Institute of Visualization and Interactive Systems
University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelor's Thesis Nr. 48

Processing Dynamic Computer Network Data for Visual Analysis

Hanna Schaefer

Course of Study:	Software Engineering
Examiner:	Prof. Dr. Ertl
Supervisor:	Dipl.Inf. Steffen Koch M.Sc. Robert Krüger Dipl.Inf. Dennis Thom
Commenced:	April 15, 2013
Completed:	October 1, 2013
CR-Classification:	H.3 INFORMATION STORAGE AND RETRIEVAL, H.5 INFORMATION INTERFACES AND PRESENTATION, I.2 ARTIFICIAL INTELLIGENCE

Abstract

In recent times datasets have become larger and more and more difficult to understand for users. Therefore Visual Analytics research investigates on combining automation methods with user related analysis. A special type of this field is security visualization. Since information like connection data or health status of each computer in the network are very abstract the help of automation methods becomes even more important to make interesting outliers and anomalies obvious to the user. This bachelor thesis compares two different approaches for anomaly detection in security visualization. The study is based on the VAST 2013 Mini Challenge 3 and its submission of a University of Stuttgart and Peking University cooperation. This thesis concentrates on comparing the two automation methods seasonal trend decomposition (STL) for numerical data fields, such as bytes and packages, and the sample entropy (or Shannon Entropy) method for categorical data fields, such as IP and port. Both approaches should enable the user to find events in the given network dataset and thus to understand the risks and attacks in the network of the VAST challenges example company Big Marketing. As a result the methods are similar in the quantity of anomalies found, but differ in the type of anomaly. Since the STL focuses on different variables, some variables show more scan events and others more DOS events. Combining all results from the different variables the STL offers a higher number of true anomalies. On the other hand, the sample entropy is more intuitive to use and gives hints on the type of event without using other visualizations. In a small user study the entropy method was clearly preferred and performed in a better way in the result of given tasks. As a conclusion this thesis suggests the entropy methods in any similar context to the given benchmark and system, but also suggests that the STL methods could be more efficient with different parameters of network security.

Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

The major incentive for this thesis was the VAST challenge 2013. This annual competition in current research fields focuses in solving "real world problems" by applying analytics methods. Visual Analytics is a research field for methods that combine automation algorithms with user input. It creates this individual feedback loops, in order to solve problems faster and more reliable. The specific case on which this thesis concentrates is the Mini challenge 3 (Chapter 2.3). It applies Visual Analytics in the field of network security. Network security is a field that detects, prevents and encounters network attacks. It provides large datasets and often has to deal with dynamic information in real time applications. The given benchmark dataset consists of two weeks network data of the artificial company "Big Marketing". The system AnNetTe was developed as a benchmark environment and won the "Outstanding Situation Awareness" award in this challenge. This system integrates different automation methods to give visual feedback and make the analysis easier for the user. This feedback loop evaluates between different automation methods. In this thesis two approaches were evaluated: the Shannon entropy and the seasonal trend decomposition (STL). These are two different promising approaches in reference to network security datasets and Visual Analytics.

1.2. Task description

The first goal of this thesis is to find anomaly detection methods that are applicable for the integration into a Visual Analytics feedback loop. At the same time it concentrates on features that are found in network datasets. Afterwards these two anomaly detection methods are evaluated for their advantages and problems. The benchmark dataset for this evaluation is provided by the IEEE VAST challenge 2013 and is available at http://www.vacommunity.org/VAST+Challenge+2013%3A+Mini-Challenge+3. Part of the task is to participate in Mini challenge 3 and to create a benchmark system for the evaluation that integrates all anomaly detection methods that are to be compared. The integration into a specific Visual Analytics system is carried out in cooperation with a parallel bachelor thesis focusing on requirements of visual, interactive techniques that are suitable for presenting and evaluating the detected anomalies in different systems [Mer13]. My thesis on the other hand focuses on the necessary research for the detection methods and the preprocessing or calculation of the anomalies in the given dataset. For evaluating the determined techniques, three different evaluation steps are taken. The resulting anomalies are evaluated in three separate steps. In the first step they are categorized by the time they are found in and their

positive or negative orientation. The second step includes the meaning of the anomalies. It sorts out the false alarms and categorizes the events into attack types. The third step evaluates the user interaction with both methods in a small study. The detection methods are evaluated using feedback as well as the users performance on given tasks.

1.3. Structure of thesis

The following thesis is structured from basic concepts over development steps up to the complete network security system with anomaly detection which is then finally evaluated by various measures. First the introduction gives an insight into the overall research aim and the question discussed in this thesis. The basic concepts (Chapter 2) for understanding the main chapters are then described in a general chapter. When the reader has understood all features needed in the thesis he is able to learn about the background and historical environment of these features in the related work (Chapter 3). The main part of the thesis first explains all necessary steps and developments for using the discussed anomaly detection methods (Chapter 4). Afterwards those automation features are set in context by describing the benchmark dataset of the VAST 2013 and the system AnNetTe (Chapter 5) in which the anomaly calculation results were integrated and evaluated. The final part of the thesis shows by which measurement the methods were evaluated and which results the different categories achieved. In the conclusion (Chapter 7) all these results are shortly summarized and the comparison is finished by presenting the most suitable method according to the given environment.

2. Basic concepts

To understand the content of this thesis there are five essential fields of necessary previous knowledge. This thesis is originated from two research fields at the same time. One is network security, which gives an insight into the amount and type of data as well as the goals of applications in this field. The second research field is Visual Analytics, which on the other hand describes "how" an application or approach is found and which concepts are used to reach the goals. The third necessary previous knowledge is the VAST challenge, which was the initial thought for doing this thesis. The VAST challenge (chapter2.3) combines network security applications and questions with visual analytic tools or methods. In case of this thesis the VAST 2013 dataset was used as a benchmark for all evaluations. The last two points that need to be explained are the two concepts that were chosen for comparison out of the field of anomaly detection methods. Those are the Shannon entropy and the seasonal trend decomposition (STL). Both are introduced below.

2.1. Network security

Network security is the overall concept of prevention, detection, reduction and active response to security risks and attacks in computer networks. Those concepts are generally used by network administrators observing large company networks including sensitive data and reliability requirements. Common risks are unauthorized access, misuse of resources, or damage to network components. The common attack types dealt with in this thesis are shortly explained. A Denial-of-Service (DOS) attack is an access of an external IP towards the network. By using large amounts of data sent to one or more computers over several source ports at the same time, the attacking system wants to claim a lot of denial of service responses. The attacked system reaches a high CPU load and might crash after some time. If this attack is done by multiple attacking IPs at the same time and with the same aim, it is called a DDOS attack. A different type of attack is the scan attack. This consists of one or more IPs scanning all the ports of an internal workstation or server. They try to detect weaknesses and find an entry into the system. If the scan is done towards one single aim it is a port scan. With multiple attacked IPs it is a network scan. Other security problem can occur after some successful attacks. Those are worms, viruses or other malevolent software that spreads in the network and damages software or files. In this thesis we focus on the detection of such attacks, to provide network administrators with fast visual feedback about problems and thus give them the opportunity for prevention or reaction to the events. By working with this data the attacks can be revealed and further damage can be prevented. To reach the goal more easily visual analytic methods are combined with anomaly detection methods and evaluated by their effect.

2.2. Visual Analytics

Visual Analytics is a research area that focuses on methods and concepts for modifying and analyzing large data. In this context it is also used for the application of the network security domain. The domain around Visual Analytics tries to combine visual research like information visualization and scientific visualization with an analytical reasoning. Therefore algorithms of the analysis domain are used in interfaces in a way that creates interactive feedback loops and thus combines the power of human perception with the accuracy of automation processes. Thomas and Cook [TC05] defined the following focus areas as cornerstones of the Visual Analytics field:

- analytical reasoning techniques that let users obtain deep insights that directly support assessment, planning, and decision making;
- visual representations and interaction techniques that exploit the human eyes broad bandwidth pathway into the mind to let users see, explore, and understand large amounts of information simultaneously;
- data representations and transformations that convert all types of conflicting and dynamic data in ways that support visualization and analysis; and
- techniques to support production, presentation, and dissemination of analytical results to communicate information in the appropriate context to a variety of audiences.

