate analysis of the entropy changes of several features as well as aggregation by complicated feature combinations.

Through an analysis of the most prominent security attacks, generalized network traffic behavior models were developed to describe various communication patterns. Based on a multivariate analysis of the entropy changes by anomalies in each of the modelled classes, anomaly classification rules were proposed and verified through the experiments. The concept of the behavior features is generalized, while the proposed data partitioning provides greater efficiency in real-time anomaly detection. The practicality of the proposed architecture for the implementation of effective anomaly detection and classification system in a general real-world network environment is demonstrated using experimental data.

Keywords: Anomaly detection, Anomaly classification, Entropy, Entropy deception, Network behavior analysis.

Scientific field: Electrical and Computer Engineering

Research area: Computer Engineering and Informatics

UDC number:

SAŽETAK

With the steady increase in reliance on computer networks in all aspects of life, computers and other connected devices have become more vulnerable to attacks, which exposes them to many major threats, especially in recent years. There are different systems to protect networks from these threats such as firewalls, antivirus programs, and data encryption, but it is still hard to provide complete protection for networks and their systems from the attacks, which are increasingly sophisticated with time. That is why it is required to use intrusion detection systems (IDS) on a large scale to be the second line of defence for computer and network systems along with other network security techniques. The main objective of intrusion detection systems is used to monitor network traffic and detect internal and external attacks.

Intrusion detection systems represent an important focus of studies today, because most protection systems, no matter how good they are, can fail due to the emergence of new (unknown/predefined) types of intrusions. Most of the existing techniques detect network intrusions by collecting information about known types of attacks, so-called signature-based IDS, using them to recognize any attempt of attack on data or resources. The major problem of this approach is its inability to detect previously unknown attacks, even if these attacks are derived slightly from the known ones (the so-called zero-day attack). Also, it is powerless to detect encryption-related attacks. On the other hand, detecting abnormalities concerning conventional behavior (anomaly-based IDS) exceeds the abovementioned limitations. Many scientific studies have tended to build modern and smart systems to detect both known and unknown intrusions. In this research, an architecture that applies a new technique for IDS using an anomaly-based detection method based on entropy is introduced.

Network behavior analysis relies on the profiling of legitimate network behavior in order to efficiently detect anomalous traffic deviations that indicate security threats. Entropy-based detection techniques are attractive due to their simplicity and applicability in real-time network traffic, with no need to train the system with labelled data. Besides the fact that the NetFlow protocol provides only a basic set of information about network communications, it is very beneficial for identifying zero-day attacks and suspicious behavior in traffic structure. Nevertheless, the challenge associated with limited NetFlow information combined with the simplicity of the entropy-based approach is providing an efficient and sensitive mechanism to detect a wide range of anomalies, including those of small intensity.

However, a recent study found of generic entropy-based anomaly detection reports its vulnerability to deceit by introducing spoofed data to mask the abnormality. Furthermore, the majority of approaches for further classification of anomalies rely on machine learning, which brings additional complexity.

Previously highlighted shortcomings and limitations of these approaches open up a space for the exploration of new techniques and methodologies for the detection of anomalies in network traffic in order to isolate security threats, which will be the main subject of the research in this thesis.

This research addresses all these issues by providing a systematic methodology with the main novelty in anomaly detection and classification based on the entropy of flow count and behavior features extracted from the basic data obtained by the NetFlow protocol.

Two new approaches are proposed to solve these concerns. Firstly, an effective protection mechanism against entropy deception derived from the study of changes in several entropy types, such as Shannon, Rényi, and Tsallis entropies, as well as the measurement of the number of distinct elements in a feature distribution as a new detection metric. The suggested method improves the reliability of entropy approaches.

Secondly, an anomaly classification technique was introduced to the existing entropy-based anomaly detection system. Entropy-based anomaly classification methods were presented and effectively confirmed by tests based on a multivariate analysis of the entropy changes of several features as well as aggregation by complicated feature combinations.

Through an analysis of the most prominent security attacks, generalized network traffic behavior models were developed to describe various communication patterns. Based on a multivariate analysis of the entropy changes by anomalies in each of the modelled classes, anomaly classification rules were proposed and verified through the experiments. The concept of the behavior features is generalized, while the proposed data partitioning provides greater efficiency in real-time anomaly detection. The practicality of the proposed architecture for the implementation of effective anomaly detection and classification system in a general real-world network environment is demonstrated using experimental data.

Keywords: Anomaly detection, Anomaly classification, Entropy, Entropy deception, Network behavior analysis.

Scientific field: Electrical and Computer Engineering

Research area: Computer Engineering and Informatics

UDC number:

LIST OF ABBREVIATIONS

DoS	Denial of Service
DDoS	Distributed Denial of Service
DNS	Domain Name System
EMA	Exponential Moving Average
FN	False Negative
FP	False Positive
FPR	False Positive Rate
FTP	File Transfer Protocol
HIDS	Host based Intrusion Detection System
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IP	Internet Protocol
IPFIX	Internet Protocol Flow Information Export
KDD	Knowledge Discovery and Data Mining
k-NN	k-Nearest Neighbor
ML	Machine Learning
MLP	Multilayer Perceptron
NADS	Network Anomaly Detection System
NB	Naive-Bayes
NIDS	Network Intrusion Detection System
NTP	Network Time Protocol
RF	Random Forest
SVM	Support Vector Machine
SYN	Synchronize
TCP	Transmission Control Protocol
TN	True Negative
TP	True Positive
UDP	User Datagram Protocol
WEKA	Waikato Environment for Knowledge Analysis

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1. INTRODUCTION

1.1 Overview

Internet has fundamentally revolutionized the way we live and work driving the growth of electronic services. The Internet has made our lives, enterprises, and professions easier and more straightforward and we increasingly rely on it. The rapid growth of the Internet and the increasing number users in various ways are shown in Figure 1. 1 the ITU[1] estimates that approximately 4.9 billion people (63% of the world's population) are using the Internet in 2021. This represents an increase of 17% since 2019, with 782 million people estimated to have come online during that period.

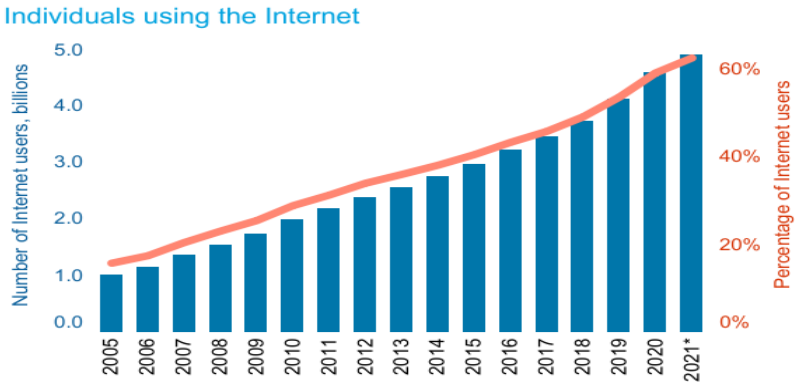

Figure 1. 1 Individuals using the Internet (source: ITU)

In contrast to this incredible advancement in technology, telecommunications, and the facilities and opportunities given by the Internet, there are several challenges and impediments that hinder the use of the broadband service. This increase in the use and reliance on the Internet in all aspects of life corresponds to a direct increase in cyber-crimes and the emergence of new types of cyber-attacks. Figure 1. 2 show the percentage of organizations compromised at least one successful attack[2].

[1] https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx

[2] https://thycotic.com/company/blog/2021/04/27/key-takeaways-2021-cyberthreat-defense-report/

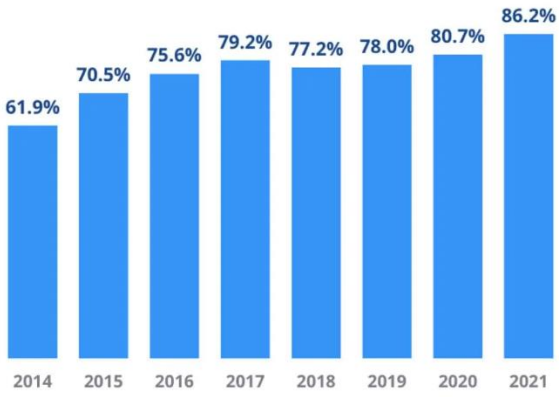

Figure 1. 2 Percentage of organizations compromised at least one successful attack (source: 2021 Cyberthreat Defense Report, CyberEdge Group).

For example, attacked network equipment, such as a router or a switch, may fail, resulting in the loss of connection with these devices as well as other devices such as clients or servers [1]. The network administrators should be prepared to deal with such issues and is ready to solve them or offer alternative solutions. In addition, experts have a tough time predicting and identifying so-called cyber-attacks or hackers who attack network infrastructure and the data that is transferred over the networks in order to disrupt their services. It seems that the struggle between good and evil will continue for the rest of life, since every new day brings a new sort of security attack aimed at breaking into a database and stealing data, in addition to sabotaging or stopping services and websites.

Securing the hosts and network has become a crucial issue, and providing reliable protection of networks has become very important. For network operators, administrators, and end-users, monitoring sudden changes or abnormalities in network usage is a crucial and difficult task. Traditional security tools such as anti-virus programs and firewalls which are considered as the first line of defence are no longer sufficient to protect networks from new sophisticated attacks.

To prevent attacks and safeguard the equipment, other hardware/software security applications must be included. However, in a complex system, achieving complete protection, as well as managing and maintaining such devices and procedures is a difficult task [2]. Intrusion Detection System (IDS) is a relatively modern tool which is one of the methods to protect sophisticated networks against threats and challenges facing computers in general.

Denning (1987) suggested that an IDS is useful for detecting, distinguishing, and attacking intruders [3]. Intrusion detection is therefore the method of tracking, detecting, and evaluating symptoms of security threats or changes that happen in a computing device or a network. It is generally difficult to obtain all information about attacks, especially if the hackers are very skilled and able to delete the hack traces.

The role of IDS is to monitor the network, waiting for any suspicious activity that violates the network management policies. These systems usually record the information related to a certain event and then inform the network administrator about these activities to deal with these issues if necessary.

IDS systems aim to recognize intrusions that are misused (misuse of the privilege) which means that the attackers are from inside the organization and have knowledge of the rules and laws followed, as well as they have knowledge of weaknesses. They have a greater chance than the external attackers who try to attack from outside the organization via the Internet, as well as they have some specific powers and aspire to increase this authority to access data and information which they are not supposed to be accessed. IDS cannot protect from intrusions, but the advantage comes with automated detection after an intrusion has occurred. The increasing complexity of modern networks is accompanied by constant changes in the security threat landscape.

The main two approaches of IDS are deterministic and statistical systems. Most commercially available IDS are deterministic systems misuse-based, also known as signature-based or knowledge-based since detection performs by comparing the attack footprint to a knowledge base of existing attacks. If a previously determined pattern has been identified, an alert is triggered. The strength of a misuse-based system is that it is quite accurate, which means that it seldom triggers an alarm due to benign activities. On the other side, the success of the system is based on the signatures being complete. As a result, a misuse-based system is unable to detect cryptographic traffic and zero-day attacks.

In statistical systems, known as anomaly-based or behavior-based, the intrusion detection relay on traffic pattern behavior analysis, which is an increasingly considered feature in current security monitoring and environmental security. They can identify both known and unknown attacks by creating a model of a network or system usual behavior and looking for deviations from that model [4][5].

According to the literature [6][7][8][9][10][11], many specific classification techniques have been applied to the problem of intrusion detection. In recent years, emphasis has turned to data mining and soft computing approaches, with efficient extraction of the patterns of network user's behavior in these methods. Different techniques have been applied to discover useful knowledge which describes the network traffic behaviors from broad audit data sets, such as artificial neural networks, rule-based inductive and associative structures, genetic algorithms, decision trees, Naïve Bayes, fuzzy logic, etc. Clustering and outlier detection methods are also among the commonly used IDS techniques since they can detect both known and unknown patterns of attacks, thereby helping the smart IDS grow.

Many researchers made a progress to optimize the classification accuracy (or overall classification cost) and omit the need to optimize interpretability [12]. Most of IDS attempt to improve accuracy and minimize the false positive rate (FPR), but present a more complex model.

Indeed, the model complexity is a common disadvantage for most of the proposed methods, because providing a balanced IDS that specifically address the trade-off between the ability to detect new forms of attack and produce low false detection levels is a fundamental challenge. The method should be online, gradual, and sensitive to the ever-changing activities of regular users and attackers, in addition to providing robust and responsive IDS, which will both increase efficiency and broaden domain expert knowledge.

Several studies show that there is a significant interest in implementing entropy-based techniques for network behavior analysis and anomaly detection [13]. Their efficiency is often demonstrated by using examples with heavily loaded anomalous traffic, such as intensive botnet or DDoS attacks. For attacks with less intensive traffic, such as SYN Flood, Port Scan, or Dictionary attacks, the volumetric features do not provide sufficient information. Therefore, additional features must be used, such as the degree of communication with other peers, so-called behavior features [14]. In all cases, a detection rate with entropy-based approaches is tightly related to the relative ratio of the amount of anomalous traffic and the amount of regular traffic. Another important fact is that the higher variation in the observed features of regular traffic diminishes the detection efficiency since these normal but frequent changes cannot be easily distinguished from abnormal network activities.

The existing scientific researches use the entropy of various features only for anomaly detection as an indication of attacks, while there is a lack of efficient entropy-based methods used for further attacks classification or providing a mechanism against deceiving the existing entropy-based anomaly detection techniques.

Therefore, the motivation behind our research was to fill the gap in this research problem. It is stareted by conducting a detailed behavior analysis of various types of anomalies caused by security attacks and investigating how they affect the entropy of observed network features. Then, an important objective was to propose a method applicable for practical usage in a general network environment. For that reason, basic flow features are chosen since they can be easily collected from network routers using NetFlow [15] or similar protocols, such as IPFIX, J-Flow, NetStream [16][17][18], etc. Accordingly, the presented research does not focus on specific attacks and particular use cases forcing the efficiency as high as possible by fine-tuning the parameters, but on providing a wider entropy-based methodology and architecture that can be adapted and further improved to apply to any type of real-life network traffic.

1.2 Challenges in Anomaly Detection

Implementing network intrusion detection systems involves several practical chalenges. The massive volume of data that has to be evaluated, with posible high false positive rates, establishing the precise event that caused an alarm, as well as the validation problem, which involves deciding whether a system has to be retrained, the absence of relevant training data are some of the examples of these difficulties.

1.2.1 Huge volume of data

Networking devices transfers a tremendous amount of data every day and processing such a large data volume is a Big Data problem. Deep data analysis, such as deep packet inspection, is almost impossible with such a massive amount. As a result, many researchers focus their efforts on evaluating aggregated network communication data, such as network traffic records. However, the amount of log records generated on daily basis might easily exceed 1 billion, necessitating detection algorithms with minimal computational and space complexity, such as O(n) or O(nlogn) [19].

1.2.2 High cost of false positives

Despite the fact that IDS is a mature technology, it still has a performance issue in terms of the rate of identifying genuine threats while avoiding errors in reporting prospective dangers. False positives errors are a form of error in which the system incorrectly reports an attack. The accuracy of IDS may be increased by reducing false positives and improving actual detection rates. The number of false positives alarms, i.e. regular activity mistakenly identified as anomalous or malicious, is a prevalent problem with all anomaly-based detector. Each false positive wastes security analyst time since they must conduct a thorough investigation of these events. As a result, numerous researchers have been looking for a means to make more precise anomaly detection systems, and several different false positive reduction strategies have been developed. All of these methods aim to find the best balance between false positives errors and attack detection accuracy (true positive rate) [19][20].

1.2.3 Detection vs. Identification

Anomaly detection systems trigger alerts when abnormal behavior is detected. This warning usually includes meta-data about the discovered abnormality. A network administrator can then utilize this information to determine what caused the anomaly detection warning in the first place. The process of determining the root cause of an alert might take up to an hour on average. One of the most significant issues in anomaly detection is the difficulty of determining the underlying cause of an alarm. Knowing the fundamental cause is essential for determining if the warning is a false positive and the steps should be taken to address the problem. Unfortunately, the research community has not paid enough attention to the problem of root-cause analysis so far [5].

1.2.4 Evaluation

Another issue in the anomaly detection field is the lack of a wide range of labelled datasets for system evaluation and training. Since each system is tested under different conditions, this makes comparing the performance of anomaly detection systems extremely difficult.

Many researchers sought to create labelled data for anomaly-based intrusion detection systems training and validation. The most well-known labelled dataset has been widely panned because modern attacks and unusual sorts of behavior were not well covered, rendering the dataset artificial.

Due to the privacy limitations imposed on network traffic, real network traffic should be anonymized, hence several researchers recommended simulating the trace. In a dynamic environment such as a computer network, however, it is difficult to repeat the pattern of abnormal or normal activity. Modelling the behavior of a typical network user is a difficult issue since even a typical network user might move between a variety of regular behaviors [5][19].

1.2.5 Anomaly detectors are heterogeneous

The anomaly detectors behave differently in various situations, and the same anomaly might be scored differently by different anomaly detectors in different contexts. Furthermore, detection performance differs depending on the kind of abnormality and the rest of the traffic. The efficiency is also affected by the detector learning process and the environment. Accuracy varies in tandem with network attributes changes (i.e. network characteristics will be radically different during the day and at night).

The side-effects of the attack might have a detrimental impact on the detector performance. Malicious activity can even alter the learning process that abnormality should be a result of hostile behavior [19].

1.3 Research goals and objectives

The aim of this research is to develop a new methodology for efficient detection of anomalies in network traffic based on individual communication flows in order to identify traffic irregularities as indications of security threats to computer networks, with special emphasis on the possibility of practical application in production computer networks. The practical application of the developed methodology will be reflected in the proposal of a new architecture for an efficient and flexible way for implementation of flow-based anomaly detection solution for real-life use cases, which is primarily based on entropy calculation.

The main challenges are to determine the method for precisely establishing the boundary between normal and anomaly behavior in the network, in order to avoid false alarms and achieve a high level of anomaly detection. It is also important that the proposed architecture and methodology apply to different types and intensities of network traffic, different patterns of behavior of participants in network communications, as well as different types of attacks and anomalies (eg DoS, DDoS, BOTNET, network and device scanning, password guessing, etc.).

The importance of the proposed research is reflected in the fact that it is necessary to constantly develop and improve the system of attack detection in computer networks, in order to effectively respond to the growing number of new and sophisticated attacks and other security threats. Although this approach requires far simpler processing, new problems are opening up, such as:

- Defining criteria for recognizing the deviation of entropy resulting from the attack concerning the variations that occur in normal traffic.
- Detection refers to the intervals of entire epochs, so it is necessary to single out the actual events and participants in the observed attacks.
- Different types of attacks and their modifications cause changes in the entropy values of different parameters, so it is necessary to consider and analyze a large number of these values, in order to correctly identify attacks and other anomalies.

The previously pointed out shortcomings and limitations of the mentioned approaches open up a space for research of new techniques and methodologies for the detection of anomalies in network traffic in order to isolate security threats, which is the main subject of our research.

In this work we provide a proof of the concept validated through the experimental results, with the following objectives:

The main objectives of the research conducted in this thesis were the following:

- To develope an effective protection mechanism against entropy deception analysing characteristics of several entropy types, such as Shannon, Rényi, and Tsallis entropies. The suggested method should improves the reliability of entropy approaches.
- To classify data features according to their importance in detecting the anomalies. This should select the most important features, reducing the complexity and reducing noise caused by irrelevant features.
- Define a precise limits between normal and abnormal behavior in order to avoid a high false positive rate or to minimize the error detection rate.
- To develop a new technique and a concept of anomaly classification based on entropy instead of using machine learning.
- To reduce the processing time and the consumption of computer resources in detection process and root-cause analysis.
- To define flexible system architecture suitable for implementation in real-life network environment.

The significance of the research is shown in the creation and use of a novel strategy and system that has been proven through experimental analysis to be capable of providing a greater degree of security in real-world network service use settings.

1.4 Starting hypotheses

The basic hypotheses from which the research is based are the following:

- Different types of anomalies leave different footprints in entropy of corresponding features that can be used.

- It is possible to generate new features for efficient entropy-based anomaly detection, which are based only on the identificatonal flow atributes instead of using the volumetric features.
  The entropy-based techniques for DDoS attacks detection in many scientific researchers are mostly based the volumetric features only. Our results generally confirm these findings, but only for anomalies with a large amount of total packets and bytes, which stands for DDoS and similar volume-intensive attacks. However, this is not the case with many other security attacks, which use different communication pattern, such as Port Scan, Network Scan or Dictionary attack. For this reason, the focus of our research is primarily oriented toward entropy of non-volumetric features, namely the flow-count and behavior-features, and the ways in which they are triggered by the modelled anomalies.
- By dividing it into smaller classes of traffic according to certain criteria, it is possible to increase the accuracy in detecting anomalies of lower intensity, which are invisible when the same technique is applied at the level of all traffic.
- The entropy-based technique can be improved to be resilient to malicious deception by generating artificial traffic masking the real anomaly.
- It is possible to classify anomalies performing multivariate analysis of the calcualted entropy of additionaly extracted features.
  We propose a multivariate analysis of entropy values, which involves observation and analysis of many features, not only to detect anomaly, but also to identify class of the anomaly as an indication of certain type of security threats. To better investigate the behavior of different anomalies in terms of aggregation keys and the corresponding features, we have analysed normal network behavior and the communication characteristics of the most prominent network security attacks.
- It is possible to define a modular architecture of an attack detection system, which flexibly uses different analysis and detection techniques, enabling efficient real-time detection of anomalies and anomalies in real-life computer networks with different and previously unknown ways of using network services and communications.

1.5 Thesis outline

The thesis is organized as follows:

The first chapter provides an overview of the intrusion detection system as well as the thesis' key aims and findings. It explains a general introduction to Intrusion Detection Systems (IDS), and then explanation of the major problem with most the IDSs which represented in generating large numbers of false alerts, as these false alerts increase with a large number of strange and suspicious traffic, also shows the anomaly IDS challenges by focusing on entropy deception, and how is supervised machine learning methods used for network behavior analysis involve significant limitations. Moreover, this chapter explains the purpose of the thesis by proposing a protection technique against entropy deception and introduce a multivariate classifier for IDS to reduce false alarm alarms. The remainder of the thesis is structured in the following manner.

Chapter 2 presents a survey on the current IDS techniques. First an analysis and surveys in the previous general intrusion detection approaches, then some entropy-based detection methods are analyzed and described in more detail.

Chapter 3 describes the most common and well-known attacks by looking at the behavior of different type of computer and network attacks, also inspecting the vulnerabilities of some protocols, operating system, user passwords and access list weakness.

Chapter 4 explains the concept of IDS in general, IDS components, detection techniques, IDS data source, and IDS relationship with Data Mining, first we introduce the different types of intrusion detection system that are used in the most of the organizations nowadays, including an overview and examples about intruders, detection methods, their data source, and comparison between their detection approach.

Chapter 5 shows the proposed methodology and the architecture for traffic anomaly detection system based on entropy analysis, discusses about the proposed system by describing its structure and focusing on the functions of its layers. Starting with the explanation of our architecture and how flow collection, partitioning, aggregation, entropy calculation and entropy change detection has been used to detect anomaly. Finally, we introduce our concept for protection against entropy deception, also a validation of the concept and ends with our validation of the hypothesis.

Chapter 6 describes the experimental results and evaluates the performance of the proposed architecture using the proposed dataset, in the beginning we introduce the available dataset that are used for testing and evaluation the IDS, which one is suitable, contains up-to-date common attack, and publicly available. Then 16-anomaly models were generated and tested to prove our hypothesis and concept for anomaly detection, entropy protection against deception and multivariate classification.

Chapter 7 presents the main conclusions of the proposed system based on the results of the previous chapter.

2. LITERATURE REVIEW AND RELATED WORK

IDS techniques can be based on data mining, statistical analysis, machine learning, including artificial neural networks, genetic algorithms, fuzzy logic, and other artificial intelligence technologies. All these approaches are based on extracting information from large amounts of data, resulting in a high volume of data transmission and processing, which could be time-consuming and expensive.

The task of selecting attributes from a gathering of data is known as feature selection. The goal of picking features is to reduce the dimensionality of the data collection and the processing time required. Feature selection is essential to improve the IDS's precision and classification performance by picking the best subset of features that defines the input dataset. The ideal subset of features for network IDS is found using a wrapper methodology based on a genetic algorithm (GA) as a research methodology and logistic regression (LR) as a training algorithm [21]. The GA-LR wrapper feature selection is the result of collaboration between a genetic algorithm for feature search and logistic regression as a learning technique. The selection technique is designed to increase classification accuracy while lowering the number of features, the classification procedure entails assessing the created subsets of attributes and comparing them to other current techniques utilizing three decision tree classifiers: C4.5, RF, and NBTree. Using only 18 features from the KDD99 dataset classification accuracy of 99.9 % is achieved, with 99.81 % Detection Rate (DR) and 0.105 % False Acceptance Rate (FAR). Additionally, the selected subset offers a robust DR for the DoS attack, with a 99.98 %. The GA-LR wrapper approach using UNSW-NB15 dataset has both the least FAR of 6.39 % and better recognition accuracy when compared to the other methods.

The authors in [22] present a new IDS with new feature selection and classification algorithms, using a genetic algorithm to pick the most appropriate characteristics, resulting in accurate classification. The J48 classifier has been modified to provide accurate classification. Because of the fewer features, the accuracy improves while the classification time and error rate decrease. The performance benefit of initially classifying a dataset based on the ‘protocol type' function over the conventional method was considering the entire dataset without prior classification.