In this thesis Visual Analytics is used in order to fulfill the first focus and third area. Better insight into a security network dataset should be reached by giving the user hints on automatically detected anomalies. By providing the anomalies in an intuitive way, the easy perception of users is integrated into the interaction. The user then can do a faster analysis of central data points without first having to search for such points. The overall aims of Visual Analytics such as problems of large, complex and abstract data are tackled by two different anomaly detection methods which then are evaluated against each other in accordance to quantity, quality and usability.

2.3. VAST challenge

Both above described research fields are regarded in the yearly IEEE VAST Symposium. This symposium is accompanied by the Visual Analytics Science and Technology challenge, which is shortly called VAST challenge. This contest was a large incentive for the given thesis. Its provided dataset was used to implement a network security application including Visual Analytics methods, and as a benchmark for the evaluations between both anomaly detection methods used. The first VAST contest was held in 2006. Its purpose is to give an incentive for the creation of Visual Analytics tools and methods tested on a provided benchmark dataset. This incentive should fasten the transfer from current research knowledge to commercial products and thus increase the evolution rate of new approaches and systems. For good results in both correctness concerning the benchmark data and utility of the tools

used in the challenges the committee provides awards and enable a presentation at the IEEE VIS. Of all three Mini challenges provided, this thesis and dataset refer to the third Mini challenge of 2013.

2.4. VAST 2013 Mini challenge 3 overview

The benchmark dataset of the VAST 2013 Mini challenge 3 is a fictional dataset gained from a company called `Big Marketing`. The network of this company is described with all its components. It consists of three major parts which are locally different. Each part employs around 400 workers who all have their own network access. Additionally each company part has its own http, smtp, domain and admin servers. The diagram in figure 2.1 shows how all those parts connect to one firewall and gateway where all the connections are logged. For this dataset the task is then to provide all unusual events or attacks towards the network and gain helpful hints for the system administrator.

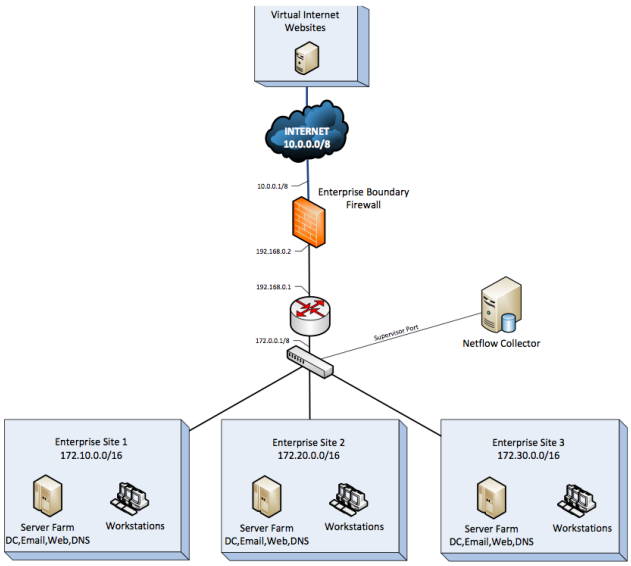
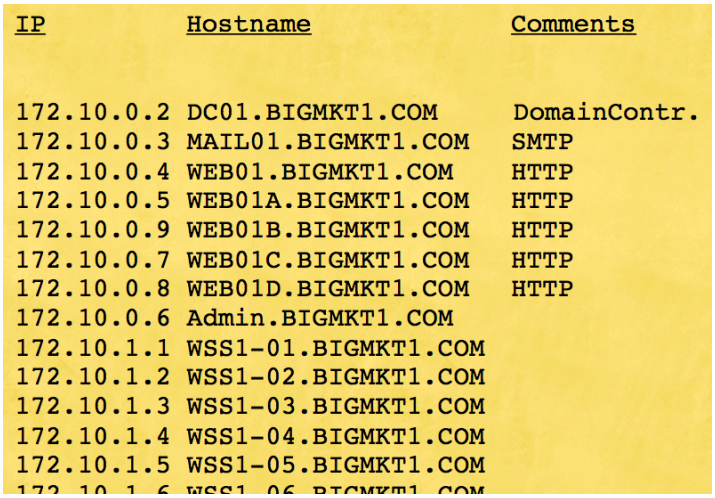


Figure 2.1.: Network Architecture (Source: VAST 2013 Mini challenge data description)

The architecture is important for the later analysis of the data since the STL decomposition is also used on a dataset split by the company parts to get a better result for the seasonal behavior and thus an easier anomaly detection. To split the dataset the correspondence of IPs with the parts of the network is important. Therefore the challenge also provided a list of all internal IPs and their meaning as shown in figure 2.2. This list was most useful for recognizing the servers in each company part, which were often the aim of an attack towards the system, since they provide the easiest entry point

The fictional `Big Marketing` company then provides four different information sources over a time period of two weeks. The sources are shortly listed with their sizes or limitations and then described in detail below:



IP	Hostname	Comments
172.10.0.2	DC01.BIGMKT1.COM	DomainContr.
172.10.0.3	MAIL01.BIGMKT1.COM	SMTP
172.10.0.4	WEB01.BIGMKT1.COM	HTTP
172.10.0.5	WEB01A.BIGMKT1.COM	HTTP
172.10.0.9	WEB01B.BIGMKT1.COM	HTTP
172.10.0.7	WEB01C.BIGMKT1.COM	HTTP
172.10.0.8	WEB01D.BIGMKT1.COM	HTTP
172.10.0.6	Admin.BIGMKT1.COM	
172.10.1.1	WSS1-01.BIGMKT1.COM	
172.10.1.2	WSS1-02.BIGMKT1.COM	
172.10.1.3	WSS1-03.BIGMKT1.COM	
172.10.1.4	WSS1-04.BIGMKT1.COM	
172.10.1.5	WSS1-05.BIGMKT1.COM	

Figure 2.2.: Example of the IP Listing (Source: VAST 2013 Mini challenge data description)

- Netflow data (9 GB, 19 columns)
- Health data (4 GB, 14 columns)
- IPS Log (2 GB, 13 columns)
- Question and Answer (5 questions)

2.4.1. Netflow data

The most important dataset in our analysis is the netflow data. It provides information about each connection from, to or inside the network. The data is given over a time span of two weeks. This dataset is the one on which the two different anomaly detection methods are applied. This dataset is used since it has the closest connection to the traffic that is observed and it is also provides a lot of variables that have seasonal behavior and are thus very useful for pattern recognition methods. The uncertainty of the dataset mentioned in the task description was not further considered in the participation. Most connections were represented in both directions, so errors in the destination and source categorization would not have been grave. Also the important attacks were all consisting of multiple connections, so a small number of false IPs would not have diminished their notability. Following a list of all variables in one connection entry is given:

- time REAL
- parsedDate DATETIME
- dateTimeStr VARCHAR (22)
- ipLayerProtocol INTEGER
- ipLayerProtocolCode VARCHAR (10)
- firstSeenSrcIp VARCHAR (15)
- firstSeenDestIp VARCHAR (15)
- firstSeenSrcPort INTEGER
- firstSeenDestPort INTEGER
- moreFragments VARCHAR (1)
- contFragments VARCHAR (1)
- durationSeconds INTEGER
- firstSeenSrcPayloadBytes INTEGER
- firstSeenDestPayloadBytes INTEGER

- firstSeenSrcTotalBytes INTEGER
- firstSeenDestTotalBytes INTEGER
- firstSeenSrcPacketCount INTEGER
- firstSeenDestPacketCount INTEGER
- recordForceOut VARCHAR (1)

Of these variables the firstSeenSrcIp and firstSeenDestIp as well as firstSeenSrcPort and firstSeenDestPort were used as the categorical input to the entropy anomaly detection. These variables best represent the connection pattern itself and also give immediate hints on attackers and attack aims. For the STL anomaly detection The numerical values for bytes, payload, packages and duration were used. These variables are indirect measurements of the regularity of each connection. Even if a connection occurs regularly in each day, it might be used for file stealing on one of these days. Such an anomaly would then be shown in the bytes analysis. An example entry of this dataset is shown in figure 2.3.

TimeSeconds	parsedDate	dateTimeStr	ipLayerProtocol	ipLayerProtocolCode	firstSeenSrcIp	firstSeenDestIp	firstSeenSrcPort	firstSeenDestPort	moreFragments	contFragments	durationSeconds	firstSeenSrcPayloadBytes	firstSeenDestPayloadBytes	firstSeenSrcTotalBytes	firstSeenDestTotalBytes	firstSeenSrcPacketCount	firstSeenDestPacketCount	recordForceOut
1365034324	4/4/13 12:12 AM	20130404001204	6	TCP	10.0.3.76	172.10.0.4	34803	80	0	0	0	188	49559	1384	51619	22	38	0
1365034326	4/4/13 12:12 AM	20130404001206	6	TCP	10.0.3.76	172.10.0.5	34796	80	0	0	4	188	1401	466	1571	5	3	0

Figure 2.3.: Snippet of the netflow data

2.4.2. Health data

The health data is the log of a regular status report for each internal workstation. In the system this data helps to confirm negative results of attacks. For our anomaly detection it is important to note that IP 172.10.0.6 is the health monitor and thus its multiple connections are not an outlier but a regular and benign event. Each row of data on the health status consists of the following variables:

- ID INTEGER
- hostname VARCHAR (22)
- servicename VARCHAR (5)
- currenttime INTEGER
- statusvalue INTEGER
- bbcontent STRING
- receivedfrom VARCHAR (20)
- diskUsagePercent INTEGER
- pageFileUsagePercent INTEGER
- numProcs INTEGER
- loadAveragePercent INTEGER
- physicalMemoryUsagePercent INTEGER
- connMade INTEGER
- parsedDate DATETIME

2.4.3. IPS log

The IPS dataset consists of the log data from a newly installed Intrusion Prevention System. This data is only available for week two but very powerful for identifying malicious IPs or

suspicious connections. In the overall system this variable is very efficient, but for the anomaly detection it is not used, since each entry already is an anomaly in itself. Each row of data on the health status consists of the following variables:

- datetime parseddate
- priority varchar(20)
- operation varchar(10)
- messageCode varchar(15)
- protocol varchar(5)
- scrIP varchar(16)
- destIP varchar(16)
- srcPort int
- destPort int
- destService varchar(5)
- directin varchar(10)
- flags varchar(10)
- command varchar(1)

2.4.4. Q&A

The Question and answer data was a great opportunity for all teams to gain insight into the real events at the company. Every team had the chance to register and then ask up to five questions. In the course of the participation some questions have been asked concerning the meaning of special data snippets, special time frames and special IPs. The answers helped to understand the events that had been discovered and to differentiate the real events from regular peaks such as for example the multiple connections of each workstation towards the health monitor. These questions are not further described since they do not influence the anomaly detection which is subject to this thesis.

2.5. Anomaly detection

Anomaly or outlier detection is a term for data processing that searches any irregular part of a dataset. The data entries that do not fit the normal pattern are then called anomalies and depending on the domain interpreted with the corresponding meaning. For the network security analysis, anomalies mark events or attacks within the network. In some cases outliers are only regular bursts in the network activity, in others they are malicious activities. To differentiate them a human analyst is needed for interpreting the data. This thesis focuses on the combination of those two steps. After using anomaly detection method, the outliers are displayed visually in order to enable the analyst to quickly find those points of interest and then interpret them with other functions of the system. The two anomaly detection methods Shannon entropy and STL are used and compared in their efficiency, specialization and usability.

2.5.1. Seasonal trend decomposition

The first method for anomaly detection is the seasonal trend decomposition. This method was first introduced in 1990 by Robert and William Cleveland [CCMT90]. The concept is based on finding a seasonal behavior in the time series given and extracting this season from the main dataset. The remaining data is further decomposed into a smooth trend line and a remainder dataset, which contains all anomalies not fitting the model. To understand the concept of seasonal, trend and remainder data the small table below shows how the model works. In the first row an easy dataset of 6 numbers is shown. These numbers increase with each step and at the same time they have a small up and down movement. After distinguishing these two movements a seasonal behavior and a trend line can be determined. In line two of the following table the seasonal behavior which goes up and down by two with every step is shown. In line three on the other hand the trend increases by one every two steps. Adding up those numbers would then result in the original dataset, except for the last data point. Here the model doesn't fit and the remaining data is written into line four as a remainder.

Data	1	3	2	4	3	6
Season	1	3	1	3	1	3
Trend	0	0	1	1	2	2
Remainder	0	0	0	0	0	1

Table 2.1.: STL example data description

This separation is achieved by using a loess smoother on the original dataset and the separating of the smoothed data from the remainder by low-pass filters. These steps are done in multiple interconnected loops [CCMT90]. The detailed algorithmic of this methods are not discussed in this thesis, since it is an established method and already provided by various libraries. An original example dataset for this method is the "Daily carbon dioxide" data over several years. In figure 2.4 this dataset is shown as a seasonal trend decomposition. This dataset and figure both demonstrate what type of data is required to receive a reasonable decomposition result. The data needs to have a clear periodical pattern in order to create meaningful remainders for the anomaly detection. Whether this method is thus applicable to network datasets are further discussed in this thesis.

2.5.2. Shannon entropy

In contrast to the STL analysis the Shannon entropy is not used on a time series of numerical data, but on categorical datasets. For them it measures the concentration or dispersal of a dataset. Originally the formula of the Shannon entropy was calculated as follows [SW49]:
$H = -\sum_{i=1}^{R} p_i \ln p_i$
Here p_i is the number of connections counted for the i-th IP. In general the logarithm can be chosen freely but is most common for a natural logarithm. This measurement was just considered a formula for studying the distribution of sets. Later [LX01] this measurement was studied as an anomaly detection method, in various adaptations. The final so-called `sample entropy` version used in this thesis and the corresponding application was introduced to us