The authors of [23] do not recommend any techniques or algorithms, but they do develop that splitting the dataset based on protocol type improves efficiency which is detection rate and time to build the model. Their proposed method was evaluated on KDD99 dataset, their method improves efficiency in terms of detection rate and time to construct a model.

The research in [24] investigates whether using the entire training dataset rather than a subset enhances machine learning classifier efficiency. The KDD99 dataset was used to train and evaluate classifiers using the Waikato Environment for Knowledge Analysis (WEKA) Machine Learning Toolbox. Binary evaluation criteria, as well as training time, working memory, and model size were used

to evaluate classifier performance and demonstrate the effects of dataset size. In comparison to previous tests, the findings indicate that classifiers have improved in standard performance metrics. The findings of this review, which evaluated classifiers based on standard binary performance metrics attributes and employed the most frequently used machine learning algorithms on the whole data set, will serve as a paradigm for future research in IDS or other large datasets.

Flow-based intrusion detection and classification system that uses two neural networks for different tasks that rely on aggregated flow statistics of network traffic are proposed by the authors in [25]. Since the metrics can be obtained by network interface hardware or standalone probes, their key advantages are host independence and accessibility on high-speed networks. The results show that using the NetFlow dataset and removing only features that substantially contribute to intrusion detection yields promising results, with one neural network detecting traffic anomalies that might be attacks and the other classifying attacks if they occur. According to noisy data and the curse of dimensionality, a learning approach takes a long time to learn in high-dimensional datasets, and the output continues to suffer. A feature filtering technique to choose a subset of appropriate and non-redundant features to solve these problems is a concern of many researchers. However, most feature selection approaches are inherently unpredictable, in that they pick various subsets of features for different training datasets, resulting in varying classification accuracy.

The authors in [26] presented Ensemble Feature Selection with Mutual Information (EFS-MI), which is a group of algorithms that selects features based on mutual information. These feature selection approaches result in an optimum subset of features by merging subsets of features selected using various filters such as ReliefF, InfoGain, GainRatio, Chi-square and SymmetricUncertainity. They tested the efficiency of the ensemble method on the UCI Machine Learning Repository, and gene expression datasets using several classifiers such as K-Nearest Neighbors (KNN), Random Forests, Decision Trees, and Support Vector Machine (SVM). The overall performance on both of these datasets has been determined to be superb. According to the Average Classification Accuracy (ACA) analysis, the suggested EFS-MI generally solves the local optimum issue of individual filters, especially for large dimensional datasets.

The research in [27] published a comparison of ensemble ML algorithms for unbalanced data sets. The GentleBoost, Bagged tree, LogitBoost, AdaBoost, and RUSBoost algorithms were utilized as a part of the ensemble. The Bagged tree and GentleBoost classifiers exceed RUSBoost, which requires a substantially higher number of training data to reach the same degree of performance. The bagging tree technique is a variance-reduction approach and because trees in practice have a larger variance, it should yield great results in most cases, while the RUSBoost algorithm would have the strength of being easy to compute. But on the other hand it seems to have the least efficient results within that testing setting.

The authors in [28] compared many machine learning classifiers, including the Sequential Minimal Optimization (SMO) and the C-style soft margin Support Vector Machine (C-SVM) algorithms,

as well as ensemble algorithms like LADTree, REPTree, RF, and MultiBoost. Weka was used to simulate their study and the UNSWNB15 as a dataset, demonstrating the RF algorithm's powerful classification abilities, whereas the REPTree algorithm stands as an appropriate suggestion in time-constrained situations. Both SVM algorithms failed to deliver fast results, while SMO may be preferable in situations where timelines are not critical, despite its speed in classification, MultiBoost is the weakest ensemble group's algorithm for this number of behavior features.

The research in [29] describes a supervised filter-based feature selection methodology. Flexible Mutual Information Feature Selection (FMIFS) is a step forward from Mutual Information Feature Selection (MIFS) and Modified Mutual Information Feature Selection (MMIFS). To minimize redundancy, FMIFS proposes a change to Battiti's algorithm. IDS is constructed using FMIFS and the Least Square Support Vector Machine (LSSVM) process. LSSVM is a least-square version of SVM which resolves a numerical solution for classification task rather than another quadratic equation and employs control problem in the concept instead of inequality. Evaluated on three well-known intrusion detection datasets: KDD99, NSL-KDD, and Kyoto 2006+, the proposed LSSVMIDS + FMIFS approach achieves a higher classification accuracy, recognition rate, false alarm rate, and F-measure compared to the existing detection approaches. The recommended detection approach looks to be useful for identifying computer system hazards based on preliminary findings from all datasets.

The authors in [30] proposed a method for detecting anomalous flows based on traffic characteristic distribution. There are three steps to the proposed solution. First, a multidimensional traffic characteristics entropy matrix was built from the viewpoint of the entire network; anomaly-relevant properties of original-destination (OD) flows and irregular flows were assessed and analyzed. By evaluating variations in the distribution of anomalous behaviors and calculating the unusual flow specific distribution on the OD flow, a potential collection of abnormal flows was produced in the second stage, finally, after additional filtering and selection, irregular flows were identified using the association rule mining based on connection matrix. The approach is accurate and outperforms current methods, according to both simulation and real-world data analysis measurements.

According to traditional attacking phases against a network systems include reconnaissance, scanning, gaining access, maintaining access, and covering tracks. Port scans are commonly used by attackers in the early phases of scanning to obtain information about their targets. As a result, detecting port scans will serve as an early warning sign of impending attacks. Owing to the vast volume of network traffic in business networks, however, detecting slow port scans is difficult. Regarding flow-based network data, two methods for identifying slow-port scans have been proposed in [31], both approaches rely on state-of-the-art pre-processing technology. Unsupervised Port Scan Detection (UPSD) and Supervised Port Scan Detection (SPSD) are two different approaches to port scan detection. SPSD uses classification methods, whereas UPSD uses sequential hypotheses checking. Owing to the translation of flows into network events with all methods, the volume of data is reduced, and the amount of research

effort for security experts is reduced. Testing the methods using the CIDDS-001 dataset, both techniques achieve a low false alarm rate and can detect slow port scans.

The author in [32] presented anomaly detection engines dependent on k-nearest neighbors (K-NN) and K-Means Clustering (KMC) techniques using information theory methods like entropy and mutual information. The traits were graded in terms of how important they are for detecting attack types like DOS, R2L, U2R, and PROBE, with this ranking reducing computing complexity by selecting the most relevant network connection features. Then, based on feature selection, k-NN, and k-means clustering, certain anomalous intrusion detection models were presented. These techniques confirmed their efficacy with a detection rate of more than 92% outperforming previous techniques. A high detection rate is achieved because of using the KDD99 dataset, which is overdated compared to the new attacks. In addition, the author in [33] employed information theory metrics like entropy and mutual information, they designed an intelligent IDS based on genetic algorithm and feature selection. First, they ranked the connection features according to their importance, then the network traffic linear classifiers were designed. These models were trained and evaluated using KDD99 data sets, and the testing results indicated a detection increase of approximately 92.94 %. This engine may be utilized in real-time mode.

Entropy has been a subject of great interest to researchers, which makes it a hot research topic with extensive studies. There are many articles, surveys, books, and journals contributing to this broad topic, resulting in a great number of researches regarding entropy. Due to the relative simplicity and application in real networks, entropy-based anomaly detection attracts great interest in the research community [34][35][36], along with more complex methods such as classification, clustering, deep learning, or statistical-based approaches [37]. Entropy means a measure of uncertainty in data distribution, where unusual changes in network behavior metrics can be detected by an abrupt entropy change. It often relies on the NetFlow data and its feature distributions, based on data taken from datasets for research purposes, or on data collected from real networks in practical implementation [38][39]. A classical approach leverages the well-known Shannon entropy in the context of the information theory [40].

Feature selection and aggregation are used to generate distributions of all distinct elements and their aggregated metrics [41]. The IP address and port distributions time series of entropy values are closely associated with each other and have a very similar ability to detect anomalies. The behavioral and flow sizes are less coupled, allowing them to recognize occurrences of anomalies that do not appear in the port and address distributions.

The research in [42] used the C4.5 classification tree approach in the WEKA data mining analysis platform to pick features from intrusion detection data. They selected the features needed to classify the four attack types named in the data using Shannon, Rényi, and Tsallis entropy from a selected group of the well-recognized KDD99. They added the algorithms for Rényi and Tsallis to WEKA which already contains the Shannon entropy.

The use of entropy in statistical methods for feature selection to detect network intrusions was used in this study [43], where the findings can be easily generalized for designing guidelines for IDS using Quinlan's C4.5 decision tree. This study started by replicating the work in [42] and demonstrated the importance of various labelling approaches in offering extra results for the deployment of IDSs. Rényi and Tsallis were applied to WEKA, and the findings were repeated to create a baseline that validated the entropy measurements produced for the records and incorporated into the data-mining method. For accurate classification numbers, the results were in fair alignment with [42], but the number of features and basic features differed. Using entropy in feature extraction was expanded in this study which included Approximate and Sample entropies, which are commonly used in time-series statistics. With the KDD99 dataset, analysis using these two entropies yielded new findings.

To demonstrate that an entropy-based technique is suited for detecting recent botnet-like infections depending on abnormal patterns in networks, an entropy-based network anomaly intrusion detection technique is provided [44]. They analyzed realistic, synthetically generated botnet traffic injected into real flow data and concluded that the parametrized Tsallis and Rényi entropy outperform the Shannon entropy mostly in terms of better detection of peaks or tails in the feature distributions, depending on the used parameter. They also confirmed the poor performance of volume-based approaches.

Out of a wide variety of entropy measures, only Shannon, parameterized Renyi, and Tsallis entropies have been utilized to detect network anomalies [45]. Using parameterized entropy network anomaly detection and supervised learning is presented they are capable of detecting a wide range of abnormalities with a low percentage of false positives. Furthermore, they offer additional information indicating the anomaly type. The findings show that their method outperforms Shannon-based and volume-based approaches; for the context of this study, they generated a data set containing classified flows and anomalous behavior they designated software in Python programming language. However, this raises the question of whether Shannon-based and volume-based methods are better when it comes to anomaly detection.

A group of researchers presented a thorough conceptual research study relating to entropy-based IP traffic anomaly detection in [46]. First, some hypotheses about the subject were provided, followed by the findings of the case study, which included multiple entropy variations and a variety of feature distributions. The results showed that the Tsallis and Renyi entropies performed better, whereas the Shannon entropy performed badly, failing to identify tiny or medium-sized attacks, this makes these results similar to the results in [44][45]. In addition to that, they concluded that the broader the spectrum of features is, the better it is at detecting different types of anomalies, adding that flow durations, addresses, and ports are the best to use when it comes to a large set of network traffic feature distribution.

Outbound denial-of-service attacks in edge networks were investigated using entropy-based analysis by authors in [47] to examine attackers near to the source of the attack and in the beginning stages, the CUmulative SUM control chart (CUSUM) technique is utilized for change-point detection. The entropy-based strategy was compared against an upgraded version of an existing technique, namely CUSUM-based checking of the number of SYN packets, using the ns2 simulator. The findings show that even though an entropy-based detector cannot meet the efficiency of a methodology tailored to a certain type of attack, it can get close results and it does perform well in general. It is also worth noting that the entropy-based technique performs admirably. To investigate the distribution pattern of warning indicators and identify network breaches, the Shannon entropy and Renyi cross-entropy are applied in [48], proposing an entropy-based technique to identify network intrusion. The results of the experiments using real network data suggest that this technology can rapidly and correctly identify network attacks. They used Snort [49] to monitor 32 C-class subnets on the Xi'an Jiaotong university campus network for two weeks, collecting data from over 4,000 users, and the results demonstrate that the approach can detect up to 96% of attacks with a really low false alarm rate.

To reduce high network activity to a single metric that describes the spread or "chaos" that occurs naturally in the network, a group of authors proposed a solution for short-time predictions and efficient reduction of a single measure for high-dimensional network activity in [50]. They designed an approach sensitive enough to detect changes in network entropy time-series data that are sudden and unexpected. By using the Simple Exponential Smoothing (SES) method to soften time series and perform short-term predictions, the disclosed approach identifies sudden changes in the network entropy time series. On a high-speed network connection, small-scale DDoS attacks may be undetectable, but the technique is a useful tool for the network operator. It is simple to set up, quick to deploy and does not require any training samples. They found that determining the underlying cause of abnormalities is typically a difficult process, and the data set that comes from an ISP's network infrastructure includes 5-days of un-sampled unidirectional data packets, the dataset contains a collection of 260 million flows, on average 36,000 flows were collected per minute. Due to the lack of an actual ground-truth dataset, they used custom-made anomalies in the flow trace to test their technique and for that reason, they used to use a customized version of the tool FLAME.

The researchers in [51] introduce the integrated technique to combine an entropy-based model with an anomaly-based system that provides multistage Distributed Denial Of Service (DDoS). The method functions efficiently in regard to degree distribution, calculating entropy for each value of a degree to identify the anomalous behavior.

Other researchers provided a specific framework for assessing IDSs based on information theory [52], which may be used to combine the study of anomaly-based and signature-based IDSs. The technique not only provides a clear image of IDSs based on information theory, but it also enables static/dynamic fine-tuning of IDSs to gain full efficiency, validate IDS performance and improve IDS design with a satisfactory or finer-grained assurance. The established model provided a strong

mathematical foundation for intrusion detection research and opened the door to the study of a variety of aspects of the area.

The authors of [53] provide an organized and thorough summary of the study into entropy-based IDS intending to provide a fast introduction to key areas of the field. The achieved high detection rates prove the effective use of entropy. Flow size, ports, and addresses are some of the many traffic features that have also been offered as options for anomalous intrusion detection based on entropy by a community of scholars in [16][14], there has been less research on recognizing the data models offered by a set of entropy performance measures used when two or more things are utilized together. Then using a month-long traffic trace gathered within a large university network, they were able to shed light on such capabilities. Furthermore, both raw data and classification abilities revealed that port and address distributions are substantially related, whereas behavioral measures and traffic volume give recognition abilities that are separate from other patterns. The authors further contributed to a better understanding of anomalous behavior in a real network. They suggested the utilization of bidirectional data flows to avoid the biases arising from unidirectional flow analysis. Then, they analyzed the entropy of volumetric data, flow count, packet size distribution, and host in/out degree of communications with other hosts, and reported a strong correlation of address and port features, emphasizing better detection abilities of behavior features.

Variations in the amount of traffic are known as deviations [54]. The problem is that not all anomalous network events cause a major shift in the amount of traffic. However, Brauckhoff [55] has shown an entropy-based strategy using traffic feature distributions outperforms volume-based techniques with flow sampling. In the earlier, other traffic feature distributions were advocated, such as header-based (addresses, ports, flags), volume-based (host or service-specific proportion of flows, packets, and bytes), and behavior-based (in / out links for specific node) [14] [56]. What function distributions do best, though is unclear. Nychis identified dependencies between addresses and ports in [14] which is based on his pair-wise correlation tests and suggested the use of volume and behavior-related function distributions.

Tellenbach [56], who used Tsallis entropy in his proof-of-concept traffic entropy telescope that could discover a broad range of irregularities, proved that a parameterized entropy-based network anomaly detection approach is successful; conversely, no association between header-based features was stated.

Entropy-based detection has certain drawbacks, especially when dealing with tiny or slow attacks. This is certainly relevant for Shannon's entropy which has also limited visual presentation. Kopylova [57] used Renyi conditional entropy to identify rapidly selected or aggressive worms with good outcomes. In [58], Lakhina et al. used entropy measurements to analyze the real traffic aggregated by origin-destination Points of Presence (PoPs) inside the research networks Internet2 in the US and Geant in Europe. Using additionally injected synthetic flows, they found significant advantages of using

entropy-based features over the traditional volume-based approach. Additionally, they used unsupervised machine learning for the automatic classification of entropy results and anomalous traffic extraction.

A straightforward approach for DDoS attack detection is based on the volumetric feature, either using total byte and packet counts [59][60][61][62][63][64] or using additionally derived features, such as average packets and bytes per flow [65][66]. However, volume-based metrics are insufficient for sophisticated attacks and less intensive anomalies. In flow-based approaches, the flow count is commonly used to express the frequency of the feature used as an aggregation key.

The authors in [67] employed association rule mining on flow features enhanced with the derived time-window and connection-window features that aggregate flows in the last T seconds or the last N flow records in an early effort in the domain of flow-based anomaly identification. The authors in [68] extended Lakhina's work using unidirectional flows and host-level granularity, modelling the behavior of outgoing and incoming traffic. The former was aggregated by the source IP address calculating the entropy of distributions of destination IP addresses, source port and destination port, while the latter was aggregated by the destination IP address calculating the entropy of distributions of source IP addresses, source port, destination port, and destination IP addresses.

In [69], the authors proposed the utilization of parametrized Tsallis entropy to separately capture the regions with high and low activity in the feature distribution. When entropy parameters are changed, the feature distribution is transformed into a two-dimensional data matrix suitable for anomaly detection by image processing techniques.

The authors in [70] used entropy for profiling per-host behavior in internet traffic. Each of the source and destination IP addresses and ports was aggregated and the entropies of the three remaining features gave a three-dimensional entropy space with a total of 27 behavior clusters. It was shown that different anomalies fit into particular clusters with high accuracy.

In [71][72], they demonstrate the notion of machine learning (ML) detection and classification of network anomalies using a collection of modelled scenarios that is further supplemented using model-dependent artificial flows, entropy computation, and ML using various supervised and unsupervised methods. This paradigm is a possible solution based on machine learning methodologies with great precision, minimal false alarms, and minimal memory and CPU usage. These first findings supported the predicted pattern, as the entropy calculation pretreatment improved the ML module findings.

In general, we can conclude that entropy helps the IDS to achieve high detection levels. But, recent research in [73] mentioned that spoofing attacks can affect IDS that use entropy-based detection methodologies, that is before launching a DDoS attack, an attacker can probe the network and determine underlying flow entropy, they could therefore fake attack packets to maintain the desired entropy value

even during an attack. To the best of our knowledge, a proper solution to this challenge has not been proposed yet.

3. WELL-KNOWN CYBER-SECURITY ATTACKS

3.1 Attacks

Since the early 1990s, there has been significant growth in the number of devices connected to the network and the usage of the Internet. In the world of digital communications, cyber-security is a crucial aspect of network operation. It has become one of the most critical computing challenges because of the rise in the number of unauthorized operations and the diversity of attacker behavior.
In general, the following three traits are commonly related with data security [74]:

1. **Confidentiality** - to prevent any intentional unauthorized data disclosure. It is important to note that such a breach can last long undetected and its consequences are difficult to resolve.
2. **Integrity** – to prevent intentional unauthorized data modification or destroying. If it is detected the data can be made safe again, for example, by restoring from backups, but it is not guaranteed that it can be fully restored.
3. **Availability** – to prevent unauthorized usage of computing resources. For examples, an attacker can block resources using DoS/DDoS attack preventing authorized users to use the system.

According to the previous three characteristics, cyber-security protection represents one of the major technological challenges for computers, networks, and organizations on a worldwide scale [75]. In general, cyber-attacks are defined as any type of attack that targets computer information systems, infrastructures, computer networks, or personal computer devices.

3.2 The most prominent types of attacks

To determine the most appropriate option for the mechanism that should be used to discover such types of attacks it is important to understand the attack behavior and how they use the vulnerability in the system or network. This overview of the most well-known attacks demonstrates that attackers have a wide range of choices when it comes to infiltrating and disabling computer systems. Additionally, proactive measures must be in place to protect and secure our networks, by keeping the protection software up-to-date, training the staff, setting a strong password, and defending the system from attacks by implementing the least-privilege IT environment paradigm. In the rest of this section, only network attacks that can produce anomalies in network traffic are listed and explained.

1.Denial-of-service (DoS) and Distributed denial-of-service (DDoS) attacks

A denial-of-service attack overloads system resources, making it impossible to react to service demands. Unlike other attacks that aim to gain or increase access, it does not provide direct benefits to the attacker. A DoS attack can also be used to bring a system down so that another form of attack can be carried out, such as session hijacking.

A distributed denial-of-service (DDoS) attack are similar to DoS attack, but it is launched from a large number of distant host machines infected with malicious applications that the attacker controls. The goal of a DDoS attack is to bring a network or service down, making it unreachable to its regular users. By flooding the targeted victim with enormous amounts of data or requests, attackers accomplish the goal that causes the system to malfunction. In either case, the DoS/DDoS attacks prevent eligible users like employees, account holders, and individuals, from using the services.

Web servers of many organizations, such as enterprises, governments, media firms, trade, and banks, are regularly targeted by DDoS attacks. Although these attacks cannot result in the loss or theft of sensitive data or other resources, they can expense the victim a significant amount of money and time to mitigate the attack.

The most frequent forms of DoS and DDoS attacks are briefly described below.

a) TCP SYN flood attack

With this attack, the attacker's device sends a large number of TCP connection requests to the target victim system, causing the input buffer space to fill up. Once the connection buffer gets full, the target system cannot process legitimate requests, causing the system to break down or become useless.

b) Teardrop attack

With this attack the length and fragmentation offset fields in subsequent Internet Protocol (IP) packets sent to the targeted host are overlapping. Some older versions of the operating systems cannot deal with this, fails to reassemble packets and get crashed.

c) Smurf attack

This attack uses IP spoofing and the ICMP protocol to overwhelm a victim machine with network traffic. Using a spoofed victim’s address as a source, the ICMP echo requests are sent to broadcast IP addresses in this attack technique. As a result, all IPs in the subnet reply, causing the targeted system and the network to become overwhelmed. This is a repeatable procedure that may be automated to create massive quantities of network congestion.

d) Ping of death attack

IP packets larger than maximam allowed 65,535 bytes could crash some old systems with early TCP/IP implementations. In this case, the packets are fragmented, but buffer overflows as well as other failures can occur after the target machine tries to reassemble it.

e) Botnets

Botnets are massive networks of compromised computers controlled by hackers to conduct DDoS attacks. These compromised computers are named bots or "zombie systems" since they are employed to attack target machines, usually congesting their bandwidth and processing capability. It is hard to protect from these DDoS attacks since botnets are dispersed throughout the world.

Since the infected hosts, controlled by a master host, extend across several administrative jurisdictions, detecting a botnet is a challenging task. Identification of the master host is a straightforward mitigation strategy, since the bots are no longer hazardous when the master is isolated. Nonetheless, despite the interest of many researchers, botnet protection is not yet effective, leaving room for research in this field [4][76].

f) DNS Amplification attack

A DNS amplification attack is a common type of distributed denial of service (DDoS) attack that uses publicly available open DNS servers to flood a victim system with DNS reply traffic. They do so by asking for information from the server that outputs large amounts of data and then redirecting that information directly back to the server by spoofing the reply-to address.

Thus, in a DNS amplification attack, the attackers sends many relatively small packets to a publicly accessible DNS server from many different sources in a botnet. Every one of them requests a very extensive response, such as DNS name lookup requests. The DNS server then replies to each of these distributed requests with response packets containing much more data than the initial request packet sent right back to the victim's DNS server. It is extremely common for these types of attacks to occur on open DNS servers. When leveraging a botnet to generate spoofed DNS requests, the target will experience a flood of DNS replies, all coming from UDP source port 53.

g) NTP Amplification attack

NTP amplification is a sort of Distributed Denial of Service (DDoS) attack where the attacker uses publicly available Network Time Protocol (NTP) servers to flood the victim with UDP traffic.

NTP is one of the earliest network protocols, and it is used to synchronize the clocks of Internet-connected equipment. Older versions of NTP include a monitoring function that allows administrators to request a traffic count from a specific NTP server in addition to clock synchronization.

An attacker makes the "get monlist" request to an NTP server frequently while spoofing the source IP address to that of the victim server. A list of the connected hosts, which is usually long enough, is sent by the NTP server to the forged IP address.

This answer is far bigger than the request, significantly increasing the quantity of traffic sent to the target server and, as a result, degrading performance for legitimate users.

2. Drive-by attack

Drive-by download attacks are a common malware distribution method. Hackers look for susceptible websites and place malware in the HTTP or PHP code on one of the pages. This script might either inject malicious on the victim's host or redirect them to a hacker-controlled website. Drive-by downloads can occur when someone visits a website, reads an email, or clicks on a pop-up window. A drive-by attack, unlike so many other forms of cyber-security attacks, does not require a user to take any action to allow it. The victim does not need to click a download link or open a malicious email attachment to become infected. Due to failed or missing updates, a drive-by download can take advantage of a security flaw in a program, operating system, or web browser.

3. Password attack

Stealing passwords is a frequent and successful attack technique since passwords have always been the most often technique for authenticating users. Scanning around the user's desk, "sniffing" the connectivity to collect plaintext passwords, employing social engineering, acquiring access to a user's password, or straight guessing can all be used to allow access to a user's password. The final strategy (straight guessing) can be used in an arbitrary or organized manner:

a) Password brute force guessing attack established by trying a number of passwords in the hopes of discovering one that succeeds is known as guessing. Using logic, test passwords that are related to the username, work title, hobbies, or other similar information.
b) A dictionary attack uses a dictionary of well-known names to get access to a user's computer and network. Copying confidential data including passwords, transforming a dictionary of frequently used passwords with the same one-way transformation function (hash), and analyzing the results is one of the methods used in this attack.

4. Christmas attack

Known as Xmas tree attacks, derived their name from the set of TCP flags that are turned on within a packet where a specially crafted TCP packet is sent to the target device. What is intriguing is how distant devices react when they are attacked by a Christmas tree offering a hint as to what is on the other side of the fence. Different devices respond differently in diverse situations. As a result, this might be an excellent approach to conducting reconnaissance and gathering knowledge on other software applications.

5. Malware attack

Malware attacks are a malicious software that is entered into the target device to secretly affect the system without the user's consent. Spyware, malware, command and control, and other forms of malicious software (malware) are all included in this thorough definition.

Malware attacks differ from other programs in terms of their ability to propagate throughout the network, causing disruption and harm while remaining undiscovered, and continuing with the compromised system. These programs can destroy network services and disable the operation of the affected computers.

Malicious software can spread by attaching itself to legal code, hiding in regular programs, or replicating itself throughout the Internet.

6. Scan

Scan attack is performed by a single host to detect the state of the destination systems, such as open TCP ports. Scans can generate a large number of flows quickly since the intruder can utilize numerous source or destination ports to connect a variety of target addresses. Scans may be classified into three categories:

- Multiple destination hosts are being scanned for a certain port (horizontal scan).
- A single destination host is examined through several ports (vertical scan).
- A combination of the previous two (block scan, diagonal scan).

7. Web attacks

a) SQL injection attack

Often known as SQL attack, it is a sort of cyber-attack in which harmful code is published in order to break into backend databases and share knowledge that is supposed to be concealed. This might involve information about customers, user databases, or vital corporation information.
A successful SQL attack can result in the destruction of whole tables, the publication of unauthorized user lists, and in certain circumstances, administrator access to several of the databases. When estimating the cost of a SQL attack, it must include the loss of consumer confidence if sensitive data such as addresses, banking information, and contact details are taken. Despite the fact that SQL may be used to attack any SQL database, attackers frequently target websites.

b) Cross-site scripting (XSS) attack

Cross-site scripting is a form of cyber-attack in which an attacker transmits harmful scripts to a website that is supposed to be trustworthy. This occurs whenever one of the suspect sources is given permission to embed its code in online applications, which is then mixed with dynamic material and transmitted to the victim's browser.

XSS attacks leverage third-party online resources to run scripts in the victim's browser or scriptable programs. A payload containing malicious JavaScript is injected into a website's database. When a victim requests a webpage from a website, the website transmits the page to the victim's browser along with the attacker's payload contained in the HTML body, which runs the malicious script. It may send the cookie of the victim to the attacker's server. The attacker can then extract the information and use it to take over the session. When XSS is used to exploit further vulnerabilities, one of the most serious consequences is that an attacker may steal cookies, log keystrokes, capture screenshots, locate and gather network information, and remotely control and administer the victim's machine. Figure 3. 1 demonstrate how XSS attack works.

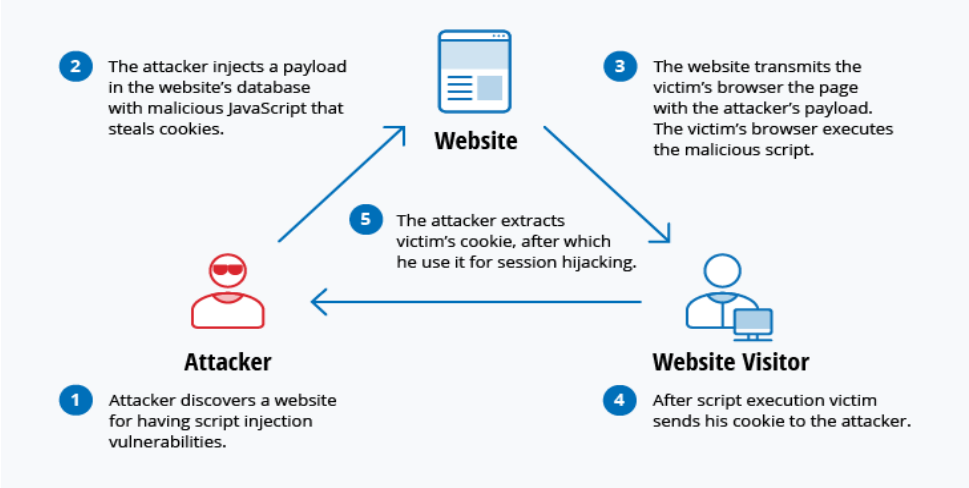

Figure 3. 1 Cross-site scripting [76].

Hackers may use a variety of programming languages to create executable malware programs, including HTML, Flash, Ajax, and Java. However, because of its extensive use on the internet, JavaScript has been the most usually abused. These attacks may be catastrophic, but eliminating the weaknesses that allow them to take place is quite simple.

8. DNS tunnelling

DNS tunnelling is a hard-to-detect attack that sends DNS requests to the attacker's server, giving them a covert command and control channel as well as a conduit for data exfiltration. To sneak data past firewalls, attackers utilize DNS tunnelling that encrypts command and control (C&C) communications

or tiny quantities of data into unnoticeable DNS traffic. Because DNS packets may only have a limited amount of data, any commands must be brief, therefore data exfiltration is sluggish. Because DNS is a loud protocol, it's difficult to tell the difference between a legitimate host query in normal DNS traffic and malicious behavior.

4. INTRUSION DETECTION SYSTEM

Intruders are those that attempt to intrude into the network, primarily interested in gaining access to the system or increasing their privileges that can exploit vulnerabilities. It is often a criminal activity driven by financial gain or any commercial, political or religious agendas.
The intruder, often known as a hacker or cracker, is one of the most well-known security threats [77][78].

Intruder attacks can range from mild to catastrophic. Many individuals, on the more benign end of the spectrum, simply want to browse the internet and discover what's out there. On the other hand, individuals seeking to access privileged data, make illegal changes to data or damage the system are on the more severe end of the spectrum.

4.1 Intrusion Detection System (IDS)

With a variety of cyber-attack techniques and procedures, firewalls which were formerly regarded as the main element of protection in networks are no longer sufficient [1]. As a result, one of the most widely utilized system for network activities monitoring and analysis is the Intrusion Detection System (IDS). The concept of IDS was first introduced by Anderson [79], while over the last two decades, it has gotten a lot of attention. This process of identifying and responding to suspicious activities is a complementary approach to the security of the access control system and encryption operations. IDSs are used to raise alerts when offensive or suspicious activities are set on the systems.

4.2 IDS Accuracy-based performance metrics

The alerts triggered by the IDS when noticing a suspicious activity while the activity is normal is known as positive error, which is a common problem in the IDS. Dealing with alerts by the IDS is not an easy thing, it is like *"boy who cried wolf all the time"*, independent security sensors generate alerts and send them to the section concerned with analyzing alerts, which analyzes the nature of the attack and attempts to reduce the error in alerts. However, false alarms were among the most serious issues that IDSs currently face. False alarms indicate that the IDS have raised concerns about network security, which must be investigated by the security staff. In fact, it turns out that 99% of alerts issued by the hacking system are not related to network security issues [74][80]. The main metrics related to the concept of errors in attack detection are shown in Figure 4. 1

1- **False Negative (FN)**: When there is an attack, an alert is not issued it can be described as a warning of a breach. This is called a negative error, representing the inability of the IDS to detect an attack.
2- **False Positive (FP)**: When there is no attack and a system is issued a breach warning of an attack, which means raising an alarm for normal operation.

3- **True Negative (TN)**: If there is no attack and the system does not generate any notifications.
4- **True Positive (TP)**: When an attack occurs and the IDS generates an alert.

		Predicted	
		+	-
Actual	+	TP	FN
	-	FP	TN

Figure 4. 1 Explanation of two-class problem metrics.

TP and TN are normal and correct activities that IDS performs and are unlike FN and FP errors.

A high FN rate means low IDS performance will put the system at risk of many intrusions, as a result for successful IDS the rate of FP and FN should be as low as possible, and the TN rate and TP rate are the most significant [81].

Based on these four main metrics additional performance indicators are defined as follows [82]:

1. Percent correct or accuracy is the portion of the test examples that the model predicts correctly: (*TP* + *TN*) / (*TP* + *FN* + *FP* + *TN*).
2. Error rate is the portion of the examples in the test set that the model predicts incorrectly: (*FN* + *FP*) / (*TP* + *FN* + *FP* + *TN*).
3. Precision is the portion of the test examples predicted as positive that were really positive: *TP* / (*TP* + *FP*).
4. True-positive rate (TPR) or recall is the portion of the positive examples that the model predicts correctly: *TP* / (*TP* + *FN*).
5. True-negative rate (TNR) is the portion of the negative test examples that the model predicts correctly: TN / (FP + TN).

In the realm of IDS, false-positive error reduction is a critical, but not sufficient topic. Some of them will cause actual attacks to be overlooked, so effective techniques for false-positive error reduction are those that increase the accuracy of the IDS or maintain existing accuracy. Among the most essential criteria of IDS is that it must be effective, detecting a significant percentage of attacks while maintaining at the same time the rate of false-positive alerts at an acceptable level.

It is challenging to create an effective IDS that generates few false-positive warnings. The reasons for this difficulty include:

1- **Limited execution time**: In many cases, the activities of real attacks on the network differ very little from normal activities, and sometimes the context of these activities is the only clue that determines whether these are attack activities or not. Due to the harsh real-time requirements, the IDS is not able to analyze all running activities to the required extent [83].

2- **Privacy of detection signatures**: It is also challenging to write signatures for signature-based IDS that describe patterns of hostile behavior on the network [84]. Sometimes, it is difficult to draw a clear balance between increases in specific signatures (unable to detect new attacks), and an increase in public signatures (which can describe normal activities as attacks).
3- **Limitations of the approved environment**: Some activities that can be classified as attacks in one environment may be considered normal activities in another environment. For example, scanning the computers on the network is considered illegal activity unless the person performing this operation is authorized to perform it. Many popular IDS are built on criteria that characterize natural activities as illegal.
4- **The base rate fallacy**: From a strictly statistical standpoint, even IDS with very few positive error rates do not produce desirable detection rates, because real attacks on the network are activities that are rare in occurrence.

Since the number of false positives represents the key limitation of the network intrusion detection anomaly-based strategy, several researchers focused on creating new approaches to decrease the number of false positives caused by anomaly detection. To achieve this goal, the prior art uses statistics, time series, and algorithms for machine learning [19][85].

4.3 Overview of the IDS detection methods

Research on intrusion detection began in the 1980s when the first papers were published and many flavours of IDSs were suggested. Current IDS taxonomy [4] is defined by several features, including the data examination (log, device, or network data), the type of analysis performed (real-time or offline), the type of data processing performed (centralized or distributed), and the response to an attack (active or passive). However, the most important classification of IDSs is related to the intrusion detection methods, recognizing signature-based or anomaly-based approaches [85].

It is also possible to classify IDS from the perspective of the reaction that it performs. There are so-called passive IDS, reporting the information about these attacks to the user interface. On the other hand, there is an active IDS that makes an automatic response when an attack occurs through firewall programming to block suspicious connections. It can also classify the IDS depending on the source of information. Network-Based IDS analyze computer events or related network activities, internet addresses being requested or received from, service, ports, etc. While Host-based IDS analyzes events or activities such as operating procedures or system calls, mainly those activities related to the operating system itself.

4.3.1 Signature-based Intrusion Detection System

Signature-based IDSs are based on recognized patterns in packet payloads, so-called signatures, and match them with suspicious activities to be classified as attacks. The number of anomalies in IP

networks is rising, and most existing IDS are signature-based (rule-based or misuse-based), do not provide adequate protection as it is ineffective against new threats, so-called zero-day attacks.

Furthermore, signature-based IDS techniques fail to identify encrypted communication. It is reactive mostly because it relies on a collection of signatures that must be updated on a regular basis. As a result, it will be unable to detect a specific attack until a rule for the relevant fault is created, confirmed, published, and implemented, which takes a significantly longer time [1][44].

4.3.2 Anomaly-based Intrusion Detection System

Various technologies, such as the anomaly-based network IDS, have been built continuously to resolve signature-based weaknesses. The anomaly-based detection addresses the problem of detecting unusual patterns in network activity which does not match the desired behavior. In various application fields, these non-conforming patterns are referred to as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, oddities, or contaminants [86].

Effective network anomaly detection is critical as being among the possible alternatives to supplement signature-based approaches. A recent study has concentrated on the creation of new methodologies, techniques, and security systems that employ smart solutions for network anomaly detection, which has been recognized as a compulsory part of modern safety defense. The anomaly distribution framework may facilitate mostly the classification of abnormalities into appropriate categories automatically. This is a significant advancement over heuristic rule-based classifications in that it can handle new, undiscovered anomalies whilst also disclosing their peculiar properties.

Anomaly detection has a wide range of applications, including fraud detection, malfunction detection, and device health monitoring. However, this study focuses on the application of anomaly detection in the network intrusion detection area, which has recently become a hot topic. [1][51][85].

4.4 Network traffic analysis

The key assumption of the IDS based on anomalies is that the attacker's behavior differs considerably from the legitimate user and that certain unauthorized activities are observable as statistical anomalies. The anomaly is described as an observation that deviates so much from other observations as to give rise to suspicions.

Anomaly detection systems are divided into two categories: adaptive and static. The static detection systems usually leverage some domain information and scan for the anomalies using a specific subset of features and thresholding. On the other hand, adaptive detectors preserve normal user behavior models or usual network status and mark the deviations from this model as anomalous. Normal network behavior models are generally unknown, widely differ among the networks, and in the strictest sense, it

is very difficult to create an anomaly detector. Therefore, they usually use the presumption that most of the network traffic is normal, and malicious behaviors are correlated with anomalies.

In terms of traffic analysis, there are two primary approaches, namely volume-based and behavior-based methods for anomaly detection. In feature-based techniques, they used packet header analysis, whereas volume-based methods monitor changes in traffic volume. Traditional detection methods used statistical approaches to detect attacks, but with the complexity of attacks, they switched to behavior-based methods in an effort to deal with the traffic deviation over time [20][44][87][88]. These two strategies are explained in more details next subsections.

4.4.1 Volume-based approach

An anomaly is viewed as finding abnormal patterns in network traffic volume. A conventional approach is based on statistical techniques to detect anomalies where simple counters like the number of flows, packets (total, forwarded, fragmented, and rejected), and bytes (per packet, per second) may be retrieved from network devices using SNMP or NetFlow, it used to detect significant volume events like network outages, bandwidth flooding, and flash crowding.

Some abnormalities are minor and hidden in the number of flows, packets, or bytes in traffic volume, so it is extremely difficult to detect them with common approaches and methods that depend mainly on changes in traffic volume. Since examining the volumetric characteristics (bytes and packets) that characterize this behavior can only detect the attacks, like Botnet, DOS, and DDOS attacks, because it generates a lot of traffic volume which can be easily detected by monitoring network traffic volume [44][89]. However, many events that do not produce volume fluctuations stay unnoticed. Furthermore, many researchers identified several technical challenges with counters, stating that sampling packets employed by several routers conserve resources. While gathering information may impact counter-based anomaly detection metrics, it has no substantial impact on the distribution of traffic characteristics [90]. Setting a lower threshold level for such volumetric features will generate a significant number of false-positive warnings, which essentially makes it useless for less intensive anomalies.

4.4.2 Behavior-based approach

Regular traffic with massive data transmission is not uncommon traffic activity in modern network communications. Unlike volume-based approaches, behavior-based techniques is based on non-volumetric traffic characteristics and treats anomalies as occurrences that disrupt distributions of their features. This approach has two main advantages:

- It allows for the identification of traffic irregularities that are difficult to distinguish. Some abnormalities, such as scans or tiny passord guessing attacks, might have a little influence on the quantity of traffic.
- The unusual distribution shows useful knowledge for anomaly patterns which is not apparent in traffic volume computation.

Some of these approaches have been commercialized and are thus not applicable to the research group [88][91].

5. PROPOSED METHODOLOGY

The research presented in this thesis focuses on the network traffic anomaly detection method, with an important objective to propose a solution feasible for practical implementation in the general network environment. For this reason, only basic flow features have been chosen because they can be easily collected from network routers using NetFlow protocol or similar industrial standards. We address this issue by providing a systematic methodology with the main novelty in anomaly classification based on the entropy of flow count and behavior features of data obtained by NetFlow protocol. We also generalize the concept of degree features, propose data partitioning for greater efficiency in real-time anomaly detection. Through an analysis of the most prominent security attacks, generalized network behavior models were developed to describe various communication patterns. Based on a multivariate analysis of entropy changes in each of the modelled classes, experiments were used to design and test anomaly classification criteria.

5.1 Data sources

When working with the massive amounts of data present in today's high-speed networks, well-known IDS like Snort and Bro need many resources. In addition, the introduction of protected communications protocols presents a new challenge to payload-based systems. Flow-based methods tend to be attractive candidates for intrusion detection work in light of these problems. Flows are tracked using sophisticated accounting modules, which are generally found in networking devices [92].

A network flow is a group of packets exchanged in the communication between two endpoints. Cisco introduced the definition of network flows, which is currently standardized by the Internet Engineering Task Force (IETF)[3]. According to the working group of IETF IPFIX workgroup “A flow is described as a collection of IP packets passing an observation point in the network over a certain time interval." A set of characteristics is shared by all packets in a given flow: source and destination IP addresses, source and destination ports and protocol [45]. A flow is commonly described as a one-way packet sequence, which implies that each connection will have two flows, one from the client to the server and the other from the server to the client. Bidirectional flows have lately gained the favour of one record for both directions between two endpoints. Flow exporting protocols, such as Cisco NetFlow [18] or the forthcoming standardized IPFIX [93], are commonly available on network devices, with the capability to collect, account and export the information about flows.

Flows provide an aggregated picture of network traffic by concentrating on the number of packets and bytes transferred across the network.[94]. Therefore, the volume of data to be analyzed is greatly decreased by flows compared to packet-based capture and inspection. This technique is based on network devices' capacity to account for delivered packets and bytesstatistics and export them to a centralized

[3] https://www.ietf.org/

collector server. As a passive appliance, modern approaches employ specialized probes that are transparently coupled so instead of employing routers to export flows, span ports or network taps are used [95].

However, in certain cases, the lack of entire payload is still considered as a major disadvantage of flow-based methods. Detecting so-called semantic attacks, for example, is extremely difficult using flow-based techniques, because the damaging force is contained inside the payload and hence does not cause apparent flow fluctuations (number of bytes, number of packets, or number of flows). In the case of Botnet detection, however, researchers propose analyzing flows to find behavioral trends, i.e., automatically grouping hosts with similar suspicious behavior over a set period of time [39].

5.2 Methodology

The payload of a packet is invariably lost in network flows. As a result, on the network layer, any attacks that manifest simply in packet payload (e.g., remote exploits) remain undetectable. Relative to the well-known attacks or the most prominent types of cyber-attacks described in Chapter 3 section 3.2, Flow-based techniques sometimes only address a subset of threats, namely those that can be identified without knowing the payload's content [50].

For high-speed networks, flow-based intrusion detection is a potential solution. The question of whether or not flow data provide sufficient information to be useful for intrusion detection compared with payload inspection? Flow statistics are measures that are aggregated by definition. As a result, they lack the accuracy of payload-based identification. However, knowledge can be derived from flows about traffic communications, for example by observing the logs of network activities. Depend on the amount of packets, bytes, and calculated flows, flow statistics provide an aggregated view of the data transported over through the network as well as between hosts. It is usually regarded as the most efficient method for monitoring network traffic in network management, i.e., in a streaming fashion [96].

We concentrated our research on flow-based network control because it is more adaptable in terms of network speed, and gives optimal amount of information, which leads to less processing overhead. Also we use entropy-based detection approaches because it is well-suited to real-time traffic analysis and can handle massive volumes of data. Entropy is a measure of network traffic pattern diversity or similarity; consequently, while changing the values of particular traffic aspects, the characteristics of the traffic may be influenced. Actually, the variations in entropy levels might signal occurrence of the malware activity, attacks, or abnormality.

This chapter presents details of the proposed architecture and methodology for network traffic anomaly detection and classification based on the collected flow records. The feature selection method is formalized and broadened by defining the aggregate key features and computed behavior features, as well as proposing feature annotations. The ability to detect anomalous behavior of all entropy kinds is examined in terms of changes in data distributions. The approach for protecting against entropy deception

and detection methodology enhanced with anomaly classification rules utilizing a multivariate analysis of entropy findings, which is based on patterns in the way the features are impacted by different abnormalities.

5.3 Entropy calculation

Knowledge theory's core premise is that the greater you know about a subject, the less new information you can get. If an incident is very likely when it occurs, it is no surprise and offers no new knowledge. Conversely, if the incident was unlikely then the occurrence is much more descriptive. The knowledge content is the likelihood of an event's reciprocal function (1/p, where p is the probability of the occurrence). If several occurrences are possible, entropy estimates the average quantity of knowledge you should anticipate if one of them happens. Clausius proposed the idea of entropy as a metric of chaos in the early 1850s, and it originates from thermodynamics. Shannon used entropy as the basis for the knowledge theory in 1948. The assessment of ambiguity due to the random variable is known as entropy in information theory, the larger the entropy greater random the variable [45].

Entropy may be used to summarize distribution of attributes in a minimal way, i.e. as a single number. Only a few types of entropy have been employed to discover network abnormalities, the most popular of which is the well-known Shannon entropy [44].
The network anomaly detection approach based on Entropy has been of great interest and a hot research subject used by many researchers in identifying unusual network activity caused by attacks. The approach is based on distributions of traffic features, with the main two benefits for using entropy to analyze changes in traffic distribution:

- Anomaly detection performance is greater in comparison to volume-based methods.
- The entropy-based approach can help characterize abnormalities (worms, DDoS attack scans) in a variety of different forms.

The most well-known and commonly used entopy type is the Shannon entropy as a systematic notion at the crossroads between probability, information theory, complex systems and statistical physics [44][51]The Shannon entropy for a discrete random variable X with a probability distribution p(X =x_i) is defined as:

$$H_S(X) = \sum_{i=1}^{n} p(x_i)\ log_a \frac{1}{p(x_i)} \qquad (1)$$

Where X represents the attributes that would take the values $\{x_1 \ldots x_n\}$ and p(x_i) represents the probability mass value of the x_i result. The estimated value of $\log_a \frac{1}{p(x_i)}$, where X drowns and as per the probability mass function p(x), may also be read as the entropy of X. Based on the logarithmic base, other units such as bits (a = 2), natural (a = e), or decades (a = 10) may be employed. For anomaly detection sampled probabilities computed from a number of x_i events over a time period (t) are often employed.

(Where $0 \log_2 0 = 0$). Probability vectors of a probability space arise naturally in conjunction with finite partitions [50][97][98].

In anomaly detection techniques entropy is used to presents the level of randomness in a data series. For the feature distributions obtained from the aggregation process, entropy gives a single number which presents the internal structure of data in a simple way. The changes of data structure in a distribution will change the entropy value. If the entropy change is significant, it is considered as an unusual behavior in network communication or an anomaly, which often indicates security threats [99].

The normalized form of the Shannon entropy [40], giving is within the range between 0 and 1, is defined by the following equation:

$$H(X) = \frac{1}{\log_a N} \sum_{i=1}^{N} p(x_i) \log_a \frac{1}{p(x_i)} \quad (2)$$

In the general case, N is the total number of elements in a data series and $p(x_i)$ is the probability of occurrence of element x_i. In our approach, the data series is a distribution of feature values, while $p(x_i)$ is an empirical probability, calculated by the relative contribution of element x_i with value m_i in the total sum of all values, M:

$$p(x_i) = \frac{m_i}{M} \quad (3)$$

$$M = \sum_{i=1}^{N} m_i \quad (4)$$

Rényi [100] and Tsallis [101] entropies involve an additional parameter α, where positive values put more weight on highest values in the distribution (*peak*), while negative values favourite elements with low values in the distribution (*tail*):

$$H_R(X) = \frac{1}{1-\alpha} \log_a \left(\sum_{i=1}^{N} p(x_i)^{\alpha} \right) \quad (5)$$

$$H_T(X) = \frac{1}{1-\alpha} \left(\sum_{i=1}^{N} p(x_i)^{\alpha} - 1 \right) \quad (6)$$

In this research, we use a scaling factor to normalize the entropy to a value of 1 for fully randomized distribution. The scaling factor for Shannon and Rényi entropy is $1/\log_a N$ and for Tsallis entropy it is $(1-\alpha)/(N^{1-\alpha}-1)$. With such a scaling, the Shannon entropy always provides values between 0 and 1, as well as Rényi and Tsallis entropies with the positive parameter α, while negative parameter α results to entropy values above 1.

The Shannon entropy defined above is normalized, providing values between 0 and 1. If a feature distribution is well balanced with more or less similar values, entropy is near 1, while a significant deviation in distribution values results in lower entropy. It is noteworthy that the proposed methodology is applicable to other types of parameterized entropy, such as Tsallis and Rényi entropy.

5.4 Entropy change detection

Over time, the aggregation and entropy calculation process will generate time series of entropy values for each feature. To detect changes in these time series, a baselining need to be done first. A commonly used approach is based on the Exponential Moving Average (EMA) technique for short trend prediction [102] or taking maximum and minimum values from the sliding time window of some epochs. Both techniques can be used, but the rest of the presented research is based on the EMA prediction technique. With this approach, a predicted value in epoch *n*, denoted as $\hat{H}_n$, is calculated recursively, taking into account the previously predicted value $\hat{H}_{n-1}$ and the calculated entropy value H_{n-1} in epoch *n-1*:

$$\hat{H}_n = (1- \alpha_h)\, \hat{H}_{n-1} + \alpha_h H_{n-1} \quad (7)$$

The coefficient α_h represents the degree of weighting decrease, so-called smoothening factor, which falls in the range between 0 and 1. A lower value for α_h indicates a stronger influence of the previously predicted value $\hat{H}_{n-1}$, resulting in smoother baselining, while at higher values for α_h the predicted values adopt and follow recent data H_{n-1} faster in the observed data sequence.
Some entropy time series can regularly vary its values more than the others. In order to identify significant entropy changes, we propose to analyse these deviations in the context of standard deviation (*S*), using the same EMA baselining approach:

$$\hat{S}_n = (1- \alpha_s)\, \hat{S}_{n-1} + \alpha_s S_{n-1} \quad (8)$$

Finally, the range of acceptable entropy values considered as normal is defined by lower and upper thresholds, as measures of acceptable deviation from the baselined entropy value $\hat{H}_n$ as follows:

$$T_n = [\underline{T}_n , \overline{T}_n] \quad (9)$$

where

$$\underline{T}_n = \hat{H}_n - k_t \hat{S}_n \quad (10)$$

$$\overline{T}_n = \hat{H}_n + k_t \hat{S}_n \quad (11)$$

and k_t is the multiplication factor that makes the range wider, the so-called threshold factor. For any entropy value H_n that fall out of the threshold range T_n in epoch *n*, an alarm is triggered as an indication of anomaly.