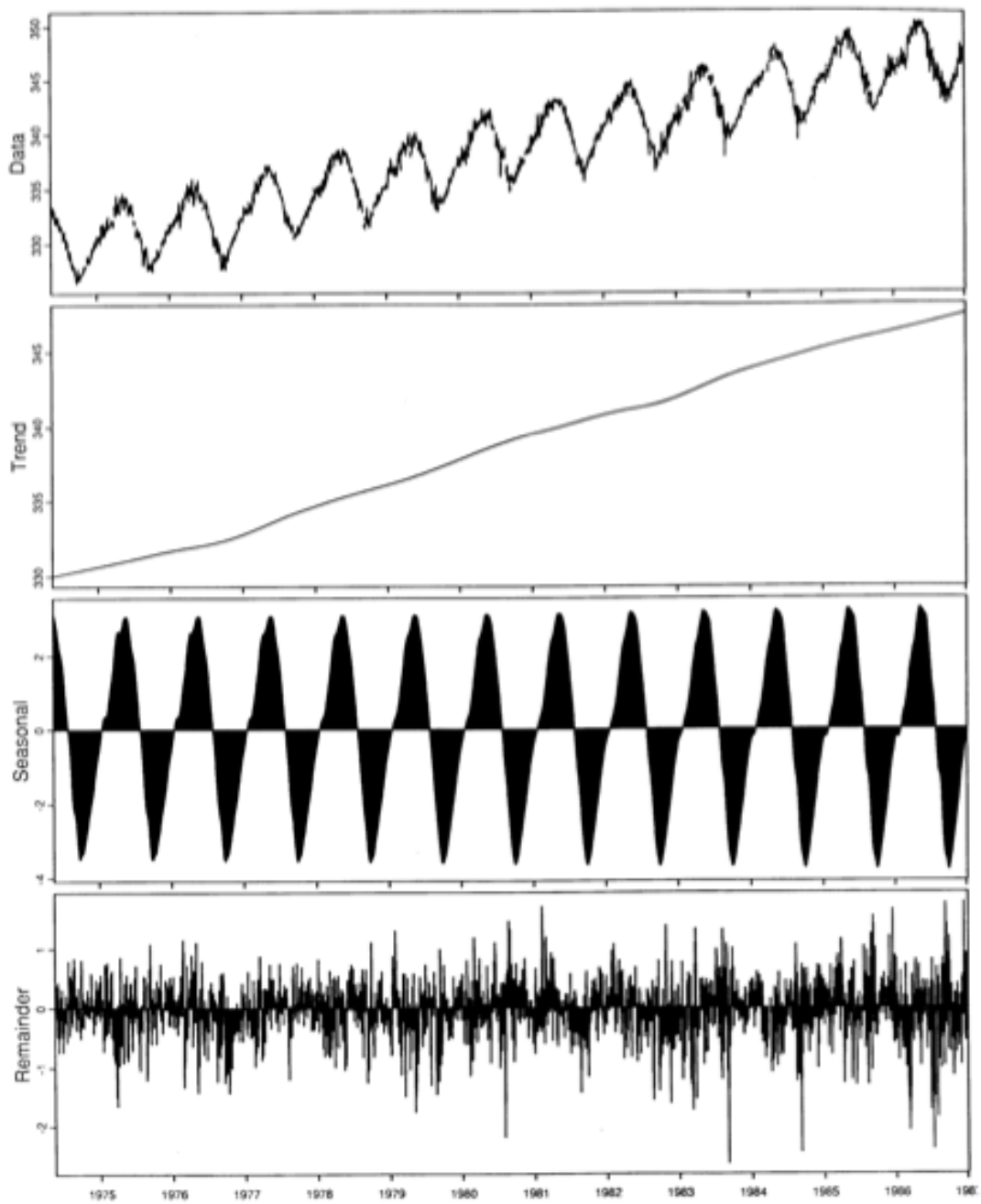


Figure 2.4.: General decomposition of the STL method (Source: [CCMT90]

by Lakhina, Crovella and Diot [LCC05]. In difference to the original Shannon entropy it is normalized against the total size N of the set and thus only produces values in between 0 and $\log_2(N)$. The formula is calculated as follows: $H = -\sum_{i=1}^{R} \frac{p_i}{N} \ln \frac{p_i}{N}$
The highest achievable value of $\log_2(N)$ describes a totally even distribution. Smaller values indicate more variety or concentration. To get an impression on how this formula affects the results three examples are compared in their results. For easier calculations the base two logarithm is used. Sample one: 2 apples, 2 peaches, 4 pears and 4 bananas

Entropy calculation: $-\frac{2}{16} * \log_2(\frac{2}{16}) - \frac{2}{16} * \log_2(\frac{2}{16}) - \frac{4}{16} * \log_2(\frac{4}{16}) - \frac{4}{16} * \log_2(\frac{4}{16})$
-> $(-\frac{2}{16} * -3) + (-\frac{2}{16} * -3) + (-\frac{4}{16} * -2) + (-\frac{4}{16} * -2)$
Entropy value: $\frac{28}{16}$ Sample two: 4 apples, 4 peaches, 0 pears and 4 bananas

Entropy calculation: $-\frac{4}{16} * \log_2 frac416 - \frac{4}{16} * \log_2(\frac{4}{16}) - \frac{4}{16} * \log_2(\frac{4}{16})$
-> $(-\frac{4}{16} * -2)(-\frac{4}{16} * -2) + (-\frac{4}{16} * -2)$
Entropy value: $\frac{32}{16}$ Sample three: 8 apples, 2 peaches, 2 pears and 0 bananas

Entropy calculation: $-\frac{8}{16} * \log_2(\frac{8}{16}) - \frac{2}{16} * \log_2(\frac{2}{16}) - \frac{2}{16} * \log_2(\frac{2}{16})$
-> $(-\frac{8}{16} * -1)(-\frac{2}{16} * -3) + (-\frac{2}{16} * -3)$
Entropy value: $\frac{20}{16}$

Between sample one and two the entropy becomes higher, because the elements focus on fewer categories. Between sample two and three the value decreases, because the same amount of categories is distributed less evenly. The Shannon or sample entropy applied in this thesis is used on the categorical variables source and destination IP or port. Those two cases are both very helpful because in case of an attack most of the IPs or ports used focus on a small group in comparison to the regular data points. Additionally the paper of Lakhina, Crovella and Diot [LCC05] suggests that the value also increases with a larger set size, which is useful for detection of large traffic in network security. To smooth the results the entropy value is calculated on a 5-minutes environment for the entropy value of each minute. How much these features help to detect real attacks even in uncertain environments is further discussed in this thesis.

3. Related Work

As seen in the basic chapter 2 this thesis focuses on two different anomaly detection methods used in a network security dataset. Related to these approaches are several previously done papers and introductions.
The STL method was first introduced by Cleveland [CCMT90] in 1990 as a basic model for seasonal trend decomposition. At that time the method was used for statistical analysis and prediction in ecological or economical datasets. The example dataset of the original paper was a decomposition of carbon dioxide measurements as shown in chapter 2.5.1. From this statistical model the STL methods developed to a useful tool in many applications with patterns in the analyzed dataset. For example a recent analysis of the dynamics in land surface vegetation was done by using the STL method on the leaf area index (LAI) [JLWX10]. They compares the STL with two other models (DHR and SARIMA). Their result was that the STL model is more sensitive to noise in the dataset than the other methods. This can be useful for the netflow dataset, since there is a lot of noise included. Another recent publication [CTB+12] about the STL method used social media analysis to gather insight into local events and movements of people. The data was derived from Twitter, Youtube, Flickr and similar applications. However these datasets have a lot of noise and unimportant geographical information. Therefore the STL method was used to exclude the global seasonal behavior and concentrate on the anomalous remainder values. In figure 3.1 the analysis of those datasets is shown as a comparison between the remainder values of the dataset and the original data. The important events can be seen in all points with remainder values (blue) that reach or exceed the original data values (red).

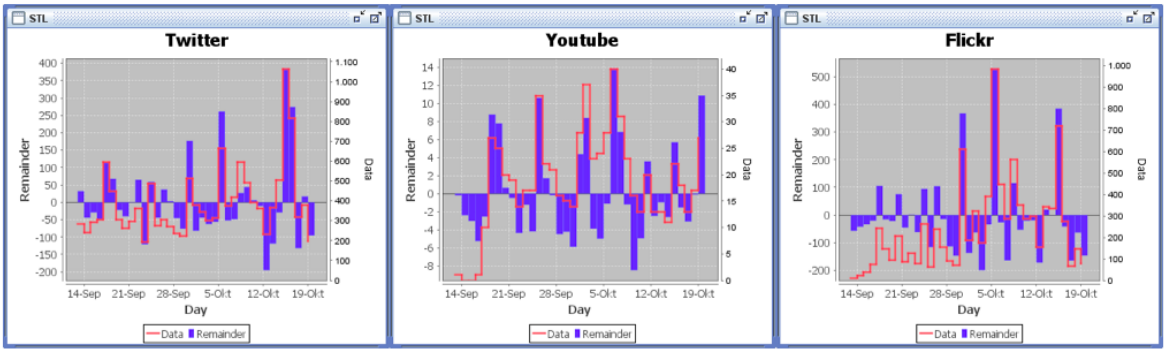


Figure 3.1.: STL decomposition - Social media analysis

In the field of network security STL is not commonly used, since other anomaly methods such as clustering and distance based methods are more established, but since the network connections mostly have strong pattern behavior they are highly suitable for an STL decomposition.

The Shannon entropy on the other hand has already been applied to a network security dataset. The entropy is based on work by Shannon and Weaver [SW49]. It was used on information theoretic measurements in the 2001 IEEE Symposium on Security and Privacy [LX01]. Based on the concepts developed in that work the Shannon entropy was used in network security datasets in 2005 [LCC05] with very good results. In this work by Lakhina, Crovella and Diot a similar comparison to this thesis was made between the entropy results and the volume-based results differentiated by the attack type. The entropy method was far more successful in this context. In figure 3.2 a port scan is detected with the help of the entropy values. While figure 3.3 shows the comparison of the entropy values with volume based anomaly detection values. Both pictures show the efficiency of entropy values and indicate how anomalies can already be interpreted by the anomaly timelines without knowing the real datasets.