With a proper tuning of parameters α_h, α_s and k_t, the above-mentioned technique efficiently detects significant changes in the observed values. We have empirically concluded that the optimal baselining trend is achieved by the following smoothing coefficient values: α_h=0.1 and α_s=0.05. For all experiments, the threshold factor was set to k_t=4, which accurately capture anomalies, while still eliminating the most of false positive alarms.

Some authors claim that parametrised Tsallis and Rényi entropy outperform the Shannon entropy in terms of the better detection of peaks or tails in the feature distributions [44][103][104]. We believe that their conclusions are tightly related to the applied detection methods, data and features used in the

experiments, and accordingly, this conclusion cannot be simply generalised. For that reason, we analyse and compare the Shannon, Rényi, and Tsallis entropies from two main aspects: the ability to detect anomalies and sensitivity to deception. For Rényi and Tsallis entropies, we will use a fixed value of the parameter α (+2 and -2), which is shown to provide optimal performances [44][103].

To better understand the behavior of each entropy type, we will consider a reciprocal distribution of 100 elements, given by the function 1/x, where the distribution starts with values 100, 50, 33, 25, and ends with 'a long tail' of value 1. According to our experiments, this distribution roughly approximates a deviation of flow feature values in real network traffic, which is also reported in [104]. Gradually increasing the peak of the distribution, from the value of 100 to 1000, the entropy is changed in the way presented in Figure 5. 1. The Shannon entropy, as well as parametrised entropies with the positive parameter α, results in decreased values, while negative parameter α leads to entropy increase.

On the other hand, increasing the tail of the distribution up to 1000 new elements with value 1 involves more similarities in the data, and consequently, the entropies approach to value 1, which is shown in Figure 5. 2. In all cases, the Rényi entropy with positive parameter gives the lowest entropy values, while Tsallis entropy gives much higher values (in a range from 1.7 to 106), which are not shown since they are out of the scale used in the chart. It is worth to highlight that the entropy with lower values leaves less space to detect a drop, especially when the standard deviation is higher. This is the case with the Rényi entropy with a positive parameter, which is more sensitive to the regular variation of data (highest slope in Figure 5. 1) and also provides the lowest values.

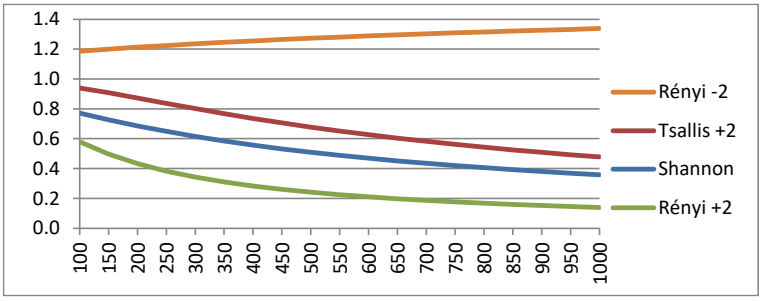

Figure 5. 1 The entropy change given by increase of the distribution peak.

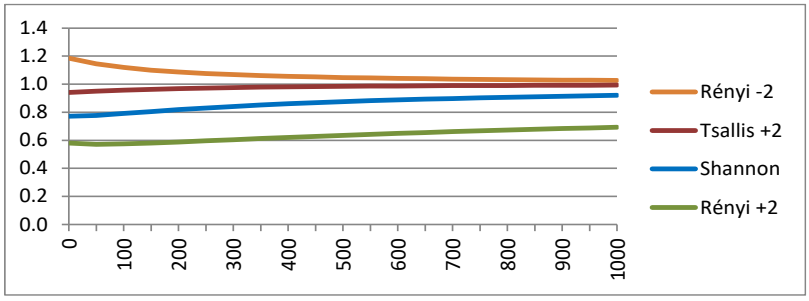

Figure 5. 2 The entropy change given by increase of the distribution tail.

It should be highlighted that features with smaller standard deviations generally provide more distinguished changes, which gives better detection ability. Also, more randomized distribution and the entropy values near 1 generally leave more space for entropy drop and its detection. From the figures presented above, it should be concluded that the Rényi +2 entropy type gives the lowest values, and in case of higher standard deviation, there will be not enough space to detect changes.

5.5 Protection against entropy deception

In entropy-based approaches, anomalies are usually detected by features that generate a peak in the data distribution. This peak will make the entropy drop or increase with regards to entropy type and parameter α. Anyhow, the authors in [105] have shown that every entropy change caused by a peak in a distribution can be suppressed by adding more elements of the average value to make distribution more even. The same effect can be also achieved using value 1 for each added element but much more elements are needed in this case. With this method, attackers can camouflage the attack by generating spoofed traffic in parallel to the attack, and effectively deceive the entropy-based detection systems.

To provide a protection mechanism to this entropy deception, we analysed the effect of entropy suppression on different entropy types, as well as to different features. For the previously mentioned reciprocal distribution with peak values of 200, 500, and 1000, the average values in the distribution are 5, 9, and 13, respectively. The number of elements needed to suppress these peaks using both average values and reference value equal to 1 for each entropy type, is given in Table 5. 1 The Rényi entropy with positive parameter α ('Rényi +2') and Tsallis entropy with negative parameter α ('Tsallis -2') require the highest number of injected elements using the average value. However, this number is much higher for 'Rényi +2' entropy when adding elements at the end of distribution using value 1.

Table 5. 1 The number of elements needed to deceive entropy.

Entropy type	Peak / average					
	200/5	200/1	500/9	500/1	1000/13	1000/1
Shannon	34	275	97	935	143	2135
Tsallis +2	53	280	130	1185	206	2750
Rényi +2	82	1135	207	5275	356	15200
Tsallis -2	38	45	365	166	694	348
Rényi -2	24	28	125	98	273	195

The results from Table 5. 1 leads to the conclusion that the deception of one entropy type does not necessarily mean that the other entropy types are deceived too. This expectation is confirmed in Table 5. 2 which shows the ratio of entropies before and after a deception in our base reciprocal distribution with a peak of 1000 and adding elements with average values 13. When nulling one entropy type (in rows), the other entropies (in columns) are below or above the initial values.

Table 5. 2 Relative differences in deceiving different entropy types.

Entropy type	Peak =1000, average=13				
	Shannon	Tsallis +2	Rényi +2	Tsallis -2	Rényi -2
Shannon	0%	-4%	-27%	220%	3%
Tsallis +2	7%	0%	-17%	153%	2%
Rényi +2	16%	4%	0%	67%	-2%
Tsallis -2	22%	5%	18%	0%	-5%
Rényi -2	12%	2%	-8%	108%	0%

The entropy deception method proposed in [105] addresses only one feature distribution, while other features are not considered. Like the analysis of different entropy types, we can generally expect that different features are differently affected by the spoofed traffic. This disbalance especially holds when injecting new elements in a behavior feature distribution using average value, since the spoofed flows with aggregation attributes must be repeated using distinct values of behavior feature. The easiest approach is to use full randomization of all attributes in the spoofed traffic, which would produce the elements with a value of 1 at the end of the feature distributions. However, this would significantly increase the number of distinct elements in a feature distribution, what is the case with the "Rényi +2" entropy in Table 5. 1 with 15.200 new elements. It is also noteworthy that it is relatively easy to generate spoofed traffic to the targeted victim network, but this traffic will be highly asymmetric, mostly with no reply in opposite direction.

According to the previous analysis, we propose a protection method against entropy deception attempts, which relies on the detection of spoofed injected traffic that camouflage the attacks, based on the following principles:

- Prefer the entropy type which requires more injected elements to deceive the entropy (such as 'Rényi +2')
- Use the number of distinct elements in a feature distribution as a new detection metric, named as a distribution length. This metric has not been used in the scientific literature so far.
- Monitor the flow count of asymmetric traffic (traffic with no reply) as an indication of spoofed traffic.

Experimental results that validate the proposed protection method are presented in the Chapter 6.

5.6 Communication pattern modelling

To better investigate the behavior of different anomalies in terms of aggregation keys and the corresponding features, we have analysed normal network behavior and the communication characteristics of the most prominent network security attacks.

The client-server communication model is considered in an ordinary network operation. In bidirectional flow records the client initiates communication as a source, choosing a random source port to access the server on a fixed destination IP address and port number. From the server point of view, a single IP address and port number reply as a destination to many different source IP addresses and source ports. The source and destination packets and bytes in bidirectional flow records are therefore related to the client and server respectively.

Security threats usually follow the same client-server model, but the magnitude of some communication characteristics is much higher. DDoS attacks are performed by sending traffic from a large number of sources towards a targeted destination. A DDoS amplification attack utilizes services such as DNS or NTP on servers that are not properly configured, the so-called 'open servers' [106]. The attacker sends a large number of small queries with a spoofed source IP address of the targeted host, and all servers reply to it, generating traffic of a much higher magnitude. In October and November 2016, two websites within the network of the University of Belgrade were attacked by NTP and DNS amplification attacks respectively. A single UDP source port number was used as a source of the attack (123 for NTP and 53 for DNS), but the destination port for the DNS attack was fixed to HTTP, while the NTP used a random destination port.

In both cases, more than 1000 open servers generated up to 4Gbps traffic for 20 to 30 minutes, bringing down not only the attacked web servers but also disrupting other services due to the overload of the uplink of the entire national research and education network AMRES. The intensity of the attacks was easily detected and mitigated by the NetFlow Analyzer tool using volumetric statistics only (bytes, packets and flows) [107]. However, to detect less intensive attacks that may remain 'under the radar', the communication pattern with other features must be analyzed.

On the other hand, many security threats start much earlier, before real damage is caused. Network scan is looking for an open service on the network, generating flows from a single source IP address and usually an arbitrary source port toward a fixed destination port on many hosts over an enterprise network. A Port scan is a method for determining which ports on a single host are open, producing many flows with a different destination port and fixed destination IP address [108].

Once a host is located with the open TCP port requiring authentication, such as port 22 for SSH or 3389 for Microsoft Remote Desktop, the attacker can perform brute-force password-guessing activities, trying commonly used phrases by a dictionary attack [109]. The footprint of this traffic structure is characterized by too many short flows with one or two packets transferred between two fixed IP addresses, using multiple source ports and a single destination port. All of the above-mentioned network behaviors have a very specific communication pattern marked by single or multiple source and destination IP addresses and port numbers involved.

These characteristics can be simply described using the label ‘1’ for a single or ‘N’ for multiple occurrences of identification features in the order from the source IP address (‘S’) and source port (‘s’) to the destination IP address (‘D’) and destination port (‘d’), in form of ‘Ss-Dd’. In this way, previously analyzed anomalies can be categorized as follows:

DNS amplification DDoS	–	N1-11;
NTP amplification DDoS	–	N1-1N;
Port scan	–	1N-1N;
Network scan	–	1N-N1;
Dictionary attack	–	1N-11.

This classification and labelling can be further generalized to cover all 16 permutations of labels of ‘1’ and ‘N’ for source and destination IP addresses and ports. With this generalized approach, a network scan with a fixed source port is related to the communication pattern of class 11-N1, while a distributed SYN flood attack falls into the class NN-11, since the attack is performed from many source IP addresses and port numbers to a single destination IP address and TCP port number.

Regular network traffic can be also described with the introduced labelling of the communication patterns. A client can initiate several connections to a certain server, which falls into 1N-11 class, while public servers, which are used by many clients, fall into the class NN-11. Additionally, a DNS, SMTP and HTTP proxy services follow the 1N-N1 pattern, acting as a client establishing communications with many other servers.

It is noteworthy that the 11-11 model is not realistic, since the NetFlow protocol in short period treats all flows with the same source and destination IP addresses and port numbers as a single communication and will, accordingly, generate only one flow record. However, this model is retained to fill the theoretical gap.

5.7 Architecture

A high-level architecture of the proposed methodology is illustrated in Figure 5. 3 The architecture consists of the following main building blocks:

- Flow Preprocessing – Since the original flows collected from network devices are unidirectional, two flows from both communication directions between two peers are paired into a single record, so-called bidirectional flow, which gives more information and provides better detection efficiency [110].
- Control dataset – Flow data of regular network traffic combined with modelled synthetic data are used for analysis, testing and tuning of the system.
- Flow partitioning – Bidirectional flows are filtered by protocols, services or IP addresses, and divided into different categories, which are analyzed separately. It is shown that this approach

leads to more efficient detection of the low-intensive anomalies that may stay undetected if the analysis is applied to the whole traffic.

- Aggregation – All flow data during fixed intervals (so-called epochs) are aggregated based on identification features, summarizing other flow attributes and calculating additional features that represent communication behavior. The results are data distributions for each aggregation key and feature used.
- Entropy calculation – The entropy is calculated over each data distribution, for every feature and epoch, generating a number of time series entropy values.
- Entropy change detection – Significant change of entropy value indicates the change in network communication behavior. The challenge is to accurately recognize changes caused by the anomaly and distinguish them from normal traffic variation.
- Multivariate Analysis – Triggered entropy changes are mutually analyzed to provide proper anomaly classification with higher detection accuracy.
- User interface – For practical implementation and usage in a real-life network, the results obtained from the anomaly detection module need to be properly presented and managed, which includes simplified and meaningful visualization, root cause analysis to extract anomalous data, efficient alarming, configuration and data curation.

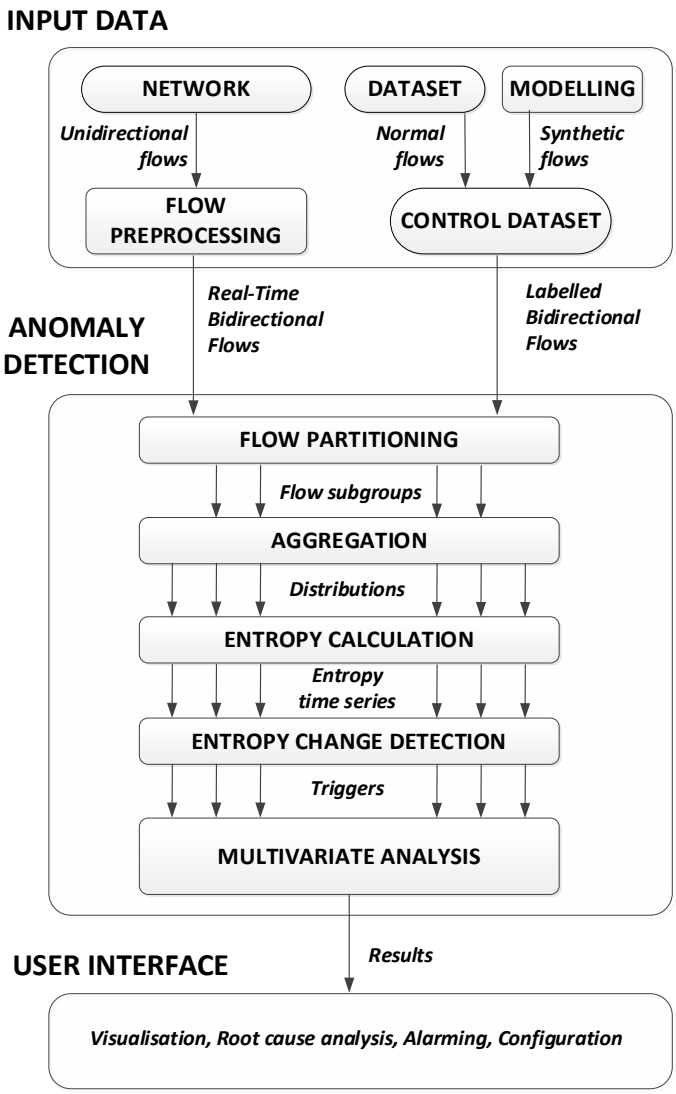

Figure 5. 3 High-level architecture of network traffic anomaly detection and classification based on the entropy of NetFlow data.

5.8 Flow collection and preprocessing

Original raw NetFlow records, the so-called flows, are unidirectional and identified by the source and destination IP addresses, protocol and port numbers, while the total packet and byte counts represent a traffic volume in the direction from the source to the destination. The communication in the opposite direction between the two peers is logged by another flow, with replaced source and destination IP addresses and ports. When these two flows are combined into a single record (bidirectional flow), the order of the IP addresses and ports is important. The first is the host that initiates communication (initiator), while the second host is the responder. Basically, the initiator is the source from the first flow and the responder is the source from the replying flow in the opposite direction. However, in the related research papers and datasets, the initiator and the responder in bidirectional flows are widely labelled as the source and the destination, respectively. Consequently, source packets and bytes relate to the data

volume transferred from the source to the destination, while the destination bytes and packets relate to data transfer in the opposite direction. Bidirectional flows, therefore, carry more information about the communication pattern, and this is confirmed to be more useful in anomaly detection [44][110].

The source and destination IP addresses and port numbers, as well as the protocol type, identify the flow, representing identification features, while packet and byte numbers in each direction are used as a metric for volume, representing volumetric features. In this research, we label the source and destination IP address with capital letters '*S*' and '*D*', the source and destination ports with lowercase letters '*s*' and '*d*', and the protocol with the letter 'P'. The source and destination packet and byte counts are labelled as '*sP*', '*dP*', '*sB*' and '*sB*' respectively. More formally, we can introduce a set of identification features *I* and a set of volumetric features *V*:

$$I = \{S, D, P, s, d\} \quad (12)$$

$$V = \{sP, dP, sB, dB\} \quad (13)$$

5.9 Flow partitioning

The detection ability of entropy-based approaches highly depends on the relative amount of anomalous activities in comparison to regular network behavior for the observed feature. If a network is heavily loaded with regular traffic, the straightforward detection approach is limited only to high-intensity anomalies, while less aggressive malicious activities may remain undetected.

To address the above-mentioned issue, we propose the partitioning of network traffic into small subgroups prior to applying a detection technique to each subgroup separately. This partitioning can be based on different criteria, such as the protocol type (TCP, UDP, ICMP), service type (DNS, email, web service, windows services etc.) or subnetworking (user traffic, voice VLANs, data centre, branch offices etc.). Even a single server can be extracted into a separate subgroup if it is sufficiently active, which is often the case with a DNS, email or web server. If the servers are protected by other cybersecurity measures, they can be whitelisted and excluded from the entropy calculation process.

By all means, the intention is to extract similar traffic types into separate subgroups in order to profile optimally its characteristic behavior and allow easier change detection. The conducted experiments, presented further in the text, confirm the great practical usability of this approach.

5.10 Flow aggregation

Efficient entropy-based anomaly detection needs to take into account more details about network activities, especially for low-intensity attacks. This is where aggregation takes place. This is the process of grouping flows based on the value of one or more flow features during a certain period, called epoch. For each distinct aggregated element, so-called aggregation key, all related flows are counted into a flow number, labelled as *f*, while the volumetric features are summarized into total packets and bytes for both directions separately.

Obviously, flow identification features are the most meaningful to be used as the aggregation key. Having in mind that the protocol feature takes just a few distinct values, mostly TCP, UDP and ICMP, aggregation by this feature would not provide useful information. A set of aggregation features is therefore defined as follows:

$$\Phi = \{S, D, s, d\} \quad (14)$$

It is noteworthy that additional features that further describe specific traffic characteristics can be extracted and used in the analysis as well. Some authors use flow time duration ('*TD*') or the average packet size, calculated by dividing the total byte number by the packet number in directions from the source or the destination ('*sPS*', '*dPS*') [110]. These new features also characterize the communications and can be used in the aggregation key. To do so in a meaningful way, they must be first bucked into certain values. For instance, packet size can be bucked in intervals of 10 bytes, giving values 70, 80, 90 etc. It is not mandatory to use constant bucket sizes. This is especially true for the duration feature and it makes sense to divide it into buckets of milliseconds, seconds or minutes. Accordingly, the aggregation set can be extended with these additional features:

$$\Phi' = \{S, D, s, d, TD, sPS, dPS\} \quad (15)$$

However, we will consider only the basic aggregation feature set Φ, given by equation (14) in this research. It should be noted that the aggregation can be done using more than one feature, altogether creating a complex aggregation key, or more formally, using features from any set of the power set of Φ, except an empty set. To annotate the complex aggregation key, we will use feature labels in the following order: '*S'*, '*D'*, '*s'*, and '*d',* separated by the character '.'. Therefore, a total of 15 aggregation keys are available:

$$A = \{\text{'}S\text{'}, \text{'}D\text{'}, \text{'}s\text{'}, \text{'}d\text{'}, \text{'}S.D\text{'}, \text{'}S.s\text{'}, \text{'}S.d\text{'}, \text{'}D.s\text{'}, \text{'}D.d\text{'}, \text{'}s.d\text{'}, \text{'}S.D.s\text{'}, \text{'}S.D.d\text{'}, \text{'}S.s.d\text{'}, D.s.d, \text{'}S.D.s.d\text{'}\} \quad (16)$$

The straightforward aggregation will result in the distribution of flow count and the sum of source/destination packets/bytes of each distinct aggregated element. We will label these distributions using the aggregation key and label of feature used in counting or summarization, separated by the character ':'. For instance, the distribution of the flow count feature, ('*f*'), for all distinct pairs of source IP addresses ('*S*') and destination ports ('*d*') is labelled as '*f*[S.d]', while the source packet number ('*sP*') for the same aggregation key is labelled as ('*sP*[S.d]').

At this point, we will generalize the concept of the in-degree and out-degree features, used in [67][111], which is defined by a number of distinct source hosts per each destination host and a number of distinct destinations hosts per each source host, labelled as ('*S*[D]') and ('*D*[S]'), respectively. Taking into consideration any other identifying features that are not used in the aggregation key, such as source and destination ports, we can additionally count the distinct occurrence of these features per aggregated element. Since they represent the communication behavior of the main aggregated elements, we will call

these additional features '*behavior*' features. More formally, for the set of aggregation features Φ and the aggregation key K, a set of available behavior features is:

$$B(\Phi,K) = \Phi \setminus \{K\} \quad (17)$$

A comprehensive methodology for anomaly detection with the novel classification proposed in this research heavily utilizes behavior features as the main source for network behavior analysis along with flow count feature. To briefly illustrate the usability of this approach, let us consider a DDoS amplification attack, where many source IP addresses send packets to a single destination host, all using the same source port, such as the UDP port number 53 used by DNS amplification attacks [106]. This unusual network behavior can be detected by counting the number of distinct source IP addresses ('*S'*) for each element aggregated by the destination IP address and the source port ('*D.s'*), labelled as ('*S*[D.s]'). The same stands for a port scanning scenario, which can be captured by aggregation with the source and the destination IP address ('*S.D'*) and by counting the occurrence of the distinct destination port, i.e. ('*d*[S.D]').

5.11 Aggregation and entropy calculation algorithm

No matter whether the input flow data are obtained from an off-line prepared dataset or from a real-time NetFlow collector, the methodology assumes that entropy is periodically calculated for each aggregation key and each feature, producing the corresponding time series values for each epoch. Calculating entropies, with EMA prediction and standard deviations, is not a complex process once the distributions are generated by the aggregation process. However, a lot of aggregation jobs need to process a great number of flow records, which is both a CPU and memory intensive process. Each flow record needs to be matched with all aggregation keys separately, counting or summarizing corresponding values of the remaining features.

The pseudo-code for the aggregation and entropy calculation algorithm is given in Algorithm 1. We have used an unordered associative array, namely a hash-map data structure in Java programming language, which has very efficient lookup and inserts operations intensively used in the aggregation process. An array of hash-maps is implemented and indexed by each possible aggregation key, with an associate data structure that includes the following items: flow count, source/destination packet/byte number, and a nested array of hash-maps for the second-degree aggregation of behavior features. For each flow in the observed epoch and all possible keys, the algorithm counts and summarizes the remaining features for each distinct aggregating element. At the end of the epoch, corresponding elements in hash-maps present the resulting data distributions. These distributions are the input for the entropy calculation process, resulting in number of time series entropy data for each aggregation key and corresponding feature.

Algorithm 1: The aggregation and entropy calculation algorithm per epoch.

```
#variable definition
map[] of {          # array of hash-maps, indexed by aggregation key type
       .f,          # flow count
       .sP,         # source packet number
       .dP,         # destination packet number
       .sB,         # source byte number
       .dB,         # destination byte number
       .map   []    # nested hash-map array for the second degree aggregation
}

#start of epoch
init()                                  # variable initialization
for each flow in epoch {                # iterate each flow in current epoch
       for each K in A {                       # iterate each aggregation key type
              k = getKeyInstance(flow, K)       # get key value from flow defined by K
              e = map[K].get(k)                 # get element from hash-map with key k
              e.f = e.f + 1                     # increment flow count
              e.sP = e.sP + f.sP                # sum source packet number
              e.dP = e.dP + f.dP                # sum destination packet number
              e.sB = e.sB + flow.sB             # sum source byte number
              e.dB = e.dB + flow.dB             # sum destination byte number
              for each K2 in Φ\{K} {            # new key type from remaining features
                     k2 = getKeyFromFlowData(flow, K2) # get new key value from flow
                     f2 = e.map[K2].get(k2)     # get flow count from nested hash-map
              e.map[K2].put(k2, f2 + 1)  # increment and store flow count
       }
       map[K].put(k, e)                         # store the element in the hash-map
}

# end of epoch
for each K in A {
# calculate entropies for key type K and features: f, sP, dP, sB, dB
       entropy[K:f][epoch] = calcEnropy(map[K], 'f')
       entropy[K:sP][epoch] = calcEnropy(map[K], 'sP')
       entropy[K:dP][epoch] = calcEnropy(map[K], 'dP')
       entropy[K:sB][epoch] = calcEnropy(map[K], 'sB')
       entropy[K:dB][epoch] = calcEnropy(map[K], 'dB')
       for each K2 in Φ\{K} {
              # calculate entropies for remaining behavior features
              entropy [K:K2][epoch] = calcEnropy(map[K].map[K2])
       }
}
```

The complexity of the algorithm highly depends on its implementation. As previously mentioned, we used an unordered associative array (hash-map), which in most cases has O(1) complexity in time, while the worst-case complexity is O(log *n*) when using balanced search trees, which is created only for a small number of entries sharing the same hash-map key. Complexity in memory space is O(n) in all cases.

Another concern with real-time aggregation process is a high rate of incoming flows, such as tens of thousands per second. A solution to this is to use a flow sampling technique, processing only a statistical fraction of the flow data stream while rejecting the rest. Obviously, some information will be lost in that case, but a sufficient amount of data (up to the processing limit) is taken into account, resulting in a fairly good statistical approximation. A similar sampling technique is also supported by network devices exporting a fraction of NetFlow data.

6. EXPERIMENTAL EVALUATION

6.1 Datasets used

The Anomaly Network Intrusion Detection System (A-NIDS) was analyzed using various databases which are divided into two groups, synthetic and real. Nowadays, no public archive is broad enough to exhaustively check and compare various algorithms to draw meaningful conclusions regarding their performance and classification capabilities.

One of the major issues in research in the field of network anomaly detection is the lack of accurate and publically available datasets for evaluation. The most useful are actual traces of the network but they are rare since they require proper labelling of regular and irregular traffic, which has to be performed mostly manually in certain situations.

In general, the lack of representative datasets makes analyzing different systems and algorithms more challenging. Data collected from the real network involves two main problems:
First, such datasets are not labelled (in other words, it is unclear which attacks are real positive and which are real negative). Labelling is an important barrier since it requires the significant efforts of many experts. The second issue is the vast majority of data acquired by IDS is private, because this data naturally contains knowledge of the network architecture, working machines, and other private information, access to such information is restricted and cannot be shared with outsiders [85]. The other option is to use a dataset with regular traffic with synthetically injected certain types of anomalies.

There are many databases available that are widely used to analysis, evaluate and test proposed methodology [112][113][114][115][116][117], although most of them are not up to the required level of accuracy due to their lack of the latest attacks.

To validate the proposed approach in network behavior analysis and anomaly detection, two labelled datasets have been chosen, namely CICIDS2017 dataset [117] and CTU-13 dataset [115]. CICIDS2017 dataset is one of the latest and most complete flow-based labelled datasets which is publicly available. It includes the most common attack scenarios, covering profiles of Web based, Brute force, DoS, DDoS, Infiltration, Heartbleed, Bot and Scan attacks, each in a different file named by the weekdays when the dataset was created. A total of 80 flow-based features related to network communications were generated by processing real traffic with simulated attacks. The data collection session lasted 5 days, beginning at 9 a.m. on Monday, July 3, 2017, and ending at 5 p.m. on Friday, July 7, 2017. Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS are among the attacks that have been employed. On Tuesday, Wednesday, Thursday, and Friday, they were executed in the morning and afternoon.

The CTU-13 labelled dataset, which consists of 13 independent parts. Each of these datasets consists of internally controlled legitimate traffic, traffic generated by a real botnet network, as well as a significant amount of such background traffic taken from the Czech Technical University network, which was intentionally left with minor ‘noise’ anomalies.

6.2 Validation of the protection against entropy deception

As described in the previous chapter, the attackers can camouflage the attack by generating spoofed traffic in parallel to the attack, and effectively deceive the entropy-based detection systems. One of the main contributes of this research is to introduce a new technique for protection against entropy deception.

The CICIDS2017 dataset was used to show the validation of the protection mechanism against entropy deception, utilizing the trace labelled 'Friday afternoon.' It includes a PortScan attack, which occurs when an attacker attempts to connect to a large number of target ports on a remote victim system in order to uncover vulnerabilities. In this scenario, the 11-1N communication pattern is used to represent the utilization of a single source port during this procedure. The destination port behavior feature using the source and destination IP addresses as the aggregation key ‘d[S.D]’ provides a very random distribution with an entropy value close to 1 and a modest standard deviation for all entropy types, as illustrated in Figure 6. 1. However, throughout the attack, large entropy decreases are visible for all entropy categories in three series from epoch 112 to epoch 145.

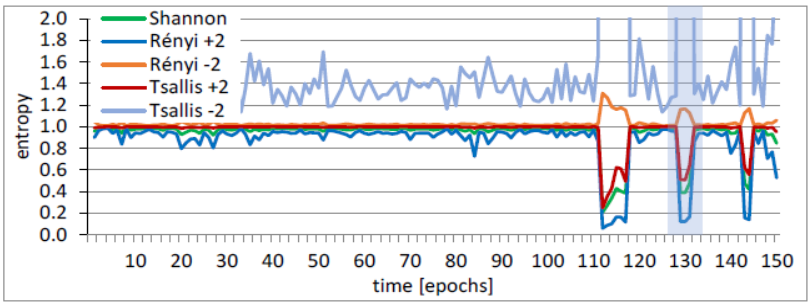

Figure 6. 1 The entropy of the ‘d[S.D]’ feature – the original dataset.

The deception technique will be used only on the second attack, which occurs between epochs 129 and 131 with an average value of 4 in the data distribution, while the first and last attacks will be kept intact for comparative purposes. To spoof the Shannon entropy, an entropy value of 0.38 must be boosted over the threshold value of 0.95 throughout the attack, which necessitates the addition of 3,000 additional elements with an average value of 4. This is accomplished by creating a 3,000-series of four synthetic flows, each with unique source and destination IP addresses and distinct destination port numbers. To camouflage the Shannon entropy, a total of 12,000 synthetic flows were created and added to the dataset during this attack. Figure 6. 2 shows that the Shannon, Rényi -2, and Tsallis +2 have been effectively deceived, while Rényi +2 and Tsallis -2 have also been deceived, but not enough to prevent

discovery. In this example, a total of 22,500 series of 4 synthetic flows must be constructed, requiring a total of 90,000 new spoofed flows throughout each of these epochs to mask the Rényi +2 entropy.

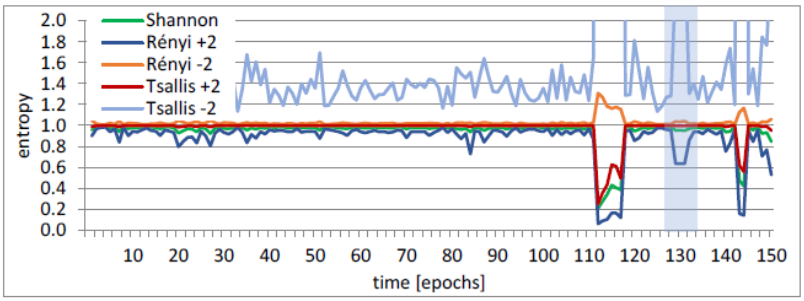

Figure 6. 2 The entropy of the 'd[S.D]' feature – deceiving the Shannon entropy in epochs 129-131.

In addition to entropy calculation, a total number of elements in feature distributions is monitored as a control mechanism to identify this deception effort. The total number of elements that must be added to the 'd[S.D]' feature distribution to deceive all entropy types is shown in Figure 6. 3. The injected traffic clearly surpasses the usual values, particularly for the Rényi +2 entropy type, which is much over the indicated scale.

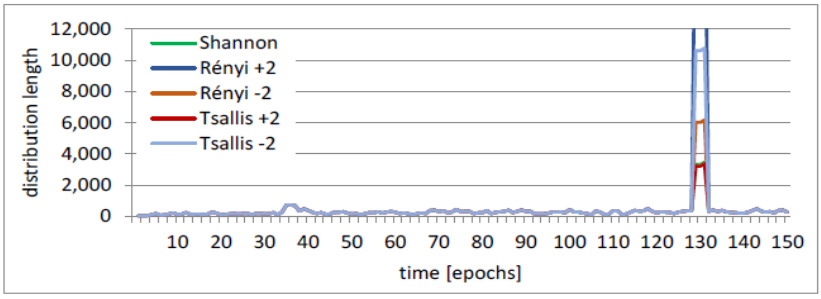

Figure 6. 3 The length of the 'd[S.D]' feature distribution with spoofed traffic.

Even a large number of additional elements for misleading the 'd[S.D]' feature is insufficient to effectively deceive the 'd[S.s]' feature for all entropy types, as illustrated in Figure 6. 4. Many more components must be injected to camouflage the 'd[S.s]' feature, making the suggested measure much simpler to detect.

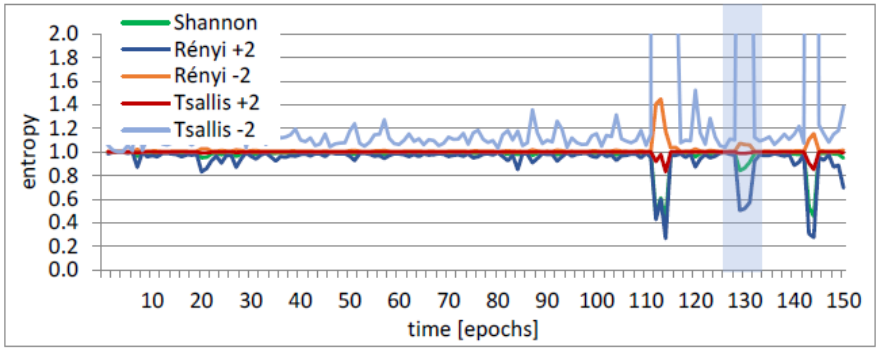

Figure 6. 4 The entropy of the 'd[S.s]' feature, deceiving the feature 'd[S.D]'.

6.3 Anomaly detection methodology validation

The entropy-based anomaly detection approach using pure NetFlow data has been validated on the CICIDS2017 dataset. A DDoS attack was analysed in the dataset part named "Friday afternoon". The attacks were generated by Low Orbit Ion Canon (LOIC) tool, which is sending the UDP, TCP, or HTTP requests to the victim server [117]. The attack follows a communication behavior, which is also used by many clients in normal network traffic, but the intensity of the attacks makes the changes in entropy values that can be easily detected by several features. It is noteworthy that NAT technique was used on the firewall during the dataset creation when three attackers' IP addresses were translated into a single IP address. With this approach, a distributed manner of the attacks was basically lost from the database.

The flow count feature is commonly used in many researches on entropy-based anomaly detection in network traffic [44][110][111][118]. Figure 6. 5 depicts the entropy of the flow count feature using both source and destination IP addresses as the aggregation key, labelled as 'f[S.D]'. Despite a larger standard deviation, a sudden entropy drop is obvious during the attack as a consequently increased flow number between the attacker and the victim hosts. However, several false positive alarms are also triggered. They are caused by larger entropy variations since the observed DoS attack shares the same communication pattern as regular traffic (class '1N-11'), which varies in its nature and occasionally occurs with higher intensity.

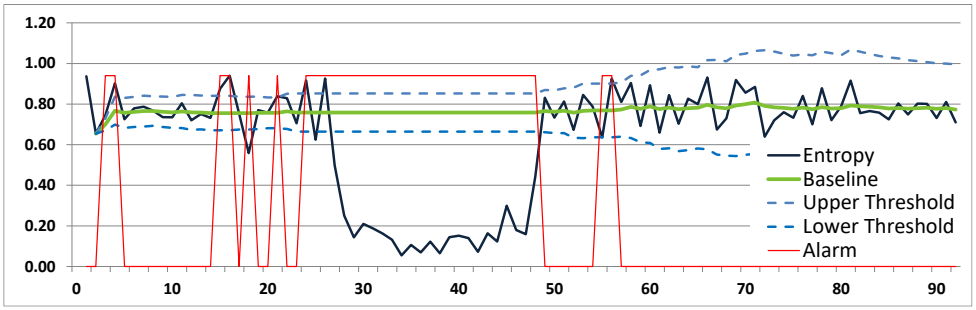

Figure 6. 5 The entropy of the flow count feature, aggregated by the source and destination IP addresses 'f[S.D]'.

False positive alarms can be minimized by increasing the threshold multiplication factor (equations 10 and 11), but this approach would not be of much practical use, since it requires a lot of ad-hoc turning.

It is worth mentioning that the previous attack type can be also detected using several other features, such as 's[S]', 'f[S]','s[D]', and 'f[D]', but using a single aggregation key result to higher entropy variation with even more false positive alarms. In general, using a complex aggregation key with two or more attributes is more specific in flow matching, produces higher value randomness and, accordingly, higher entropy with less standard deviation, which results in easier detection of entropy changes.

Another part of the dataset taken in different periods contains PortScan attack when an attacker is trying to establish connections to many destination ports on a remote victim system in order to find vulnerabilities. When a single source port is used during this process the attack is described by the'11-1N' communication pattern, while using multiple source ports it falls into the '1N-1N' class.

Similarly, to the previous DoS attack, the flow count feature and aggregation by source and destination IP addresses is affected by the attack, but in this case, a standard deviation is even higher, leaving not enough space for a significant entropy drop (not shown). On the other hand, a behavior feature calculated by the second-degree aggregation of the destination port feature 'd[S.D]', generates more randomness distribution with an entropy value near the maximum value (1) and a much smaller standard deviation. Accordingly, the thresholds range used for the normalization is narrow, so the normalized entropy can much more efficiently distinguish PortScan attack from regular network behavior, as it is shown in Figure 6. 6. It is worth mentioning that this feature was not affected by the previous DoS attack. The observed PortScan attack follows the same communication pattern as regular traffic (class '11-1N')

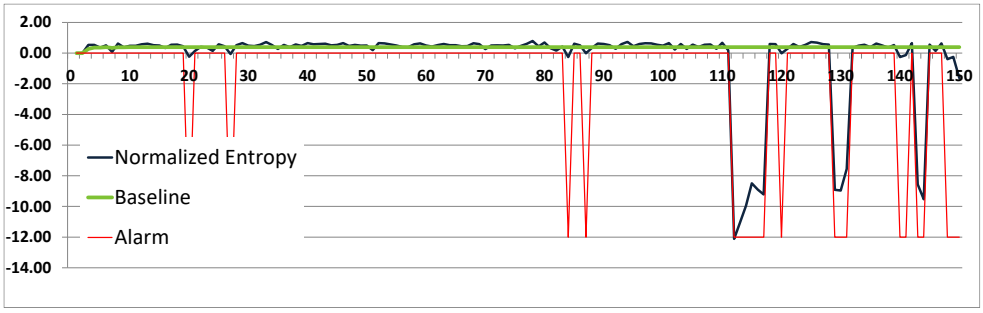

Figure 6. 6 The normalized entropy of the destination port behavior feature, aggregated by the source and destination IP addresses 'd[S.D]'.

Another example demonstrates the infiltration attack captured by "Thursday afternoon" part of CICIDS2017 dataset. During the second phase of the infiltration attack, after leaving a backdoor on the victim's computer, NMAP is performed with the same communication characteristics as the previously described PortScan attack. Figure 6. 7 shows the destination port behavior feature, aggregated by the source IP addresses 'd[S]'. However, the first part of the attack in the dataset occurs with just a few flows per epoch (sometimes only one flow), which is far from enough to be detected by the entropy approach.

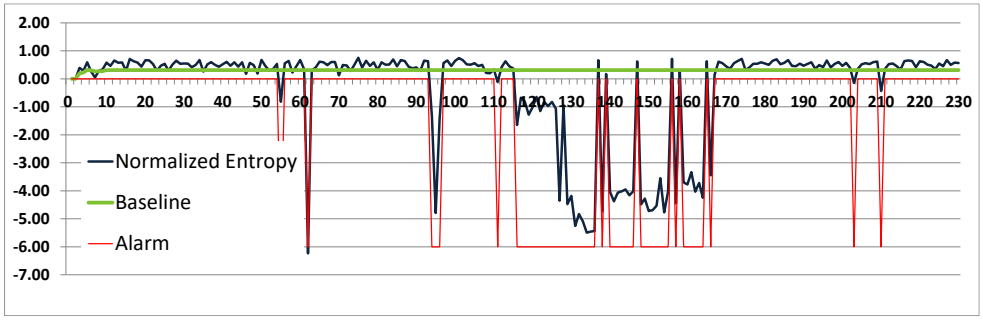

Figure 6. 7 The anomaly score of normalized entropy of the destination port behavior feature, aggregated by the source IP addresses 'd[S]'.

Popular brute force attacks on password cracking are another illustrative example. In the CICIDS2017 dataset, this attack was generated using the SSH-Patator tool in the part named “Tuesday afternoon”. The attack follows the communication pattern of the class ‘1N-11’, similarly to the intensive but regular traffic of web and DNS services. It was performed with less magnitude than total regular traffic, with no significant changes in entropy values, which makes the attack undetectable in this case. This is where the flow partitioning comes into play to separate a large partition of regular traffic from the rest. Filtering TCP protocol only and excluding port numbers 80, 443 and 53, the attack activity becomes noticeable in the entropy drop of several features. Figure 6. 8 shows the normalized entropy of the flow count feature aggregated by the destination port number 'f[d]'. The attack follows the communication pattern of class ‘1N-11’

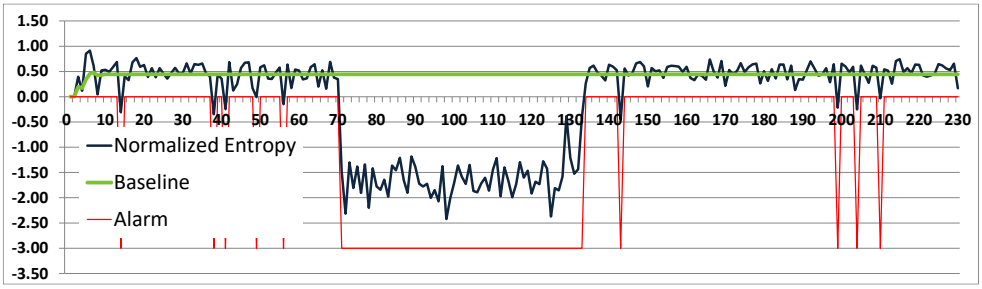

Figure 6. 8 The normalized entropy of the flow count feature, aggregated by the destination port number ‘f[d]’.

On the other hand, the Web attack presented in the ‘Thursday morning’ part of the dataset, has the most frequent occurrences flow count distribution aggregated by source and destination IP addresses 'f[S.D]', but still not enough to significantly change the corresponding entropy values. The same stands even for the parametrized Renyi and Tsallis entropy (not shown), which have the ability to better express more frequent occurrences in data distribution. Flow partitioning in this case is useless since the anomalous flows cannot be distinguished and filtered out from the regular web traffic. This is an example of a network behavior pattern where the entropy of basic flow features is not efficient in anomaly detection.

6.5 Anomaly modelling

The above presented results confirm that the anomalies in network traffic can be efficiently detected if the occurrences of the anomalous behavior are frequent enough to make a significant spike in data distribution and drop in entropy values of the observed feature. However, our attention was attracted by the fact that different types of anomalies leave different footprints in the entropy of corresponding features. In order to better analyse this behavior and provide a key instrument for multivariate analysis, we have modelled characteristic anomalies for each of 16 communication pattern classes (from ‘11-11’ to ‘NN-NN’).

To validate the proposed approach in network behavior analysis and anomaly detection, a reliable dataset needs to be provided, based on real and legitimate network traffic combined with the flow data representing the analysed anomalies. A customized dataset was created for each anomaly model,

combining flows of normal network traffic with the synthetically generated flows representing modelled anomalous behavior. Normal traffic was extracted from the CTU-13 dataset, the trace named '51', with around one million flow records collected during a four-hour period. Using the proposed approach, we have analysed the background traffic and semi-manually removed all recognized anomalies, considering the rest of the traffic as regular and legitimate network behavior. This traffic is not used to test anomaly detection accuracy but rather as ground truth for the analysis of which features are affected by different anomalies, even those of small intensity

Long-lasting flows were proportionally fragmented into short equivalent flows, reaching 60 seconds in duration, which was set as the default duration of one epoch. This was done according to the usual practice for NetFlow configuration in enterprise networks to avoid burst traffic load in only one epoch at the end of the long communication. The timestamp format was converted from a text string into an integer number of milliseconds since the UNIX epoch. Also, some unimportant features were removed, while some additional features were calculated, such as average packet size. Basically, we thereby generated a new dataset with rather clean data representing real and regular traffic with no anomalies. It consisted of 72% UDP, 26% of TCP and 2% of ICMP flow records, where half of the total data belonged to a DNS service, while 20% belonged to an HTTP/HTTPS service. More importantly, we obtained a dataset with a stable traffic structure with no significant deviations over time. This traffic was taken as background used for the analysis of different anomalies, even those of small intensity.

We have modelled anomalous traffic for each class of communication pattern. This synthetic traffic was created by a flow generator, developed by Bereziński [44] and slightly modified in accordance with our own dataset format. Starting very modestly with only 10 synthetic flows per epoch, the intensity gradually increased with 25, 50, 100, 200, 500 and 5000 flows per epoch. Small random variations were involved to present more realistically a stochastic behavior. It should be mentioned that the last anomaly burst was extremely huge in order to check whether the entropy of some features was immune to the anomaly. Moreover, this burst was repeated twice. The first traffic burst had 5000 purely random and mostly unique values of the aggregation feature (labelled in the model with 'N'). The second burst had the same number of flows, but contained 10 times fewer distinct elements, each of them repeated 10 times on average. In all cases, each flow had 1000 packets with 1 Kbyte per packet, both values with a uniform variation of 25%. With this method, having a port scan attack as an example described by the '1N-1N' model, the source and destination IP addresses were fixed in the corresponding synthetic flows, while the source and destination port numbers were randomized. The generated synthetic flows for each modelled anomaly class were injected separately into the dataset with the regular traffic, starting from epoch 80 in short series of three epochs, increasing the intensity every 20 epochs.

In all experiments smoothing coefficients were set to α_h=0.1 and α_s=0.05, while the factor of tolerance was set to k_t=4. In some cases, less intensive anomalies can be better captured with smaller values of factor kt, while higher values can eliminate false positive alarms if they occur. To better address the real efficiency of entropy-based approach, we kept the parameters fixed with no manual adaption to avoid artificially achieved performances.

6.5 The entropy of anomaly models (synthetic traffic results)

The experiments were conducted for each of the 16 anomaly models separately, starting from 11-11 up to NN-NN, aggregating by all aggregation keys from equations (16). Summarizing source and destination byte and packet counts and counting the total flow count and all second-degree features, a total of 103 distributions were generated for each model. As the final result, a total of 1648 series of the Shannon entropies were calculated. In this section, only the most characteristic examples are chosen to present how the entropy of different features is affected by the modelled anomalies.

6.5.1 Volumetric features

Volumetric features, such as the source and destination byte and packet counts, have been used since they are efficient for DDoS and similar volume-intensive attacks. It has already been mentioned that the efficiency of the entropy of the volumetric features (packets and bytes number) in detecting DDoS attacks is widely demonstrated in many scientific papers. This type of attack consists of traffic from many source IP addresses toward one destination IP address, which is expressed by the N1-11, N1-1N, NN-11 and NN-1N models.

It is already highlighted that the entropy is changed due to a *spike* or *a long tail* in feature distribution. Taing the N1-1N model as an example, which relates to DDoS NTP amplification attacks [107], the destination IP address and the source port number are unique, so that the aggregation by this key ‘D.s’ can the most efficiently capture all of the attacker’s traffic, summarizing all belonging packets and bytes. Consequently, the element related to the victim's IP address and the source port used in the attack will make a significant spike at the top of the feature distribution, resulting, in turn, in decreased entropy. This is clearly demonstrated in Figure 6. 9 which presents the entropy of the source packet count feature aggregated by the destination IP address and the source port ‘sP[D.s]’. Increasing the amount of the anomalous traffic up to extreme values used in the model, cause a more dramatic drop in entropy values.

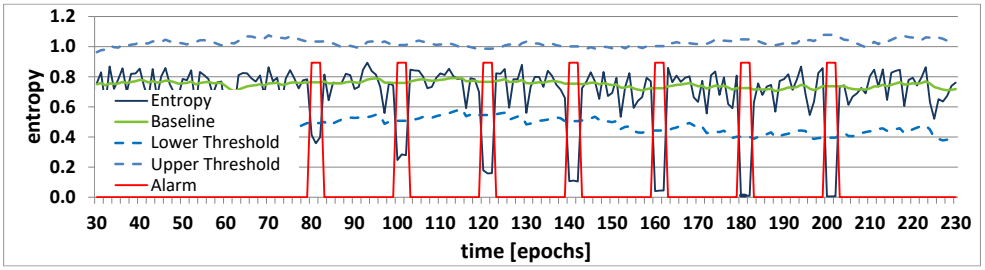

Figure 6. 9 The N1-1N model, the entropy of the source packet count feature aggregated by the destination IP address and destination port ‘sP [D.s]’.