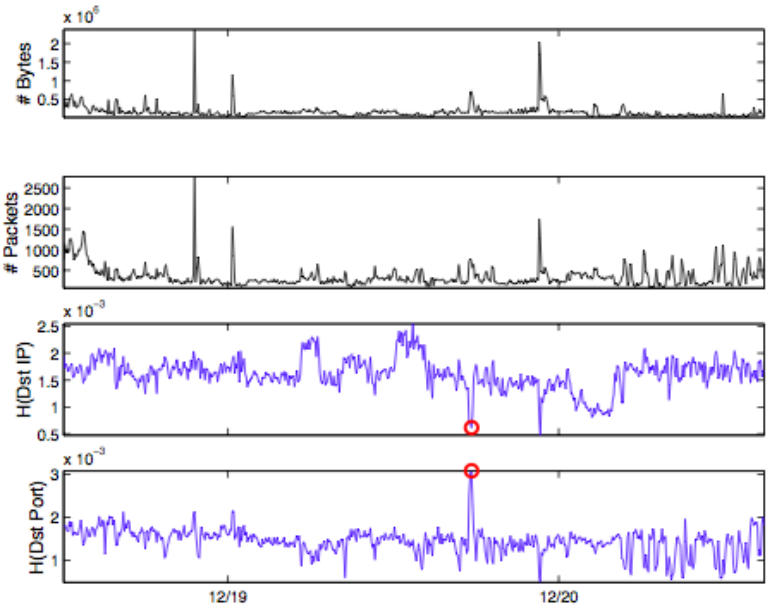
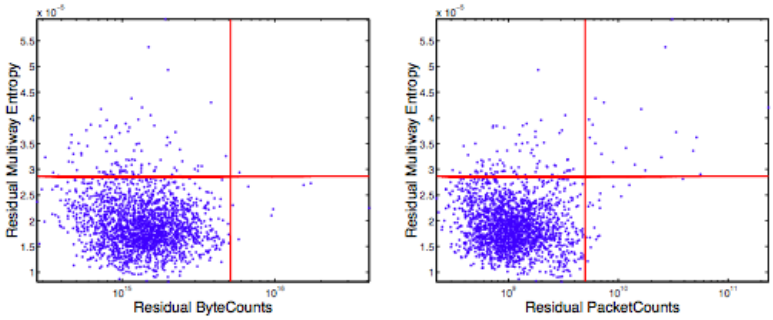


Figure 3.2.: Entropy - Port scan anomaly

Later in 2010 entropy based methods were again used in the context of multivariate time series mining where it replaced the pure traffic volume data with its more granular insight [HHZ10].
Other methods than those discussed in this thesis have been used on network security datasets before. Some very common anomaly detection methods are clustering methods like K-means clustering and Gaussian Mixture Model. For example in a recent paper [LLLS13] the combination of fuzzy clustering and Gaussian Mixture Model was used on intrusion detection datasets. For the more complex application to real time intrusion detection an anomaly detection by using change point outlier detection is suggested by Naveen, Natajaran and Srinivasan [NNS12]. Another quite novel concept uses lossless compression instead of statistical premises to enable information theoretic outlier detection [WHF12]. Although very powerful these concepts were



(a) Entropy vs. # Bytes (b) Entropy vs. # Packets

Figure 3.3.: Entropy - Comparison with volume analysis

not chosen for this thesis because they mainly work with intrusion detection datasets, which were only a minor factor in the VAST 2013 dataset.

4. Anomaly detection

In this thesis two very different anomaly detection methods are used on a network security dataset and then compared in their efficiency and characteristics. Therefore this chapter shows, why these methods were chosen, how the dataset had to be adapted for the usage and which implementations were necessary for the calculation of both anomaly detections. How the two methods Shannon entropy and the STL decomposition are applied is described in this chapter. The calculation of the entropy values was done in a moving window preprocessing in Python. For the STL decomposition the same Python program was used for an accumulation on which is later used a STL library in R to calculate the decomposition.

4.1. Method decision

The decision to use those two anomaly detection methods was based on two very different reasons. While the Shannon entropy is a method already used successfully on network security data but otherwise not very common the STL methods is already quite established, but was not common in the field of network security datasets. Of both methods the Shannon entropy was first included into the system for the VAST 2013. Searching for a clustering method for IPs and ports a paper of Lakhina, Crovella and Diot [LCC05] showed very promising visualizations of 3D clusters. Having read of the methods for those clusters by calculating the Shannon entropy, this method showed promising features for a direct insight into the anomalies of the dataset. Therefore it was integrated into the overview visualization of the newly implemented system. At first it was just calculated as one value for each minute. However one minute proved a much to small sample to calculate a representative entropy value. Thus the calculation was further improved by using a moving time window of five minutes for each minute in the dataset. On the other hand the STL method for decomposing datasets was already known when the system was implemented, but never considered appropriate for the data provided. Later in the course of implementation many regular patterns like a day and night pattern were distinguishable in the entropy values. That made using the STL decomposition an option, although it had not been used on network security data before. Later in this thesis both methods are compared by various parameters. First a brief introduction of the calculation and implementation of both methods is given in the following chapters.

4.2. Preprocessing

For creating the entropy and STL datasets, the netflow datasets over both weeks are used. To be able to calculate the values in one loop for each week all data of the first week is concatenated into one file. Therefore one file week1.csv and one file week2.csv are left for the calculation. Since the algorithms is working in time based loops, the data files need

to be sorted. After these preprocessing steps the python program is used to create the accumulated datasets. To achieve best comparability the moving window algorithm is used for both approaches, although it is only necessary for the entropy algorithm. The moving window means that all data entries in the five minute window surrounding the data point are accumulate and consider . Since the original data files have been concatenated, all header entries in the file must be skipped. The central part for the preprocessing is the moving time window. The algorithm loops through all entries and accumulates until a five minute data piece is complete. Then this data is put into a calculation function, whose code contains all steps needed for one of both anomaly detection algorithms. This is shortly shown as a pseudo code demonstration.

```
MovingWindow <- 5 minutes
Step <- 1 minute
StartTime <- Time of first entry
EndTime <- StartTime + MovingWindow
For all data:
        if (currentTIme<EndTime) -> save data to this window
        else ->
                calculate and save value for this window
                StartTime +1
                EndTime +1
                Clear window data
```

4.3. Entropy

The function for the entropy dataset uses the algorithm described in 2.5.2. It accumulates the count of each IP/port from source and destination. Afterwards it sets those numbers into relation to the number of entries as well as the number of different IPs/ports. To do this, a function iterates through all data point of one given 5-minutes dataset. The function then separates and counts all different source and destination IPs and ports. The result are four lists which each contain key value pairs with the name of the IP or port and its count in the dataset. These values are then entered into the entropy formula as described in the entropy introduction of chapter 2.5.2. The function is shown in a short pseudo code example:

```
new Key-value List for sourceIP
new Key-value List for destinationIP
new Key-value List for sourcePort
new Key-value List for destinationPort
For each data entry:
        if (sourceIP is unknown) -> create a list entry with this key
        Increase value of this sourceIP key by 1
        if (destinationIP is unknown) -> create a list entry with this key
        Increase value of this destinationIP key by 1
        if (sourcePort is unknown) -> create a list entry with this key
        Increase value of this sourcePort key by 1
        if (destinationPort is unknown) -> create a list entry with this key
        Increase value of this destinationPort key by 1
For each key-value list:
        N <- Sum of all values in this list
```

```
16          Entropy <- Sum over all values with (-value/N*log2(value/N))
17          Save entropy of this list
```

As described in the basic chapter 2.5.2 the sets are first analyzed for the amount of connections for each IP and port. Then the adapted entropy formula is used to get a normalized measurement of concentration. These values are later displayed along a timeline and anomalies in some minutes for one of the lists are represented visually.