The baseline of entropy is around 0.8, while the variation of the entropy values is noticeable, resulting in a higher standard deviation and a wider margin of tolerance of +/-0.2. This is the reason why

the first and the smallest anomaly (in epochs 60–62) remained undetected inside the margin of tolerance. There were several false positive alarms at the very beginning of the series since the standard deviation had not been stabilized. It is noteworthy that aggregation by the destination IP address only, which is a less specific key, yields similar entropy, but with a smaller standard deviation, resulting in a greater number of false positive alarms.

Another typical, but less frequent, type of entropy change during DDoS attacks is depicted in Figure 6. 10, which relates to the source packet count feature aggregated by the destination port 'sP[d]' in the same model. Since the destination port is used as an aggregation key, its values are randomized in the case of observed DDoS anomaly model N1-1N. In the corresponding distribution, this behavior produces 'a long tail' of elements with only one appearance, also known as ('a low activity region') in [69], resulting in higher evenness of the distribution and increased entropy. The average entropy was lower, around 0.3, and the standard deviation was less wide, but the smallest anomaly was still not detected, while some false positive alarms were triggered for the entropy below the lower margin.

This behavior of Shannon entropy is well reported in the literature [60][58], while some authors rather suggest parametrized entropy to capture this anomaly [44][69].

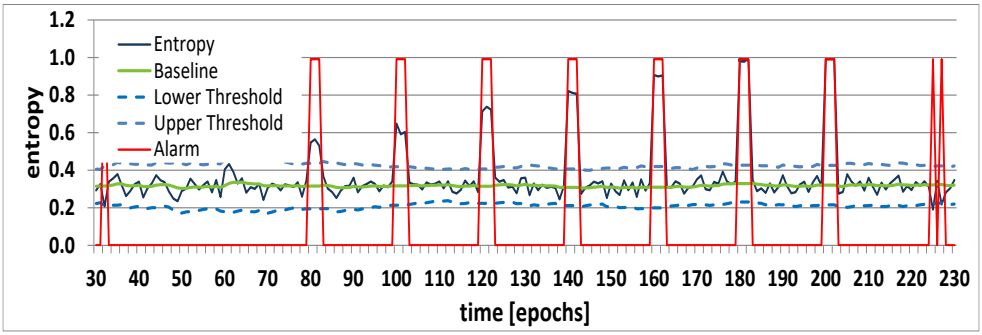

Figure 6. 10 The N1-1N model, the entropy of the source packet count feature aggregated by the destination port 'sP[d]'.

A similar analysis of the entropy of source packets count feature for each aggregation key and anomaly model yields interesting results, which are summarized in Table 6. 1. The label '-' denotes the entropy drop during the anomaly, while the label '+' denotes the entropy increase. The length of the labels describes the detection sensitivity of the observed features, where more characters reflect a higher sensitivity, while only one character indicates detection of only extremely intensive anomalies.

Table 6. 1 Entropy of the source packet count feature affected by anomaly models.

Volumetric features	Anomaly Module															
	1111	111N	11N1	11NN	1N11	1N1N	1NN1	1NNN	N111	N11N	N1N1	N1NN	NN11	NN1N	NNN1	NNNN
sP[S]	---	---	---	---	---	---	---	---	++	++	++	++	++	++	++	++
sP[D]	---	---	+	+	---	---	+	+	---	---	+	+	---	---	+	+
sP[s]	---	---	---	---					---	---	---	---				
sP[d]	---	++ +	---	++ +	---	++ +	---	++ +	---	++ +	---	++ +	---	++ +	---	++ +
sP[S.D]	---	---			---	---										
sP[S.s]	---	---	---	---												
sP[S.d]	---	+	---	+	---	+	---	+	+	+	+	+	+	+	+	+
sP[D.s]	---	---							---	---						
sP[D.d]	---	+	+	+	---	+	+	+	---	+	+	+	---	+	+	+
sP[s.d]	---		---						---		---					
sP[S.D.s]	---	---														
sP[S.D.d]	---				---											
sP[S.s.d]	---		---													
sP[D.s.d]	---								---							
sP[S.D.d.s]	---															

Even a brief look at the table reveals the following characteristics:

- The entropy of 'sP[D.s]' and 'sP[d]' for the N1-1N model, shown in the Figure 6. 9 and Figure 6. 10, are able to detect most of the generated anomalies, except the one with very small intensity, and, therefore, can be considered as a very sensitive, labelled with '---' and '+++' respectively.
- Entropies behave differently for different anomaly models, while some of them are not affected by a particular anomaly at all (the empty cells in the table).
- More importantly, the ways in which the entropies are affected by the modelled anomalies follow a very specific pattern. It can be observed that the entropy drops (labelled '-') occurs only when all features in the aggregation key have a single occurrence in the model (marked with '1'). For instance, aggregation by the source IP address will cause an entropy drop in the first 8 models, with labels starting with the character '1', since a single host as a source of the anomaly greatly contributes to the calculated distributions.
- On the other hand, if the anomaly is caused by too many source IP addresses (models with labels starting with 'N'), the distribution will be long and more even, increasing the entropy value.
- It should be noted that when the source port feature is used in the aggregation key, it can result only in the entropy drop, and not in an increase. The reason for this lies in the behavior of the regular network communication pattern, where a source host as a client initiates many connections to different servers using random source port numbers so that the corresponding distribution is already randomized.
- It is noteworthy that the 11-11 model is not realistic, since the NetFlow protocol treats all flows with the same source and destination IP addresses and port numbers as a single communication and will,

accordingly, generate only one flow record. However, this model is retained to fill the theoretical gap.

The explained periodic pattern leads to an important conclusion that each model has a unique footprint of triggered entropies. A similar analysis and conclusion also apply to the destination packet feature, as well as for the source and destination byte features. However, the synthetic traffic in all models is generated with a large number of packets and bytes, which is the case with DDoS and similar volume-intensive attacks, but still not realistic for many other attacks, such as a port scan or network scan. For this reason, the focus of our research is further oriented toward other features, namely the flow count and behavior features and the ways in which they are triggered by the anomaly models which are explained in more details in the next section.

6.5.2 Non-volumetric features

Since the volumetric features, such as the source and destination byte and packet counts, are efficient only for DDoS and similar volume-intensive attacks, and it is useless for many other attacks, such as Port Scan, Network Scan or Dictionary attack, we made other experiment using flow count and second-degree features, also called behavior features.

For the N1-1N model, wich relates to DDoS NTP amplification attacks [107] as explained in the previous section, the unique destination IP address and the source port number are suitable candidates for the aggregation key to capturing a *spike* in the distribution of other feature. On the other hand, the source IP address and destination port, which correspond to the label "N" in the N1-1N model, can be employed in the aggregate key to detecting a *long tail* of the distribution. These two representative situations are shown in Figure 6. 11 and Figure 6. 12 for the features ‘f[s]’ and ‘S[d]’, respectively, using the Shannon entropy.

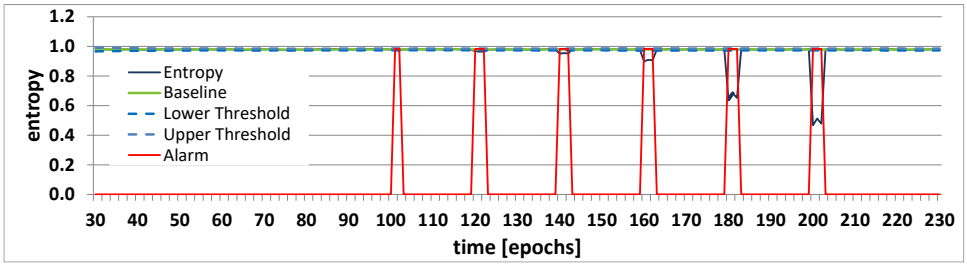

Figure 6. 11 The Shannon entropy and the N1-1N model - the flow count feature aggregated by the source port ‘f[s]’.

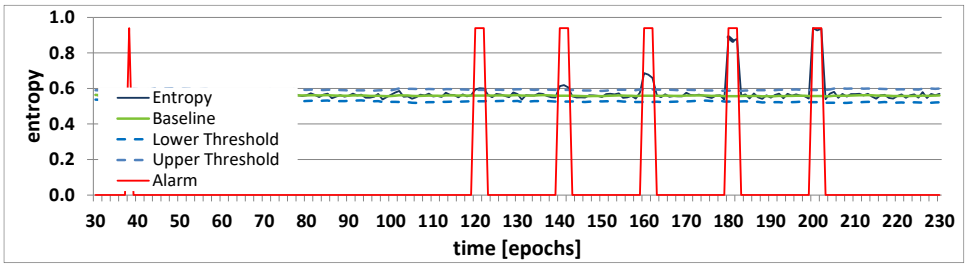

Figure 6. 12 The Shannon entropy and the N1-1N model - the source IP address feature aggregated by the destination port ‘S[d]’.

In the case of the feature ‘f[s]’ (Figure 6. 11), the entropy of normal traffic reaches its maximum value of 1, with a modest standard deviation, due to the natural unpredictability of the source port number in network communications, which is employed in the aggregation key. The unique combination of the victim IP address and the source port number utilized in many flows of the attack (UDP port number 123) causes a substantial spike in the flow count distribution, resulting in dramatically lower entropy. In the case of the feature S[d], the entropy values of ordinary traffic is lower (about 0.55), since the utilizing a highly randomized destination port number as an aggregation key results in numerous elements in the feature distribution with just one occurrence.

Figure 6. 11 shows how the high sensitivity of the feature ‘f[s]’ on the observed anomaly is used to further validate the entropy deception defense mechanism for low-rate attacks. The growth in distribution length caused by forged traffic to camouflage all entropy types of the f[s] feature, as shown in Figure 6. 13, is clearly detectable. Because the source port is highly randomized in typical network communications, only a deception of the smallest and hardly apparent anomaly around epoch 100 (with 25 synthetic flows alone) cannot be identified owing to the huge number of data elements in the distribution of the feature ‘f[s]’.

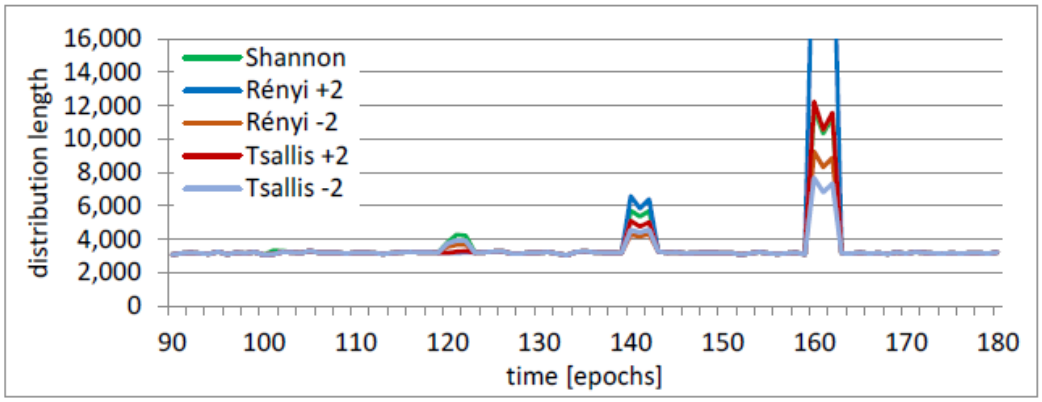

Figure 6. 13 Protection against entropy deception - the length of the ‘f[s]’ feature distribution with spoofed traffic applied on the N1-1N model.

For all anomaly models and the whole collection of behavior features, an extensive study and analysis of all entropy types were performed. Despite the fact that some researchers claim that the Rényi and Tsallis entropies have a greater detection capacity than Shannon entropy, our studies and experimental results show that this is not the case. Table 6. 2 shows the number of anomalies discovered by different entropy types for the most typical characteristics features using the previously mentioned base ground dataset with synthetically created 7 anomaly series and the N1-1N anomaly model as an example.

While the Shannon entropy surpasses certain other entropy types for the feature ‘S[s]’, other entropy types perform better in other circumstances and features. The difference in detection ability is only relevant to the tiniest abnormalities in all circumstances, whereas all entropy types correctly recognized all anomalies of moderate to high intensity. All the results confirm that choosing the proper

entropy type is a difficult process that should take into account a number of factors, including specific network traffic and its diversity and deviations, entropy change detection approach, and previously examined resilience to deception.

Table 6. 2 Number of detected anomalies in N1-1N model by different entropy types.

N11N	Shannon	Renyi +2	Renyi -2	Tsallis +2	Tsallis -2
f[S]	3	2	4	3	3
S[D]	3	2	4	2	4
d[D]	6	6	6	6	6
S[s]	6	6	5	5	6
d[s]	5	4	5	4	6
S[d]	5	3	4	3	4
f[S.d]	3	2	3	3	3
S[D.s]	6	6	5	5	6

Another experiment using the flow count feature for the N1-1N anomaly model shows that the aggregation by the source IP address ‘f[S]’ results in increased entropy (Figure 6. 14), while the aggregation by the destination IP address ‘f[D]’ decreases entropy (Figure 6. 15). Similarly, to the previously explained entropy of the packet feature, due to the anomaly, the source IP addresses are more evenly spread over the distribution, while all flows to the destination IP address are concentrated into a single element. In both cases, only the most intensive anomalies are detected. The reason for this is the relatively small average entropy (around 0.65 and 0.45), which means that the distribution is rather uneven with a spike at the beginning and a long tail at the end, so the anomaly must contribute much more in order to be detected.

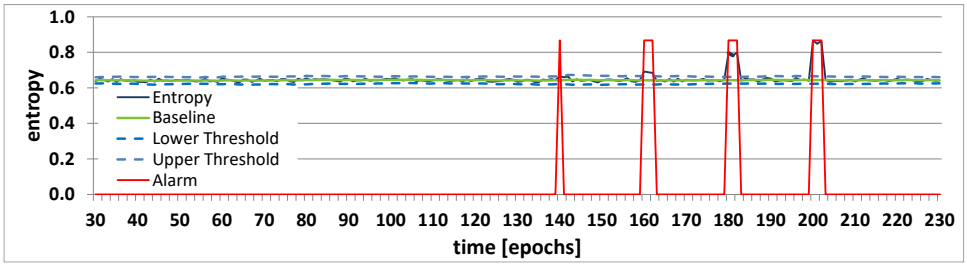

Figure 6. 14 The N1-1N model, the entropy of the flow count feature aggregated by the source IP address ‘f[S]’.

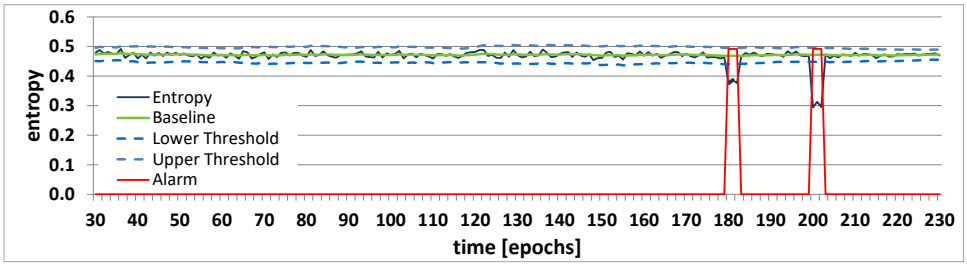

Figure 6. 15 The N1-1N model, the entropy of the flow count feature aggregated by the destination IP address 'f[D]'.

Anomaly detection efficiency can be further improved using a more specific aggregation key, as shown in Figure 6. 16 for a combination of a destination IP address and a source port 'f[D.s]'. In this case, regular traffic distributes the aggregated elements more evenly, resulting in the entropy value near the maximum limit of 1, with a very small standard deviation and a barely noticeable margin of tolerance. Under these conditions, the anomaly of the N1-1N type easily makes a spike in the distribution, due to which even a small decrease in entropy triggers an alarm.

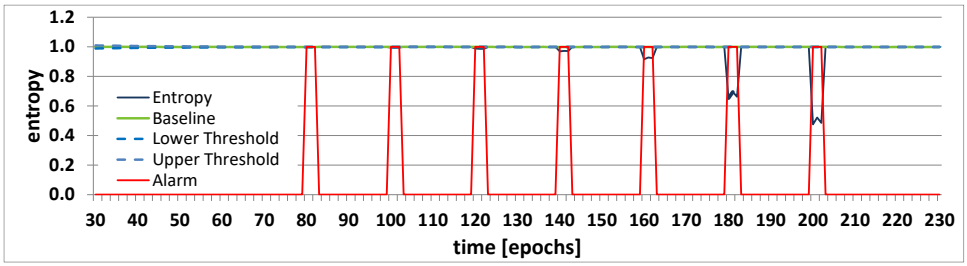

Figure 6. 16 The N1-1N model, the entropy of the flow count feature aggregated by the destination IP address and source port 'f[D.s]'.

In addition to the flow count feature, the second-degree features can capture specific anomalies more efficiently, even for less specific aggregation keys, such as the entropy of the destination port feature aggregated by the destination IP address 'd[D]' shown in Figure 6. 17.

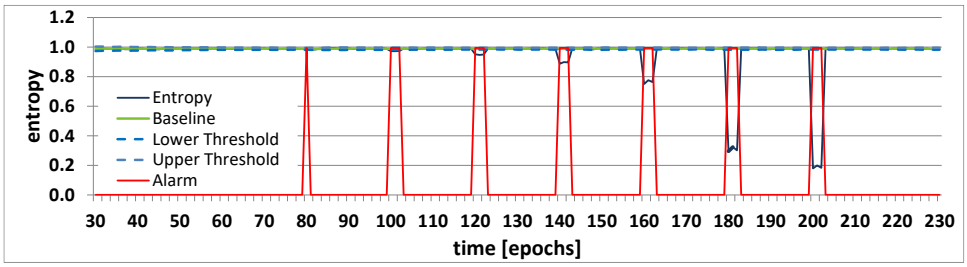

Figure 6. 17 The N1-1N model, the entropy of the destination port feature aggregated by the destination IP address 'd[D]'.

Anomaly detection based on the entropy of second-degree features provides more details about the communication pattern regardless of traffic volume expressed by packets and bytes, which makes them equally efficient for any type of anomaly and applicable in real enterprise networks.

A port scans attack, described by the 1N-1N model, or a network scan modelled with 1N-N1 or 11-N1, uses small traffic volume to reach a large number of destination ports and/or IP addresses. Figure 6. 18 demonstrates the entropy drop for the destination port as a second-degree feature aggregated by the source and destination IP address 'd[S.D]', while Figure 6. 19 shows the entropy increase for a distinct number of source IP addresses per destination port 'S[d]'. A small standard deviation can also make false positive alarms, but they can be eliminated by better tuning or using more advanced detection techniques.

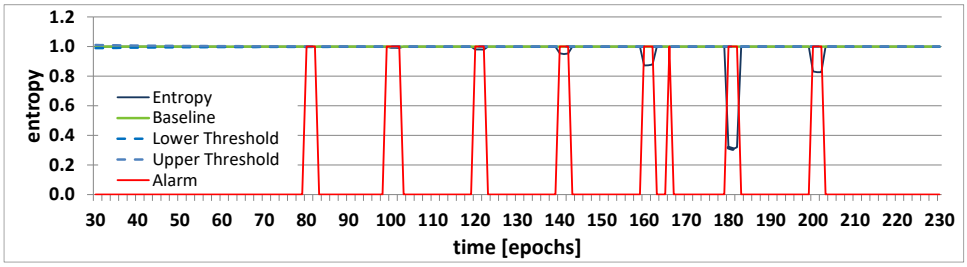

Figure 6. 18 The 1N-1N model, the entropy of the destination port feature aggregated by the source and destination IP address 'd[S.D]'.

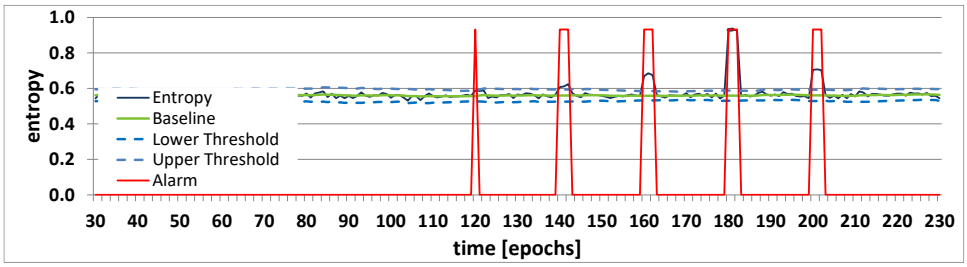

Figure 6. 19 The 1N-1N model, the entropy of the source IP address feature aggregated by the destination port 'S[d]'.

The results describing how the entropies of the flow count and second-degree features are affected by each anomaly model are summarized in Table 6. 3, using the same labelling convention for entropy changes and detection sensitivity described in Table 6. 1. The fundamental reason for the distinct periodic pattern, shown in the table, is the mentioned behavior, i.e., how different anomaly models impact the entropy of different attributes. This also explains why some features are extremely successful in identifying some anomalies while being absolutely useless in detecting others. This conclusion is similar to what we found when using volumetric features (Table 6. 1) and similar to that of prior studies [69][111][119], but it encompasses the whole feature set as well as all anomaly models. Furthermore, it demonstrates that for specific anomalous models, behavior features or sophisticated aggregation keys surpass the typically used flow count feature of basic flow attributes.

The authors of [119], for example, categorize a DDoS attack by noticing a drop in entropy in the flow count of the destination IP address and port number. This relates to our NN-11 anomaly model, which has features ‘f[D]’ and ‘f[d]’, which are only sensitive to the most severe anomalies, whereas behavior feature S[D] can identify less severe abnormalities (up to 10 times in our experiment). The difference in metric performance is especially noticeable in the case of a port scan attack, which is specified by the 11-1N model (using a fixed source port) against the 1N-1N model (using a random source port). In both situations, the best results are obtained by employing the behavior attributes ‘d[S]’, ‘d[D]’, and ‘d[S.D]’, which are labelled with a '---' in Table 6. 3.

Table 6. 3 Entropy changes of the flow count and second-degree (behavior features) affected by the anomaly models.

Features		Anomaly Module															
Behavior	Flow Count	11-11	11-1N	11-N1	11-NN	1N-11	1N-1N	1N-N1	1N-NN	N1-11	N1-1N	N1-N1	N1-NN	NN-11	NN-1N	NN-N1	NN-NN
D[S]				---	---			---	--	++	++	++	++	++	++	++	++
s[S]						-	-	-	-	++	+	++	+	+	++	+	++
d[S]			---		---		----		---								
	f[S]	-	-	-	-	-	-	-	-	+	+	+	+	+	++	+	+
S[D]				++	++			++	++	--	--	++	++	-	--	++	++
s[D]				++	++			++	++	--		++	++			++	++
d[D]			---				---				---				---		
	f[D]			++	++			++	++			++	++			++	++
S[s]										---	---	---	---				
D[s]				---	---							---	---				
d[s]			---		---						--		---				
	f[s]	---	---	---	---					---	--	---	---				
S[d]			+++		+++		+++		+++	-	++	-	+++	-	+++	-	+++
D[d]			+++	--	+++		+++	--	+++		+++	--	+++		+++	--	+++
s[d]			+++		+++	++	+++		+++		++		+++	++	+++	++	+++
	f[d]	++	+++	++	++	++	++	++	++	++	++	++	+++	++	++	++	+++
s[S.D]				+	+	-		+	+	+	+	+	+	+	+	+	+
d[S.D]			---				---										
	f[S.D]	-	-	+	+	-	-	+	+	+	+	+	+	+	+	+	+
D[S.s]				---	---												
d[S.s]			---		---												
	f[S.s]	---	---	---	---												
D[S.d]			+	---	+		+	---	+	+	+	+	+	+	+	+	+
s[S.d]			++		++	-	+	-	++	+	++	++	++	++	++	++	++
	f[S.d]	-	++	-	++	-	++	-	++	+	++	++	++	++	++	++	++
S[D.s]										---	---						
d[D.s]			---								---						
	f[D.s]	---	---							---	---						
S[D.d]			++	++	++		++	++	++	--	++	++	++	--	++	++	++
s[D.d]			++	++	++		++	++	++		++	++	++		++	++	++
	f[D.d]		++	++	++		++	++	++		++	++	++		++	++	++
S[s.d]										---		---					
D[s.d]				---								---					
	f[s.d]	---		---						---		---					
d[S.D.s]			---														
	f[S.D.s]	---	---														
s[S.D.d]			+	+	+	-	+	+	+	+	+	+	+	+	+	+	+
	f[S.D.d]	-	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+
D[SA.s.d]				---													
	f[S.s.d]	---		---													
S[D.s.d]										---							
	f[D.s.d]	---								---							
	f[S.D.d.s]	---															

In this case, a periodic pattern of entropy behavior is more sophisticated as a consequence of using the identification data as a second-degree feature. It can be generally concluded that higher detection sensitivity is achieved when a feature in the aggregation key relates to the lable ‘1’ and the second-degree feature relates to the lable ‘N’, such as the ‘d[S.D]’ feature in the1N-1N model, previously shown in Figure 6. 18.

This recurring pattern in feature sensitivity to various anomalies leads to the critical finding that every anomaly type has a distinct footprint of triggered entropies, which is employed in the formulation of classification rules, as described later in the text. Table 6. 4 summarize the attacks, aggregation and modules, mapping for each attack.

Table 6. 4 Attacks, aggregation and modules, mapping.

No.	Attack	Description	Module - aggregation	Aggregation features	Figure No.
1	**DDoS attacks.** This type of attacks consists of traffic from many source IP addresses toward one destination IP address, which is expressed by the N1-11, N1-1N, NN-11 and NN-1N models.	Despite larger standard deviation, sudden entropy drop is obvious during the attack as a consequent increased flow number between the attacker and the victim hosts.	**1N-11 f[S.D]**	**Flow-count feature**	Figure 6.5
		In the N1-1N model, which relates to DDoS NTP amplification attacks, the destination IP address and the source port number are unique, so that the aggregation by this key ('D.s') can the most efficiently capture all of the attacker's traffic, summarizing all belonging packets and bytes. Consequently, the element related to the victim's IP address and the source port used in the attack will make a significant spike at the top of the feature distribution, resulting, in turn, in decreased entropy.	**N1-1N sP[D.s]**	**Volumetric feature**	Figure 6.9
		Another type of entropy change during DDoS attacks which is related to the source packet count feature aggregated by the destination port. Since the destination port is used as an aggregation key, its values are randomized in the case of observed DDoS anomaly model N1-1N. In the corresponding distribution, this behavior produces 'a long tail' of elements with only one appearance ('a low activity region'), resulting in higher randomness in the distribution and increased entropy.	**N1-1N sP[d]**	**Volumetric feature**	Figure 6.10
		The destination IP address and the source port number are unique throughout the attack and are suitable candidates for the aggregation key to capturing a *spike* in distribution.	**N1-1N f[s]**	**Flow-count feature**	Figure 6.11
		The source IP address and destination port, which correspond to the label "N" in the N1-1N model, can be employed in the aggregate key to detecting a *long tail* of the distribution.	**N1-1N S[d]**	**Second-degree (Behavior feature)**	Figure 6.12
		The aggregation by the source IP address results in increased entropy.	**N1-1N f[S]**	**Flow-count feature**	Figure 6.14
		The aggregation by the destination IP address decreases entropy.	**N1-1N f[D]**	**Flow-count feature**	Figure 6.15
		Anomaly detection efficiency can be significantly improved using a more specific aggregation key, as a combination of a destination IP address and a source port.	**N1-1N f[D.s]**	**Flow-count feature**	Figure 6.16
		The second-degree features can capture specific anomalies more efficiently, even for less specific aggregation keys, such as the entropy of the destination port feature aggregated by the destination IP address.	**N1-1N d[D]**	**Second-degree (Behavior feature)**	Figure 6.17

No.	Attack	Description	Module - aggregation	Aggregation features	Figure No.
2	**PortScan attacks.** Occurs when an attacker attempts to connect to a large number of target ports on a remote victim system in order to uncover vulnerabilities. When a single source port is used during this process the attack is described by the '11-1N' communication pattern, while using multiple source ports falls into the '1N-1N' class. **Network scan** modelled with 1N-N1 or 11-N1, use a small traffic volume to reach a large number of destination ports and/or IP addresses.	The destination port behavior feature aggregated by the source and destination IP addresses 'd[S.D]', generates more randomness distribution with an entropy value near the maximum value (1), so the normalized entropy can much more efficiently distinguish PortScan attack from regular network behavior.	**11-1N d[S.D]**	**Second-degree (Behavior feature)**	Figure 6.6
		The entropy drops for the destination port as a second-degree feature aggregated by the source and destination IP address	**1N-1N d[S.D]**	**Second-degree (Behavior feature)**	Figure 6.18
		The entropy increases for a distinct number of source IP addresses per destination port.	**1N-1N S[d]**	**Second-degree (Behavior feature)**	Figure 6.19
3	**Infiltration attack**	The destination port behavior feature, aggregated by the source IP addresses 'd[S]'.	**11-1N d[S]**	**Second-degree (Behavior feature)**	Figure 6.7
4	**Brute force attack**	The attack was generated using SSH-Patator tool in its part named "Tuesday afternoon". The attack is similar to intensive but regular traffic of web and DNS services. It was performed with less magnitude than total regular traffic, with no significant changes in entropy values. flow partitioning comes into play to separate large partitions of regular traffic from the rest.	**1N-11 f[d]**	**Flow-count feature**	Figure 6.8
5	**Detection of minor anomalies in background traffic**	Entropy applied on the partition of the traffic, filtered by the protocol field, reveals new anomalies in different epochs. For TCP traffic only, the entropy of the destination port behavior feature aggregated by the source IP addresses 'd[S]' is shown in	**TCP only d[S]**	**Second-degree (Behavior feature)**	Figure 6.22
		While the entropy for the ICPM traffic using only the flow count feature aggregated by the source and the destination IP addresses. These fewer intensive anomalies were masked by the total traffic, but taking only a smaller portion of the traffic into account, the entropy changes become obvious and relevant.	**ICPM only** f[S.D]	**Flow-count feature**	Figure 6.23

6.6 Anomaly model classification rules

Previous results also confirm that each anomaly model has a unique signature in terms of the effect on entropy. Due to feature correlation, also pointed out in other research [14][58] not all of them are necessary for the unique identification of anomaly models. It is possible to minimize the set of features while keeping uniqueness in model recognition and classification. This could be done in several different ways so that the following principles can be used to select the optimal rules for anomaly model recognition:

- Prefer the most sensitive features ('---' or '+++');
- Prefer features affected by the minimal number of models;
- Prefer features with a simple aggregation key;
- Use the 'Not affected' rule for a feature to differentiate its behavior from another model (empty cells in the table);
- Use the 'Not decrease/increase' rule for a feature to differentiate its behavior from another model (make a difference between '-' and '+' cells in the table)
- Select model identified by the smallest number of unique features first, then proceed with others.

By applying these principles to the results in Table 6. 3, the following anomaly model identification rules can be extracted in Table 6. 5.

Table 6. 5 Anomaly models identification and classification rules.

Anomaly Model	Affected (Decreased)	Affected (Increased)	Not Affected	Not Decreased
11-11	f[D.s] S[D.s]		d[D.s]	
11-1N	f[s], d[S.D]			
11-N1	D[s.d]		S[s.d]	
11-NN	d[s], D[S.s]			
1N-11	f[S.D]		d[S.D]	
1N-1N	d[S.D]		f[s]	
1N-N1	D[S.d]		D[s.d]	
1N-NN	D[S], d[S]		d[s]	
N1-11	S[D.s]		d[D.s]	
N1-1N	S[D.s], d[D.s]			
N1-N1	S[s.d], D[s.d]			
N1-NN	S[s], D[s], d[s]			
NN-11	s[D]		d[D]	d[S]
NN-1N	d[D]		S[s]	d[S]
NN-N1	S[d], D[d]		S[s]	
NN-NN		d[S], S[d]	S[s]	

The rules defined above include a minimal set of features, which is important for performance optimization since aggregation is a CPU and memory consuming process. However, in an anomaly detection process, it is useful to keep more features, even if they are correlated, in order to minimize false alarms. Our methodology leaves room for further improvement of the detection efficiency using more

advanced detection techniques, such as fuzzy logic, machine learning or neural network, which is out of the scope of this research.

6.7 The validation of the classification rules

The classification ability and usefulness of the methodology presented in this research are demonstrated on real network data taken from the dataset CTU-13. More precisely, data capture named '43' from CTU-13 dataset was used, where botnet traffic was excluded from the dataset, keeping a large portion of real-life background traffic with several anomalies of smaller intensity. Using only the flow count and second-degree (behavior) features, the most characteristic results are presented below.

The entropy of the flow count feature aggregated by the source IP addresses 'f[S]', shown in Figure 6. 20, reviles several smaller anomalies, including some minor deviations which are possible false positive alarms.

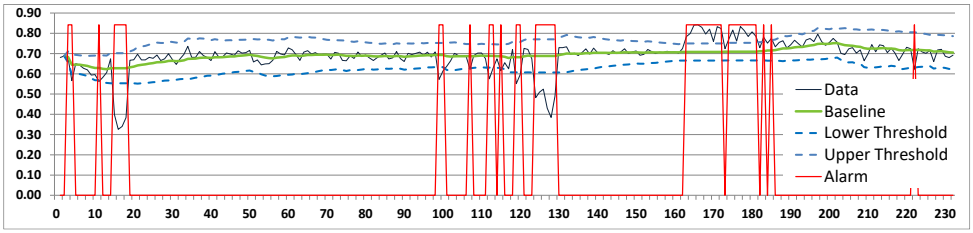

Figure 6. 20 CTU-13 dataset, capture 43, regular traffic: feature 'f[S]'.

Observing entropy of other features, such as the destination port behavior feature with the same aggregation key 'd[S]', illustrated in Figure 6. 21, part of the anomalies is disappeared, indicating the presence of different anomaly types in the traffic over time.

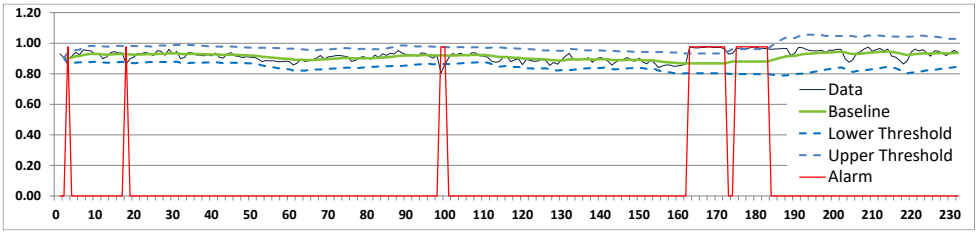

Figure 6. 21 CTU-13 dataset, capture 43, regular traffic: feature 'd[S]'.

On the other hand, entropy applied on the partition of the traffic, filtered by the protocol field, reveals new anomalies in different epochs. For TCP traffic only, the entropy of the destination port behavior feature aggregated by the source IP addresses 'd[S]' is shown in Figure 6. 22, while the entropy for the ICPM traffic using only the flow count feature aggregated by the source and the destination IP addresses 'f[S.D]' are shown in Figure 6. 23. These less intensive anomalies were masked by the total traffic, but taking only a smaller portion of the traffic into account, the entropy changes become obvious and relevant. Small entropy deviation around epochs 100 in total traffic was barely noticeable in (Figure 6. 21) and subject to suspicion of false positive alarm until it is analysed for TCP traffic only (Figure 6.

22). These cases clearly demonstrate the proposed flow partitioning method as a simple approach to achieving better detection sensitivity and efficiency.

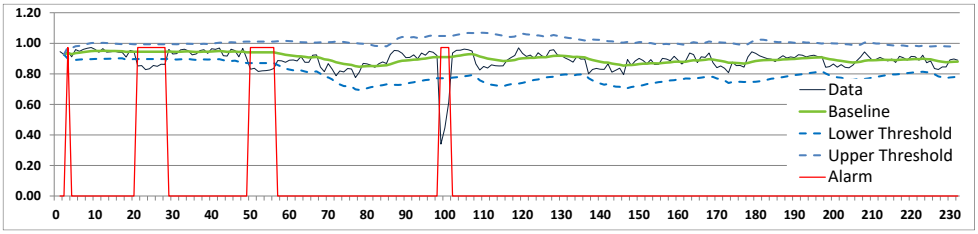

Figure 6. 22 CTU-13 dataset, capture 43, regular traffic, TCP only: feature 'd[S]'.

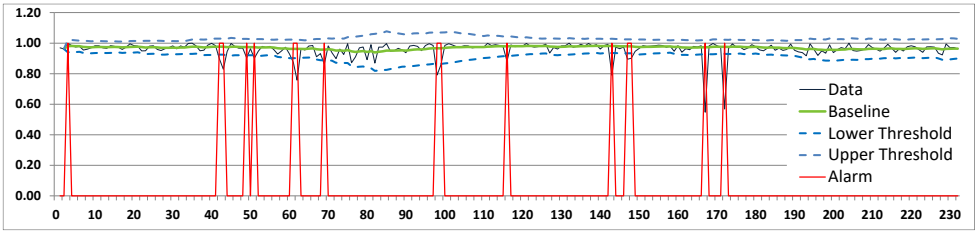

Figure 6. 23 CTU-13 dataset, capture 43, regular traffic, ICMP only: feature 'f[S.D]'.

The results of entropy analysis using the proposed anomaly model identification rules in different epochs for the most severe anomalies are presented in Table 6. 6. All detected anomalies follow the unique signature presented in Table 6. 3 and can be classified by the developed rules. Only the TCP anomaly in epochs 51–57 presents a combination of two similar models: 1N-1N and 1N-11. A drill-down raw data analysis has confirmed that the anomaly consists of flows with a larger number of distinct destination ports according to the 1N-1N model, while one of them occurs more frequently, following the 1N-11 model.

Table 6. 6 Verification of anomaly classification rules in real network traffic.

Traffic	All	All	All	TCP	TCP	TCP	TCP	ICMP	ICMP
Epochs	15-19	124-130	163-180	21-29	51-57	99-102	170-171	167-168	172-173
Anomaly	1N-11	1N-11	NN-11	1N-1N	1N-11 1N-1N	1N-1N	11-N1	1N-11	1N-1N
D[S]							---		
s[S]	---	--	+		---	-			
d[S]			+	--	--	---			---
f[S]	---	---	+		---	--			
S[D]			--						
s[D]	-	-	-	-	---	---			
d[D]				---	---	---			---
f[D]	---	-	-	-	---	---		--	---
f[s]							---		
S[d]			--			++			
D[d]						++	--		
s[S.D]	---	--			---	---			
d[S.D]				---	---	---			
f[S.D]	---	---		-	---	---		---	---
D[S.s]							---		
D[S.d]							---		
s[S.d]	---	---			---				
f[S.d]	---	---			---		-	---	
S[D.d]			--						
S[s.d]									
D[s.d]							---		

6.8 Results discussion and comparison with machine learning approaches

Entropy-based network traffic anomaly detection in many aspects completely differs from the machine learning methods, which makes their performances hard or even impossible to directly compare. For that reason, we rather discuss their general characteristics and leave a decision on which one is better for specific use-cases. The main difference lies in the fact that entropy-based detections work on a time interval level, detecting an anomaly in an epoch, while machine learning detection methods provide detection granularity on the data level, classifying each data point as normal or anomalous. This fundamental difference implies the following consequences:

- Anomalies detected using an entropy-based approach requires further root-cause analysis to extract the information about the attackers, victims and services used.
- The entropy-based approach does not require a training process with a labelled dataset, which is the case with supervised machine learning, which makes it attractive for general purpose application in a real-live network with any kind of traffic unknown in advance.
- The entropy-based approach requires less processing power than most of the other techniques, which makes it attractive for real-time application.
- Performance metrics used in machine learning (Accuracy, Precision, Recall, ROC curve etc.) take into account individual labelled data, and therefore they are impractical for usage in an entropy-based approach.

As previously stated, the motivation behind our research in network behavior analysis was to propose both anomaly detection and classification methods, applicable for practical usage in a general network environment. Entropy-based approach was chosen having in mind the above-mentioned characteristics. Anomaly detection is based on data obtained by NetFlow or similar protocols, which are the industry standards and the most convenient way to gather information about network traffic structure. NetFlow data, collected from network routers, provides only basic information about communication peers (IP addresses, protocol and port numbers), duration and total bytes and packets transferred. Enriched with the flow count and behavior features obtained by the aggregation process, this basic information appears to be sufficient for entropy calculation. Our experimental results confirm that this approach is efficient when a traffic structure is significantly changed during the attack, while it is useless for other attacks whose communication characteristics cannot be distinguished from regular traffic. This was the case with web attacks in the CICIDS2017 dataset named “Thursday morning”, since its communication behavior is the same as the regular web traffic and cannot be recognized as an anomaly.

On the other hand, machine learning approaches on network behavior analysis rely on more communication details, such as TCP flags and window size, packet length, packet inter-arrival time, jitters and their statistical parameters (average, min, max, standard deviation). Obtaining these data is based on processing raw traffic on the packet level, which requires direct access to network traffic and demanding data processing, especially for real-time applications.

The CICIDS2017 dataset was generated in this way and the authors originally used it for anomaly detection based on supervised machine learning [117]. For each simulated attack, they have achieved very high detection performances using various features and the following metrics: Precision (Pr - the ratio of the correctly detected attacks to all triggered alarms), Recall (Rc - the ratio of the correctly detected attacks to all attacks), and F-Measure (F1 - the harmonic mean of the Precision and Recall).

We have reproduced their experiment with the same dataset "Thursday morning" with web attacks, which consists of Brute Force, Cross Site Scripting (XSS) and SQL Injection attacks. We have also used Random Forest (RF), Multilayer Perceptron (MLP), and Naive-Bayes (NB) machine learning algorithms, and the same features used by the authors in [117], namely the initial TCP window size in both directions and the total bytes transferred from the source to destination.

In our reproduced experiments in Weka software, using 70% of training and 30% of testing data randomly chosen from the dataset, we generally confirmed their results, especially in terms of Recall performance metrics. Furthermore, we performed a deeper investigation of raw data, which reviled that most of the attack flows used an initial TCP window size of 29,200 and 28,960 bytes from the source and destination directions respectively. Since the TCP window can take an arbitrary value even in attack communications, we wanted to check the detection capability of the machine learning algorithms if these values were changed. For that reason, we increased the initial TCP windows of attack flows in the testing dataset by 3%, 10% and 30% and repeated the experiments. From Table 6. 7, which summarises the results, it is obvious that the Random Forest algorithm dramatically loses a detection capability even with small changes of 3%, while the Multilayer Perceptron algorithm was not able to detect any attack at all. Only the Naive-Bayes algorithm is more resilient to the initial TCP windows value changes, but it achieves the lowest performances.

Table 6. 7 Supervised machine learning performance evaluation.

Algorithm	Dataset	Precision	Recall	F1
RF	Original	0.850	0.981	0.911
	Modified, 3%	0.176	0.037	0.061
	Modified, 10%	0.176	0.037	0.061
	Modified, 30%	0.176	0.037	0.061
MLP	Original	0.771	0.840	0.804
	Modified, 3%	0.000	0.000	N/A
	Modified, 10%	0.000	0.000	N/A
	Modified, 30%	0.000	0.000	N/A
NB	Original	0.132	0.909	0.230
	Modified, 3%	0.132	0.908	0.230
	Modified, 10%	0.123	0.842	0.215
	Modified, 30%	0.123	0.842	0.215

The above example demonstrates that some machine learning algorithms, which are based on such specific values, can be easily spoofed with just a small variation in the attack scenario. Rather than just present pure performance measurement, we suggest further analysing of raw data and the meaning of the features in the context of the applied machine learning algorithms.

7. CONCLUSIONS

In this study, a new architecture of network traffic anomaly detection system has been proposed based on the novel method of multivariate analysis of the entropy changes of multiple features derived from basic NetFlow data only. The method is backed up by a unique safety mechanism against entropy detection being deceived. An important objective for the proposed solution was the feasibility for practical implementation in the general network environment. For this reason, only basic flow features have been chosen because they can be easily collected from network routers using NetFlow protocol or similar industrial standards. We address this issue by providing a systematic methodology with the main novelty in anomaly classification based on entropy of flow count and behavior features of data obtained by NetFlow protocol. We also proposed data partitioning for greater efficiency in real-time anomaly detection. Through an analysis of the most prominent security attacks, a generalized network behavior models were developed to describe various communication patterns. Based on a multivariate analysis of entropy changes in each of the modelled classes, experiments were used to design and test anomaly classification criteria.

The main achievements and conclusions made by the conducted research and the experimental results are the following:

- We compared the responses of the Shannon, Tsallis, and Rényi entropies to changes in feature distribution produced by spoofed traffic inserted to deceive entropy detection systems. The total number of elements in a distribution, also known as the distribution length, has been demonstrated to be an effective indicator for detecting entropy deception techniques.
- We defined and expanded the idea of aggregation and behavior features, making it adaptable to new entropy domain features. The experiments revealed that behavior features, followed by the flow count feature, perform the best. The packet and byte features are only useful for detecting heavy traffic loads; they are ineffective for detecting a wide variety of deviations produced by various security concerns.
- 16-anomaly models were created, based on these aggregation and behavior attributes, which are linked to a variety of security threats. For all anomaly models, extensive experiments were carried out with the whole feature set, computing the Shannon, Tsallis, and Rényi entropies with both positive and negative parameters, a comparison of the ways how all these entropies respond to the changes in feature distribution caused by spoofed traffic injected to deceive the entropy detection system is studied.
- It is shown that there is no large difference in anomaly detection capabilities between the parameterized Tsallis and Rényi entropies and the Shannon entropy, contrary to popular opinion. The proper entropy type depends on the specific network traffic, its variety and variations, the features that are employed, and other factors and qualities, such as deception resistance.

- The aspects that boost detection performance are included in the suggested high-level design. by providing flow partitioning method. This enables improved profiling and separation of specific abnormalities. The reported experimental results substantiate this hypothesis.
- Based on the extensive experimental findings and analysis, we suggest that the supervised machine learning algorithms applied for real-time network behavior analysis have more drawbacks than advantages in terms of practical application.
- It has been demonstrated that each anomalous model leaves a distinct behavioral footprint, revealing how various attributes' entropies are altered. The unique anomaly classification rules have been established based on the complete experiments and the suggested multivariate analysis, which is the key innovative contribution offered in this research, filling the gap in the current methodologies. The effectiveness of the anomaly classification approach is verified by the experimental findings reported.
- The supervised machine learning approaches utilized for network behavior analysis had substantial limits for efficient real-time application. As a result, the presented approach based on the entropy of the fundamental flow data appears to be more practical and widely applicable.

Despite the fact that there are numerous studies on this subject, we feel that our study contributes to a better understanding of entropy-based network behavior analysis and anomaly detection in a variety of ways.

The developed technique is based on Shannon entropy, but it is also open to parametrized Renyi and Tsallis entropy implementation and expandable through multivariate analysis automation, with the goal of improving detection and classification efficiency.

The extensive usage of feature-based anomaly detection systems necessitates ongoing efforts to reduce the number of costly false alarms. As a result, a potential focus of future work can be directed to unsupervised machine learning in more accurate multivariate analysis of the entropy results, as well as evaluating the idea and performances in a variety of real-time network situations. This involves both a traditional strategy that relies on an external data gathering and processing device and data plane programmability approaches on current software-defined networking architecture.

8. BIBLIOGRAPHY

[1] A. Elsadai, J. Ibrahim, F. Hajjaj, and P. Jakić, “The Overview of Intrusion Detection System Methods and Techniques,” pp. 155–161, 2019.

[2] J. Ibrahim and S. Gajin, “SDN-Based Intrusion Detection System Literature review,” Infoteh-Jahorina, vol. 16, no. March, pp. 621–624, 2017.

[3] D. E. Denning, 1987. "An Intrusion-Detection Model. IEEE Transactions on Software Engineering, ", SE-13(2), pp.222-232.

[4] A. Sperotto, 2010. “Flow-Based Intrusion Detection,”. Doctors of science. Centre for Telecommunication and Information Technology University of Twente, Italy.

[5] Brauckhoff Daniela, 2010. “Network traffic anomaly detection and evaluation, ”. Doctoral Thesis, ETH ZURICH.

[6] G. Giacinto, F. Roli, and L. Didaci, “A modular multiple classifier system for the detection of intrusions in computer networks,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 2709, pp. 346–355, 2003.

[7] G. G. Liu, “Intrusion detection systems,” Appl. Mech. Mater., vol. 596, pp. 852–855, 2014.

[8] V. Chandola, Arindam Banerjee, and Vipin Kumar “Anomaly Detection for Discrete Sequences: A Survey,” 2009.

[9] S. J. Han and S. B. Cho, “Detecting intrusion with rule-based integration of multiple models,” Comput. Secur., vol. 22, no. 7, pp. 613–623, 2003.

[10] T. Özyer, R. Alhajj, and K. Barker, “Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening,” J. Netw. Comput. Appl., vol. 30, no. 1, pp. 99–113, 2007.

[11] A. Patcha and J. M. Park, “An overview of anomaly detection techniques: Existing solutions and latest technological trends,” Comput. Networks, vol. 51, no. 12, pp. 3448–3470, 2007.

[12] C. H. Tsang, S. Kwong, and H. Wang, “Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection,” Pattern Recognit., vol. 40, no. 9, pp. 2373–2391, 2007.

[13] J. Mazel, R. Fontugne, and K. Fukuda, “A taxonomy of anomalies in backbone network traffic,” IWCMC 2014 - 10th Int. Wirel. Commun. Mob. Comput. Conf., pp. 30–36, 2014.

[14] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang, “An empirical evaluation of entropy-based traffic anomaly detection,” Proc. ACM SIGCOMM Internet Meas. Conf. IMC, pp. 151–156, 2008.

[15] Hjp.at. 2022. hjp: doc: RFC 3954, "Cisco Systems NetFlow Services Export Version 9, ". [online] Available at: <https://www.hjp.at/doc/rfc/rfc3954.html> [Accessed 23 January 2022].

[16] Pras, A., Sadre, R., Sperotto, A., Fioreze, T., Hausheer, D. and Schönwälder, J., 2009. “Using NetFlow/IPFIX for Network Management, ”. Journal of Network and Systems Management, 17(4), pp.482-487.

[17] Kb.juniper.net,2022.[Online].Available: https://kb.juniper.net/library/CUSTOMERSERVICE/GLOBAL_JTAC/BK16677/3500204-en.pdf. [Accessed: 23- Jan- 2022].

[18] B. Claise, Cisco Systems NetFlow Services Export Version 9, RFC 3954

[19] M. Grill, 2016. “Combining Network Anomaly Detectors,”. Doctors of science. Czech Technical University in Prague-Faculty of Electrical Engineering, Czech.

[20] GHANEM, W., 2019. METAHEURISTIC-BASED NEURAL NETWORK TRAINING AND FEATURE SELECTOR FOR INTRUSION DETECTION. PhD. Universiti Sains Malaysia (USM) - School of Computer Sciences.

[21] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," Comput. Secur., vol. 70, no. October, pp. 255–277, 2017.