4.4. Seasonal trend decomposition

4.4.1. Python accumulation

The function for the STL dataset only accumulates the variables for each five minutes data piece. The creation of the season, trend and remainder data is later used on the resulting dataset in a R application. In the function the given 5-minutes data package is iterated and each variables values are summed up for this time period. The function is shown in a short pseudo code example:

```
1  For each variable define:
2          variableSum <- 0
3  For each data entry and each variable:
4          variableSum + newVariableValue
5  Save all variableSums
```

The variable in this pseudo code is an example for any variable to be used. The real code includes seven different variables. After the timelines are created, they are put together into the following packages:

- Entropy
- STL - Bytes
- STL - Payload
- STL - Packages
- STL - Overview and Duration
- STL - Bytes Subnet Remainder
- STL - Bytes Subnet Trend
- STL - Duration Subnet Remainder
- STL - Duration Subnet Trend
- STL - Packages Subnet Remainder
- STL - Packages Subnet Trend
- STL - Payload Subnet Remainder
- STL - Payload Subnet Trend

4.4.2. R implementation

To create the above mentioned packages the accumulated dataset was read into an R program and then used in the R library for STL functions. The resulting STL has then again been exported into CSV documents. The following pseudo code describes how the final version of the STL usage with R is done:

```
Read accumulated datasets of both weeks separately
For each variable and dataset:
        Go through week 1:
                Save values of each day to a vector
        Change full vector to time series with period 1440
        Calculate and save STL week 1 from time series
        Go through week 2:
                Save values of each day to a vector
        Change full vector to time series with period 1440
        Calculate and save STL week 2 from time series
        Combine and save STL values of this variable
```

For each of the twelve resulting packages the gap between day seven at 9.02 am and day ten at 6.51 am had to be added first. Also the calculations which were created at the previously added fake entries had to be deleted. This does not influence the data, since it is only an artificially created minimum at the end of each week that is removed. Before exporting the packages, the time format has to be adjusted to the one used by our tool, which is "yyyy-mm-dd hh:mm". After that the package was exported as a csv file and converted to a json-array. Then it replaced the previously used entropy data file for the overview timelines in AnNetTe.

4.4.3. Options on seasonal decomposition

To use the final accumulation dataset for a time series variable in R, a few fake entries have to be added to each week. This is necessary because the number of entries needs to be a full multiple of number of entries in one day. The dataset of the first week consists of 8640 entries for 6 days and the file for the second week of 7200 entries for five full days. The required period for the time series is one day which means 1440 entries. Using one hour was also an option, but since there is no seasonal behavior in each hour of the dataset the STL results were not useful. After creating the first calculations some problems occurred, because of a two days gap between the two weeks of the dataset. If the STL decomposition was created of one file containing both weeks, there would be remainders and anomalies at the time of the gap. These are in some way correct, since the gap is an anomaly, but would be very confusing for the user. That leaves three options for calculation which are shortly described below.

Option 1 - both weeks

Option one is the original result of calculating the decomposition of both weeks with anomalies in the gap. This is shown in figure 4.1, where the gap is filled with high outliers at all times. Knowing the dataset this is impossible from a real event view. In order not to confuse the user this option was discarded.

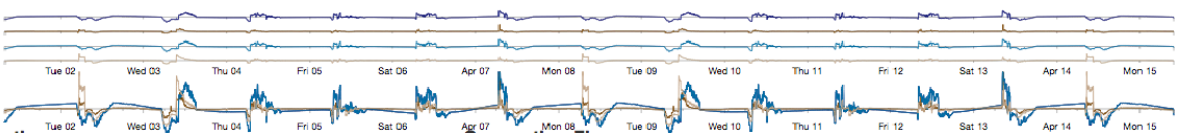


Figure 4.1.: STL decomposition - Option 1

Option 2 - both weeks

Option two is the same calculation as in option one but with a recreated gap between day seven and ten. This is shown in figure 4.2, where the gap is visible again. In comparison to figure 4.3 the anomalies in both weeks behave differently. That is the result of the background computation. Although the gap now exists again the anomalies are still influenced by the original consideration of the gap in the seasonal behavior. Therefore this option might suggest different results than intended and was discarded as well.

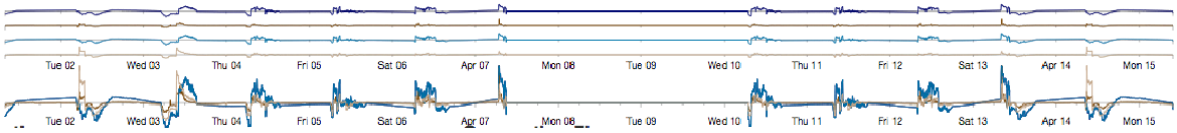


Figure 4.2.: STL decomposition - Option 2

Option 3 - combination of single weeks

Option three uses the algorithm for each week and then combines them to one data file. This keeps the seasonal data reasonable and avoids the anomalies in the gap. The code for this option is used in the process described before. A small preview is given in figure 4.3, where the final version of the STL presentation in AnNetTe is show. The different features in comparison to the previous is the result of a later stage in the overall implementation of AnNetTe.

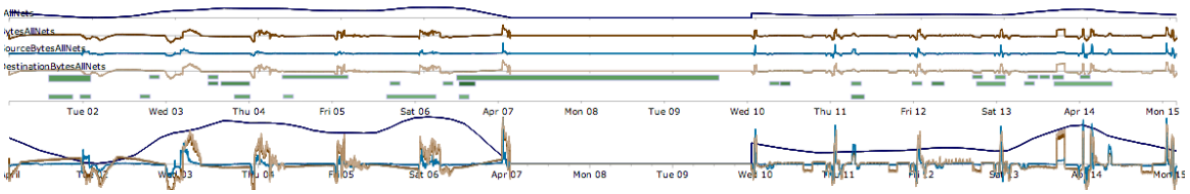


Figure 4.3.: STL decomposition - Option 3

5. Description of VAST 2013 Solution

To evaluate the two methods for anomaly detection and make them comparable to each other they have to be integrated into a system and applied to a benchmark dataset. In the case of this thesis the benchmark dataset was the VAST 2013 Mini challenge 3. The system used for the comparison was implemented in the course of this thesis' project work and used for solving the given benchmark dataset as a participation to the challenge. In reference to the benchmark dataset description in the basic chapter 2.4 the system in which the anomaly detection methods are integrated and the challenges' task is solved is described. The description should show in detail how the anomaly detection methods are used to help the user find events in the dataset. At the same time an overview of the whole system and the visualizations and interaction methods used should be given. This overview helps understand what role the anomaly detection plays in the overall system and how it connects to other parts by the users interaction. For those readers interested in a detailed description of the visualization, interaction and its conception, all this information can be found in the parallel written thesis [Mer13]. This thesis only describes how the three level visualization consisting of an overview timeline (see chapter 5.1), a ring graph (see chapter 5.2) and a connection river (see chapter 5.3) works together in one interaction pipeline (see chapter 5.4)

5.1. Overview timeline

The overview visualization of AnNetTe shows a number of different values along the time of the two weeks which were provided by the dataset. There are 3 different types of timelines. The first part of the Overview consists of 4 static timelines representing different variables from the real dataset. The second part of the Overview shows 4 calculated anomaly detection timelines. These 4 timelines are a package with related variables and can be changed to other packages by the user. The third part of the overview provides a zoom function. For the time period selected in the timelines above this part shows the details of all 4 anomaly timelines chosen by the user. The timelines in the detailed view are overlaying to provide easier comparison.

5.1.1. Static timelines

The four static timelines provide information about accumulated values for each minute. The values provide information from all three different datasets. A zoomed version of the overview visualization is shown in figure 5.2. The four lines presented in that figure are afterwards explained shortly.

Health Status The first timeline shows the sum of all status values reported to the health monitor in that time frame. The values are shifted from 1-4 to 0-3 to prevent high sums of healthy (1) values.