[22] B. Senthilnayaki, K. Venkatalakshmi, and A. Kannan, "An intelligent intrusion detection system using genetic based feature selection and Modified J48 decision tree classifier," 2013 5th Int. Conf. Adv. Comput. ICoAC 2013, no. December, pp. 1–7, 2014.

[23] D. A. Kumar and S. R. Venugopalan, "Intrusion detection by initial classification-based on protocol type," Int. J. Adv. Intell. Paradig., vol. 9, no. 2/3, p. 122, 2017.

[24] A. Özgür and H. Erdem, "The impact of using large training data set KDD99 on classification accuracy," PeerJ, vol. 5, 2017.

[25] Y. Abuadlla, G. Kvascev, S. Gajin, and Z. Jovanović, "Flow-based anomaly intrusion detection system using two neural network stages," Comput. Sci. Inf. Syst., vol. 11, no. 2, pp. 601–622, 2014.

[26] N. Hoque, M. Singh, and D. K. Bhattacharyya, "EFS-MI: an ensemble feature selection method for classification," Complex Intell. Syst., vol. 4, no. 2, pp. 105–118, 2018.

[27] V. Timčenko and S. Gajin, "Ensemble classifiers for supervised anomaly based network intrusion detection," Proc. - 2017 IEEE 13th Int. Conf. Intell. Comput. Commun. Process. ICCP 2017, pp. 13–19, 2017.

[28] V. Timčenko and S. Gajin, "Machine Learning based Network Anomaly Detection for IoT environments," Icist, 2018.

[29] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," IEEE Trans. Comput., vol. 65, no. 10, pp. 2986–2998, 2016.

[30] Y. K. Qian, D. S. Shan, D. Wei, Y. C. Li, and Z. F. Luo, "Network-wide anomalous flow identification method based on traffic characteristics distribution," Procedia Comput. Sci., vol. 131, pp. 1014–1022, 2018.

[31] M. Ring, D. Landes, and A. Hotho, "Detection of slow port scans in flow-based network traffic," PLoS One, vol. 13, no. 9, pp. 1–18, 2018.

[32] H. M. Shirazi, "Anomaly intrusion detection system using information theory, K-NN and KMC algorithms," Aust. J. Basic Appl. Sci., vol. 3, no. 3, pp. 2581–2597, 2009.

[33] H. M. Shirazi and Y. Kalaji, "An Intelligent Intrusion Detection System Using Genetic Algorithms and Features Selection," Majlesi J. Electr. Eng., vol. 4, no. 1, p. 33, 2010.

[34] B. Li, J. Springer, G. Bebis, and M. Hadi Gunes, "A survey of network flow applications," J. Netw. Comput. Appl., vol. 36, no. 2, pp. 567–581, 2013.

[35] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," J. Netw. Comput. Appl., vol. 60, pp. 19–31, 2016.

[36] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. "Anomaly detection: A survey, ". ACM Comput. Surv. 41, 3, Article 15 (July 2009), 58 pages.

[37] N. Moustafa, J. Hu, and J. Slay, "A holistic review of Network Anomaly Detection Systems: A comprehensive survey," J. Netw. Comput. Appl., vol. 128, no. December, pp. 33–55, 2019.

[38] M. F. Umer, M. Sher, and Y. Bi, "Flow-based intrusion detection: Techniques and challenges," Comput. Secur., vol. 70, pp. 238–254, 2017.

[39] Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A. and Stiller, B., 2010. "An Overview of IP Flow-Based Intrusion Detection, ". IEEE Communications Surveys & Tutorials, 12(3), pp.343-356.

[40] Shannon, C., 1948. "A Mathematical Theory of Communication, ". Bell System Technical Journal, 27(3), pp.379-423. DOI: 10.1002/j.1538-7305.1948.tb01338.x

[41] N. Moustafa, G. Creech, and J. Slay, "Flow aggregator module for analysing network traffic," Adv. Intell. Syst. Comput., vol. 710, no. December, pp. 19–29, 2018.

[42] C. Ferreira Lemos Lima, F. M. de Assis, and C. Protásio de Souza, "An Empirical Investigation of Attribute Selection Techniques based on Shannon, Rényi and Tsallis Entropies for Network Intrusion Detection," Am. J. Intell. Syst., vol. 2, no. 5, pp. 111–117, 2012.

[43] F. L. Acker, "Use of Entropy for Feature Selection with Intrusion Detection System Parameters," ProQuest Diss. Thesis, no. 370, p. 177, 2015.

[44] Berezinski, B. Jasiul, M. Szpyrka, An entropy-based network anomaly detection method, Entropy 17 (4): 2367–2408, (2015) doi: doi.org/10.3390/e17042367.

[45] P. Bereziński, M. Szpyrka, B. Jasiul, and M. Mazur, "Network anomaly detection using parameterized entropy," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8838, pp. 465–478, 2014.

[46] P. Bereziński, J. Pawelec, M. Małowidzki, and R. Piotrowski, "Entropy-Based Internet Traffic Anomaly Detection A Case Study.pdf," vol. 2012, 2012.

[47] Basicevic, I., Ocovaj, S. and Popovic, M., 2014. "Evaluation of entropy-based detection of outbound denial-of-service attacks in edge networks, ". Security and Communication Networks, 8(5), pp.837-844.

[48] T. Liu, Z. Wang, H. Wang, and K. Lu, "An entropy-based method for attack detection in large scale network," Int. J. Comput. Commun. Control, vol. 7, no. 3, pp. 509–517, 2012.

[49] 2022. [online] Available at: <https://www.snort.org/> [Accessed 23 January 2022]..

[50] P. Winter, H. Lampesberger, M. Zeilinger, and E. Hermann, "On Detecting Abrupt Changes in Network Entropy Time Series," pp. 194–205, 2011.

[51] A. S. S. Navaz, V. Sangeetha and C.Prabhadevi, "Entropy based Anomaly Detection System to Prevent DDoS Attacks in Cloud," vol. 62, no. 15, pp. 42–47, 2013.

[52] O. Gu, P. Fogla, D. Dagon, W. Lee, and B. Skoric, "Towards an information-theoretic framework for analyzing intrusion detection systems," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 4189 LNCS, pp. 527–546, 2006.

[53] J. Santiago-Paz and D. Torres-Roman, "On Entropy in Network Traffic Anomaly Detection," no. 2009, p. B008, 2015.

[54] Lionel Fillatre, Igor Nikiforov, Pedro Casas Hernandez, Sandrine Vaton. "Optimal volume anomaly detection in network traffc flows, ". EUSIPCO'08 : 16th European Signal Processing Conference, Aug 2008, Lausanne, Switzerland. ffhal-00540901.

[55] Daniela Brauckhoff, Bernhard Tellenbach, Arno Wagner, Martin May, and Anukool Lakhina. 2006. "Impact of packet sampling on anomaly detection metrics, ". In Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC). Association for Computing Machinery, New York, NY, USA, 159–164. DOI:https://doi.org/10.1145/1177080.1177101.

[56] B. M. Tellenbach, 2012. "Detection, Classification and Visualization of Anomalies using Generalized Entropy Metrics, Doctoral of Science. ETH ZURICH.

[57] Kopylova, Y., Buell, D., Huang, C. and Janies, J., 2008. "Mutual information applied to anomaly detection, ". Journal of Communications and Networks, 10(1), pp.89-97.

[58] A. Lakhina, M. Crovella, C. Diot," Mining Anomalies Using Traffic Feature Distributions", Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications 35 (4), (2005) 217-228. doi: 10.1145/1080091.1080118.

[59] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," Comput. Commun. Rev., vol. 34, no. 4, pp. 219–230, 2004.

[60] P. D. Bojović, I. Bašičević, S. Ocovaj, and M. Popović, "A practical approach to detection of distributed denial-of-service attacks using a hybrid detection method," Comput. Electr. Eng., vol. 73, pp. 84–96, 2019.

[61] O. Joldzic, Z. Djuric, and P. Vuletic, "A transparent and scalable anomaly-based DoS detection method," Comput. Networks, vol. 104, pp. 27–42, 2016.

[62] D. Rossi and S. Valenti, "Fine-grained traffic classification with Netflow data," IWCMC 2010 - Proc. 6th Int. Wirel. Commun. Mob. Comput. Conf., pp. 479–483, 2010.

[63] P. Barford, J. Kline, D. Plonka, and A. Ron, "A Signal Analysis of Network Traffic Anomalies," Proc. 2nd Internet Meas. Work. (IMW 2002), pp. 71–82, 2002.

[64] A. N. Huy, V. N. Tam, I. K. Dong, and D. Choi, "Network traffic anomalies detection and identification with flow monitoring," 5th IEEE IFIP Int. Conf. Wirel. Opt. Commun. Networks, WOCN 2008, 2008.

[65] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," Proc. - Conf. Local Comput. Networks, LCN, no. October 2010, pp. 408–415, 2010.

[66] R. Guo,H. Yin, D. Wang, and B. Zhang, "Research on the active DDoS filtering algorithm based on IP flow," 5th Int. Conf. Nat. Comput. ICNC 2009, vol. 4, no. December, pp. 628–632, 2009.

[67] L. Ertoz, E. Eilertson, A. Lazarevic, P. Tan, V. Kumar, J. Srivastava and P. Dokas. (2004). "Minds-minnesota intrusion detection system, ". Next generation data mining, 199-218.

[68] T. Pevny, M. Rehak, and M. Grill, "Identifying suspicious users in corporate networks," Proc. Work. Inf. forensics Secur., pp. 1–6, 2012.

[69] B. Tellenbach, M. Burkhart, D. Schatzmann, D. Gugelmann, and D. Sornette, "Accurate network anomaly classification with generalized entropy metrics," Comput. Networks, vol. 55, no. 15, pp. 3485–3502, 2011.

[70] K. Xu, Z. Zhang, and S. Bhattacharyya, "Internet Traffic Behavior Profiling for Network Security Monitoring," vol. 16, no. 6, pp. 1241–1252, 2008.

[71] J. Ibrahim, V. Timčenko, and S. Gajin, "A comprehensive flow-based anomaly detection architecture using entropy calculation and machine learning classification," 2019.

[72] V. Timčenko, J. Ibrahim, and S. Gajin, "The Hybrid Machine Learning Support for Entropy Based Network Traffic Anomaly Detection," no. 1.

[73] I. Özçelik and R. R. Brooks, "Deceiving entropy based DoS detection," Comput. Secur., vol. 48, pp. 234–245, 2015.

[74] T. Pietraszek, 2006. "ALERT CLASSIFICATION TO REDUCE FALSE POSITIVES IN INTRUSION DETECTION, ". Doctoral of Science. Faculty of Applied Sciences, Albert-Ludwigs-University of Freiburg im Breisgau.

[75] S. Sharifi, A., Zad, F., Farokhmanesh, F., Noorollahi, A. and Sharif, J., 2014. An "Overview of Intrusion Detection and Prevention Systems (IDPS) and Security Issues, ". IOSR Journal of Computer Engineering, 16(1), pp.47-52.

[76] Jeff Melnick, 2022. Top 10 Most Common Types of Cyber Attacks. [online] Blog.netwrix.com. Available at: < https://blog.netwrix.com/2018/05/15/top-10-most-common-types-of-cyber-attacks > [Accessed 24 January 2022].

[77] Stallings William, and Lawrie Brown, Computer Security: Principles and Practice, Second Edition. 2013.
[78] E. Edition, O. Systems, S. Edition, and B. D. Communications, the William Stallings Books on Computer Data and Computer Communications , Eighth Edition, vol. 139, no. 3. 2011.
[79] J. P. Anderson, "Computer security threat monitoring and surveillance," Tech. Rep. James P Anderson Co Fort Washingt. Pa, p. 56, 1980.
[80] K. Julisch, 2003. "Using root cause analysis to handle intrusion detection alarms,". Doctors of science. IBM Zurich Research Laboratory, Switzerland.
[81] O. E. Elejla, M. Anbar, B. Belaton, and B. O. Alijla, "Flow-Based IDS for ICMPv6-Based DDoS Attacks Detection," Arab. J. Sci. Eng., vol. 43, no. 12, pp. 7757–7775, 2018.
[82] S. Garcıa, 2014. "Identifying, Modeling and Detecting Botnet Behaviors in the Network,". Doctoral Thesis. Higher Institute of Software Engineering Tandil - Department of Computing and Systems, Argentina.
[83] T. Ptacek and T. Newsham, "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection," CEUR Workshop Proc., vol. 1542, pp. 33–36, 1998.
[84] V. Paxson, "Bro: A system for detecting network intruders in real-time," Proc. 7th USENIX Secur. Symp., 1998.
[85] A. Mokarian, A. Faraahi, and A. G. Delavar, "False Positives Reduction Techniques in Intrusion Detection Systems-A Review," Int. J. Comput. Sci. Netw. Secur., vol. 13, no. 10, pp. 128–134, 2013.
[86] Chinyere. I. Akobundu and G. E. Okereke, "RESEARCH ARTICLE THE TAXONOMY OF INTERNET TRAFFIC ANOMALY DETECTION," vol. 7, pp. 775–782, 2019.
[87] C. Hu, C., Han, L., & Yiu, S. M. (2016). Efficient and secure multi-functional searchable symmetric encryption schemes. Security and Communication Networks, 9(1), 34-42.
[88] Sebastián García, "Identifying , Modeling and Detecting Botnet Behaviors in the Network," no. November 2014, 2015.
[89] Przemyslaw, B., 2015. *Entropy-Based Network Anomaly Detection*. PhD. AGH University of Science and Technology - Faculty of Electrical Engineering.
[90] D. Brauckhoff, B. Tellenbach, A. Wagner, M. May, and A. Lakhina, "Impact of packet sampling on anomaly detection metrics," Proc. ACM SIGCOMM Internet Meas. Conf. IMC, pp. 159–164, 2006.
[91] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang, "An empirical evaluation of entropy-based traffic anomaly detection," Proc. ACM SIGCOMM Internet Meas. Conf. IMC, pp. 151–156, 2008.
[92] Y. Abuadlla, G. Kvascev, S. Gajin, and Z. Jovanović, "Flow-based anomaly intrusion detection system using two neural network stages," Computer Science and Information Systems, vol. 11, no. 2, pp. 601–622, 2014, doi: 10.2298/CSIS130415035A.
[93] "What is NetFlow or IPFIX? Efficient Network Monitoring | Flowmon." https://www.flowmon.com/en/solutions/network-and-cloud-operations/netflow-ipfix (accessed Nov. 17, 2021).
[94] R. Hofstede *et al.*, "Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX," in *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2037-2064, Fourthquarter 2014, doi: 10.1109/COMST.2014.2321898.
[95] H. Nguyen and C. Deokjai. "Network Anomaly Detection: Flow-based or Packet-based Approach?." arXiv preprint arXiv:1007.1266 (2010).

[96] A. Sperotto and A. Pras, "Flow-based intrusion detection," 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops, 2011, pp. 958-963, doi: 10.1109/INM.2011.5990529.

[97] G. Androulidakis, V. Chatzigiannakis and S. Papavassiliou, "Network anomaly detection and classification via opportunistic sampling," in *IEEE Network*, vol. 23, no. 1, pp. 6-12, January-February 2009, doi: 10.1109/MNET.2009.4804318.

[98] F. Balibrea, "On Clausius, Boltzmann and Shannon Notions of Entropy," 2016. Journal of Modern Physics, 7, 219-227. doi: 10.4236/jmp.2016.72022.

[99] Winter, P., Lampesberger, H., Zeilinger, M., & Hermann, E. (2011). On detecting abrupt changes in network entropy time series. In *Communications and Multimedia Security - 12th IFIP TC 6 / TC 11 International Conference, CMS 2011, Proceedings* (7025 ed., Vol. 7025, pp. 194-205). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 7025 LNCS). https://doi.org/10.1007/978-3-642-24712-5_18.

[100] A. Rényi, "On measures of entropy and information. ", in Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability 1, (1961) 547–561.

[101] C. Tsallis, "Possible generalization of Boltzmann-Gibbs statistics," Journal of Statistical Physics, vol. 52, no. 1–2, pp. 479–487, Jul. 1988, doi: 10.1007/BF01016429.

[102] A. J. Lawrance and P. A. W. Lewis. "An Exponential Moving-Average Sequence and Point Process (EMA1)." Journal of Applied Probability 14, no. 1 (1977): 98–113. https://doi.org/10.2307/3213263.

[103] Dušan Pantić, "Software for the analysis of the efficiency of calculating the entropy of network traffic", Master thesis, University of Belgrade - Faculty of Electrical Engineering, 2018.

[104] B. Tellenbach, M. Burkhart, D. Schatzmann, D. Gugelmann, and D. Sornette, "Accurate network anomaly classification with generalized entropy metrics," Computer Networks, vol. 55, no. 15, pp. 3485–3502, Oct. 2011, doi: 10.1016/j.comnet.2011.07.008.

[105] I. Özçelik, İlker and R. R. Brooks, "Deceiving entropy-based DoS detection," in Signal Processing, Sensor/Information Fusion, and Target Recognition XXIII, Jun. 2014, vol. 9091, p. 90911P. doi: 10.1117/12.2054434.

[106] C. Fachkha, E. Bou-Harb, and M. Debbabi, "Fingerprinting internet DNS amplification DDoS Activities," 2014 6th International Conference on New Technologies, Mobility and Security - Proceedings of NTMS 2014 Conference and Workshops, 2014, doi: 10.1109/NTMS.2014.6814019.

[107] NetVizura, "NetVizura Netflow Analyzer, Case study – DDoS Attack by NTP Amplification.", Accessed 22 July 2020. https://www.netvizura.com/files/products/netflow/resources/doc/DDoS-Attack-by-NTP-Amplification-NetVizura.pdf

[108] M. Allman, V. Paxson, and J. Terrell, "A brief history of scanning," Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC, pp. 77–82, 2007, doi: 10.1145/1298306.1298316.

[109] R. Hofstede, L. Hendriks, A. Sperotto and A. Pras, , "Public Review for SSH Compromise Detection using NetFlow / IPFIX, " vol. 44, no. 5, pp. 20–26, 2014.

[110] G. Nychis, V. Sekar, D.G. Andersen, H. Kim, and H. Zhang, An Empirical Evaluation of Entropy-based Traffic Anomaly Detection, in: Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement (IMC "08), Vouliagmeni, Greece, 20–22 October 2008: 151–156. ACM New York, NY, USA. doi: 10.1145/1452520.1452539

[111] K. Xu, Z.L. Zhang, S. Bhattacharyya, "Internet traffic behavior profiling for network security monitoring" , IEEE/ACM Transactions on Networking 16 (6), (2008) 1241-1252, doi: 10.1109/TNET.2007.911438.

[112] M. V. Mahoney and P. K. Chan, "An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 2820, no. Ll, pp. 220–237, 2003, doi: 10.1007/978-3-540-45248-5_13.

[113] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," 2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings, no. December, 2015, doi: 10.1109/MilCIS.2015.7348942.

[114] "ISOT Botnet dataset – Cyber Data Scientist." http://cyberdatascientist.com/product/isot-botnet-dataset/ (accessed Nov. 11, 2021).

[115] "The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background traffic. — Stratosphere IPS." https://www.stratosphereips.org/datasets-ctu13 (accessed Nov. 11, 2021).

[116] Sperotto, A., Sadre, R., van Vliet, F., & Pras, A. (2009). A Labeled Data Set For Flow-based Intrusion Detection. In G. Nunzi, C. Scoglio, & X. Li (Eds.), IP Operations and Management: 9th IEEE International Workshop, IPOM 2009, Venice, Italy, October 29-30, 2009. Proceedings (pp. 39-50). (Lecture Notes in Computer Science; Vol. 5843). Springer. https://doi.org/10.1007/978-3-642-04968-2_4

[117] I. Sharafaldin, A.H. Lashkari, A. Ghorbani, Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterizatio, in: Proceedings of the 4th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP, ISBN 978-989-758-282-0, (2018) pages 108-116. doi: 10.5220/0006639801080116.

[118] A. Lakhina, M. Crovella, C. Diot, Diagnosing Network-Wide Traffic Anomalies, ACM SIGCOMM Computer Communication Review 34 (4), (2004) 219-230. doi: 10.1145/1030194.1015492.

[119] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras and V.Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments", Computer Networks, Vol 62, (2014), 122-136, doi: 10.1016/j.bjp.2013.10.014.

BIOGRAPHY

The candidate was born on December 8, 1965 in Libya, in the city of Alriyahinah. He completed his basic studies at the University of Tripoli, Faculty of Sciences, Department of Computer Science.

He completed his master's degree studies at the University of Belgrade - Faculty of Electrical Engineering, where he defended the master's work of real time operating systems, with a focus on their application in scientific-research canters.

From May 1989 to March 2006 he was employed at the Research Center in Tripoli in the workplace of the programmer, where he worked on software system implementation, installation and maintenance of software and computers, as well as training for different types of user applications (MS Windows, MS Office, Unix) and some programming languages.

In the period from June 2006 to November 2013, he was employed at the College of Computer Technology Tripoli (CCTT), in Libya as teaching assistant. During that period, he was the director of the office for scientific academic affairs at the College of Computer Technology Tripoli. He also lectured from various computer areas and was a CCNA instructor within the Cisco Network Academy Tripoli and the Cisco Network Academy Injella in Libya.

He enrolled in academic doctoral studies in 2014 at the University of Belgrade - Faculty of Electrical Engineering, Computer Science and Informatics. During the studies he successfully passed all the prescribed exams with a score of 9.8, out of which 9 professional subjects with a score of 10, and scored 120 ESPB points. During the doctoral studies, he conducts research work on a wide range of computer networking technologies by researching software-defined networks (SDNs) in order to quickly focus on the security of computer networks. The research in this field is focused on Intrusion Detection Systems based on entropy and machine learning, from which the author or co-author in four scientific papers was published and presented at international expert conferences.

Scientific area: Computer Networks, Network Security, Machine Learning, Intrusion Detecting System

LIST OF PAPERS

The results of the research presented in the doctoral dissertation:

[1] Ibrahim, Juma, and Slavko Gajin. "Entropy-based network traffic anomaly classification method resilient to deception." Computer Science and Information Systems 19(1):87–116, (2021), DOI: https://doi.org/10.2298/CSIS201229045I (IF: 1.167)

[2] J. Ibrahim, V. Timčenko, and S. Gajin, „A comprehensive flow-based anomaly detection architecture using entropy calculation and machine learning classification “, in 9th Int. Conf. Information Society and Technology – ICIST2019, pp. 138-143, 2019. Online: https://www.eventiotic.com/eventiotic/library/paper/466

[3] A. Elsadai, J. Ibrahim, F. Hajjaj, P. Jakić, “The Overview of Intrusion Detection System Methods and Techniques,” in Sinteza 2019 - International Scientific Conference on Information Technology and Data Related Research, Belgrade, Singidunum University, Serbia, 2019, pp. 155-161. doi:10.15308/Sinteza-2019-155-161

[4] V. Timčenko, J. Ibrahim, and S. Gajin, „The hybrid machine learning support for entropy based network traffic anomaly detection“, in 9th Int. Conf. Information Society and Technology – ICIST2019, pp. 144-149, 2019. Online: https://www.eventiotic.com/eventiotic/library/paper/467

Works on conferences of international significance:

[5] Juma Ibrahim, S. Gajin: "SDN-BasedIntrusionDetection System Literature Review", 16th International Symposium INFOTEH-JAHORINA 2017, 22-24 March 2017, Jahorina, Bosnia and Herzegovina.

[6] Juma Ibrahim: "The role of open source software in the success of e-management", first international electronic management conference, 1-4 June 2010, Tripoli, Libya.

[7] Juma Ibrahim, Najat Darweel: "The digital gap in the word and its impact in e-learning", the first International Conference on E-learning for ALL (ELFA2010), 3-5 June 2010, Hammamet, Tunisia.

Works on conferences of national importance (Republic of Libya):

[8] Musbah Al-ahrash, MusbahAbuajella, Juma Ibrahim: "Information technology and computer science", Aljabel Algharbi University, Faculty of Science 25-27 April 2009, Gharyan, Libya.

[9] Juma Ibrahim: "Digital gap in the Arab world", The Islamic organization for education, culture and science, 5-7 October 2009, Kairouan, Tunisia.

[10] Juma Ibrahim, Mustafa Shfelow: "Security requirements for the electronic archive", the general electricity company, 19-20 December 2009, Tripoli, Libya.

образац изјаве о истоветности штампане и електронске верзије докторског рада

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора Жума Ибрахим (Juma Ibrahim)

Број индекса 2014/5051

Студијски програм Рачунарска техника и информатика

Наслов рада Архитектура система за препознавање неправилности у мрежном саобраћају засновано на анализи ентропије

Ментор др Славко Гајин, ванредни професор

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла ради похрањивања у **Дигиталном репозиторијуму Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис аутора

У Београду, 2.6.2022.

образац изјаве о ауторству

Изјава о ауторству

Име и презиме аутора Жума Ибрахим (Juma Ibrahim)

Број индекса 2014/5051

Изјављујем

да је докторска дисертација под насловом

Архитектура система за препознавање неправилности у мрежном саобраћају засновано на анализи ентропије

- резултат сопственог истраживачког рада;
- да дисертација у целини ни у деловима није била предложена за стицање друге дипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио/ла интелектуалну својину других лица.

Потпис аутора

У Београду, 2.6.2022.

образац изјаве о коришћењу

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

Архитектура система за препознавање неправилности у мрежном саобраћају засновано на анализи ентропије

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство (CC BY)

2. Ауторство – некомерцијално (CC BY-NC)

3. Ауторство – некомерцијално – без прерада (CC BY-NC-ND)

4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)

5. Ауторство – без прерада (CC BY-ND)

6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци.
Кратак опис лиценци је саставни део ове изјаве).

Потпис аутора

У Београду, 2.6.2022.