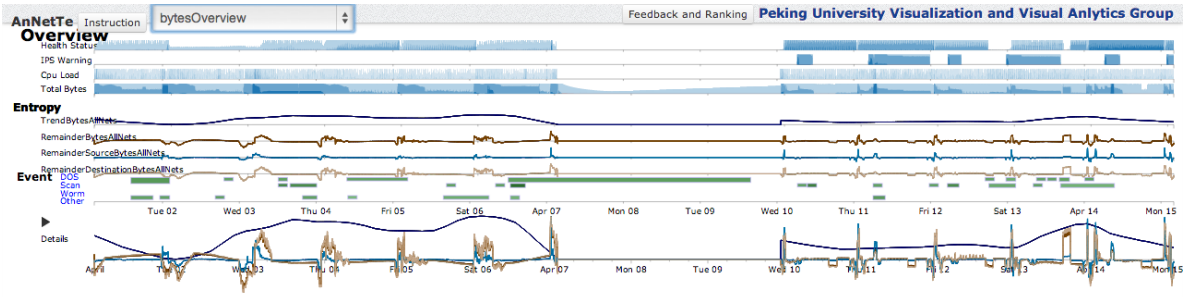


Figure 5.1.: Overview Timeline - STL of bytes timeline

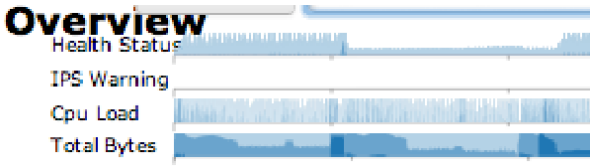


Figure 5.2.: Overview Timeline - Static timelines

IPS log The second timeline shows the count of IPS log entries with priority warning, that refer to a denial of the connection described in the log. This dataset is only available in the second week, but the line is still drawn over both weeks for consistency.

CPU load The third timeline shows the sum of all CPU loads reported to the health monitor in that timeframe. This value was chosen above others because of its strong relation to DOS attacks.

Transferred Bytes The fourth timeline shows the sum of all transferred bytes from the netflow dataset. The variable was chosen because of its strong relation to network scan attacks.

5.1.2. Dynamic timelines

The dynamic timelines represent the visualization of the anomaly detection. Here the different methods are used on the two weeks dataset and the resulting values are displayed as timeline. To compare them the user can choose between the following packages of four different timelines:

- Entropy
- STL - Bytes
- STL - Payload
- STL - Packages
- STL - Overview and Duration
- STL - Bytes Subnet Remainder
- STL - Bytes Subnet Trend
- STL - Duration Subnet Remainder
- STL - Duration Subnet Trend
- STL - Packages Subnet Remainder

- STL - Packages Subnet Trend
- STL - Payload Subnet Trend
- STL - Payload Subnet Remainder

Entropy package The entropy packages consists of four entropy lines. The first two show the entropy of the destination IPs and the source IPs. The third and fourth line show the entropy of the destination port and source port. The preprocessing and calculation for those lines was described in chapter 4.3. In figure 5.3 these entropy lines are shown integrated into the complete overview timeline. In contrast to the later shown STL timelines the entropy values are all positive.

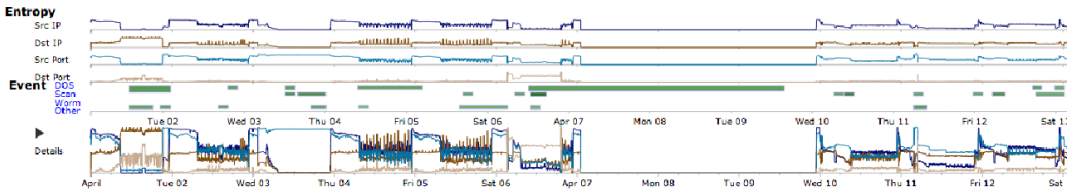


Figure 5.3.: Entropy package selected in the overview visualization

Multivariable packages The multivariable packages are all built after the same pattern. The bytes, payload and packages lines each consist of one trend line for the combined variable and three remainder lines for the combined variable, the source variable and the destination variable. Since the duration has only one variable field it is displayed as one line of the duration remainders put together with all three combined variables' trend lines. In figure 5.4 this combination is presented. The meaning of each line can be read at the left side, while the details are visible in the lower timeline.

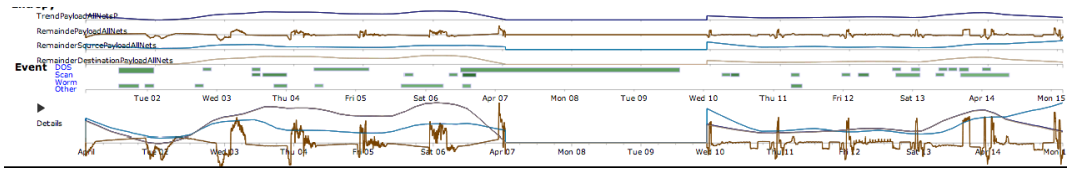


Figure 5.4.: Multivariable payload package selected in the overview visualization

Subnet packages The subnet packages are all built after the same pattern. First they show the trend of all subnets together then they are differentiated into trend and remainder packages. The remainder packages shown in figure 5.5, visualizes each subnets' remainder in the last three lines. The trend packages shown in figure 5.6, visualizes each subnets' trend in the last three lines.

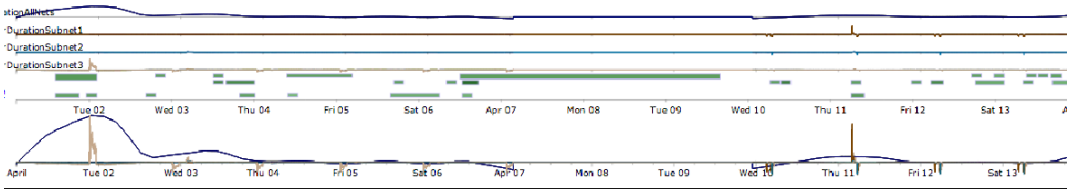


Figure 5.5.: Subnet duration package with remainders selected in the overview visualization

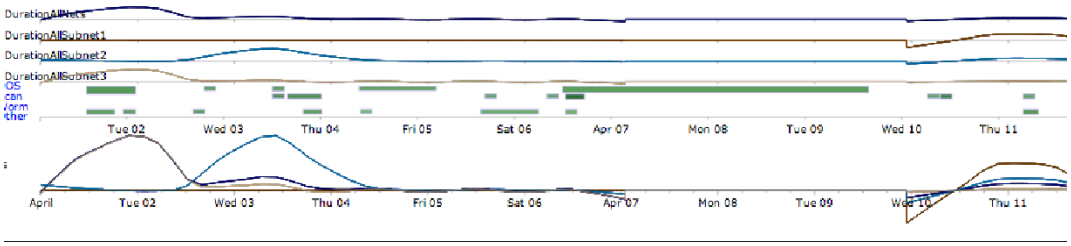


Figure 5.6.: Subnet duration package with trends selected in the overview visualization

5.1.3. Detailed timelines

The detailed timelines show the selected dynamic timelines, focused on the selected time range. The relationship between different timelines of one package can be seen easily, since the function graphs are all integrated into one coordinate system. The colors of the lines represent the same colors as in the dynamic timelines for easier recognition. Inside the detailed timelines the user can further specify the timeframe he is interested in. After selecting this focus time range, the tool provides the connection graph of all IPs in this timeframe. Start and end time are also shown as labels to the connection graph as well as inside the detailed timeline. The selection can either be created new, or adapted by resizing the selection window.

5.1.4. Event markers

In addition to the general overview over provided data, the user can later see markers above the time lines at points in time where events were already committed. The events selection can be reproduced by clicking on the marker. After that the user might leave comments and agree or disagree with the information.

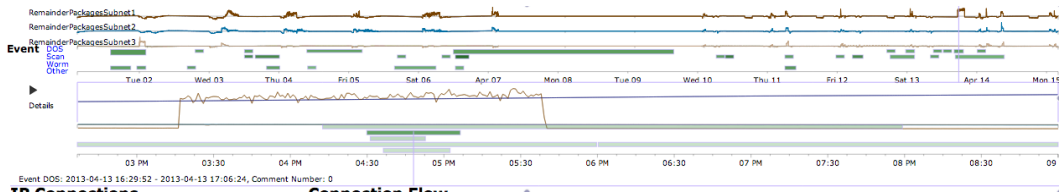


Figure 5.7.: Detailed timeline including event markers with one chosen event

5.2. Connection graph

The ring graph shows the connections of the selected time structured and grouped by subnet (Figure 5.8). The color of each connection and ring part is used to encode the subnet as well as the source of each connection. Since all internal IPs have a blue color scheme and all external ones a scheme ranging from green to red, it is easy to determine which kind of connection is shown. For more clutter reduction IPs can be excluded from the graph by interaction. For a faster overview the user can play an animation of the ring graph over the time he selected. After deciding on an event he selects the IP group and the level three visualization of that group is shown. More details on this visualization and its interaction can be read in the parallel thesis [Mer13].

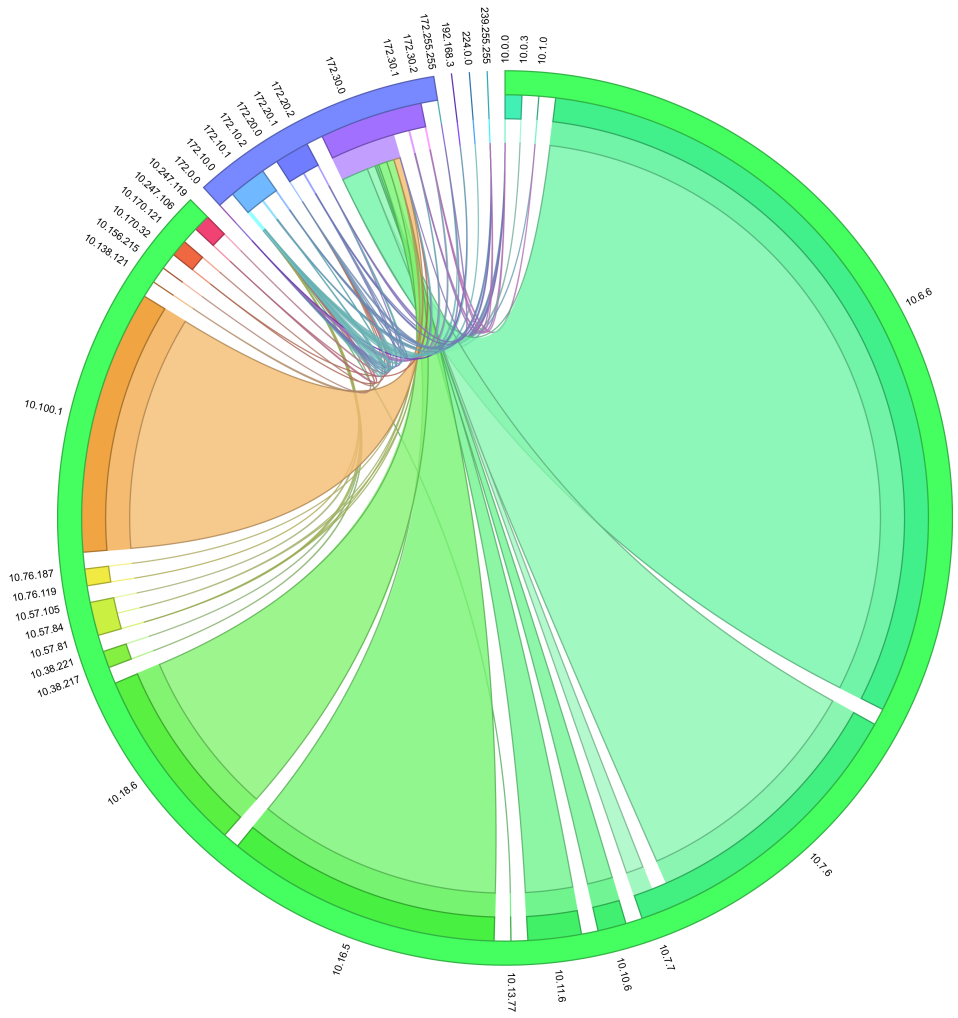


Figure 5.8.: Connection graph - Example of connection graph

5.3. Connection river

The connection river shows all details as one fingerprint adapted from a parallel coordinates view. The view consists of three axes for source and destination each. The first axis shows the IPs the second the connection time and the third the ports used in the connection. As you can see in figure 5.9 each line represents a connection flowing from the three source axes to the symmetrical destination axes. Between the two port axes one of the connection variables is displayed as the height of the flow lines. This variable can be chosen from the icons displaying a preview of each of the variables' connection river below the visualization (Figure 5.9). The bar of colored squares besides the river shows the relative stage of each accumulated health attribute for internal IPs. For each IP and time connection that has been logged in the IPS log a red line is drawn in the connection river. This line ends at the port axis to signify the denial of this connection. In the connection river the user can exclude IPs just as in the ring graph. If he selects one IP only this IP is left in the river view and all others as well as their destinations are excluded. This way the user can focus the visualization to the data of his event and in the end has a singular fingerprint of his data. More details on this visualization and its interaction can be read in the parallel thesis [Mer13].

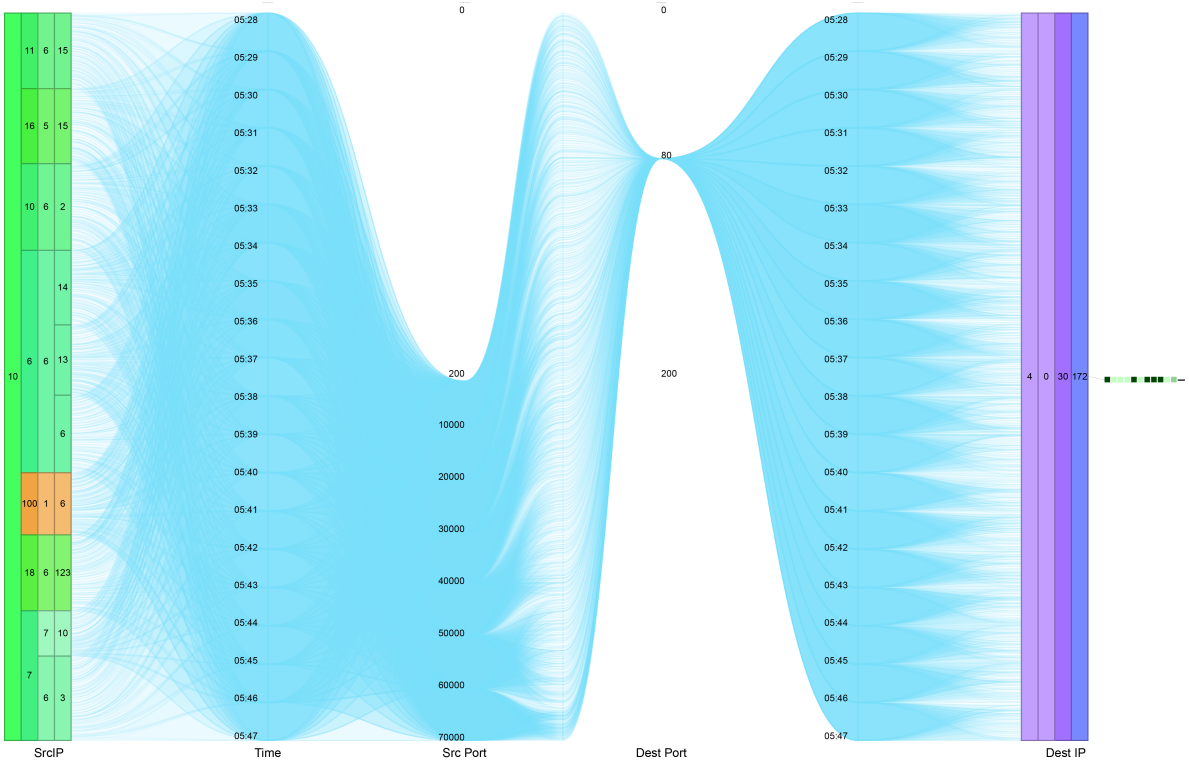


Figure 5.9.: Connection graph - Example of connection river

5.4. Interaction pipeline

To find events with the help of AnNetTe there is a clear interaction pipeline. Steps are to be taken in order but can be modified later. This pipeline is shown at the example of a port scan attack found in the VAST dataset. With the help of AnNetTe more than 12 events were identified. The most common event types are DOS attacks, network scans and port scans. The steps to find event 11 (Figure 5.10) with the help of characteristic functions in AnNetTe are shown below.

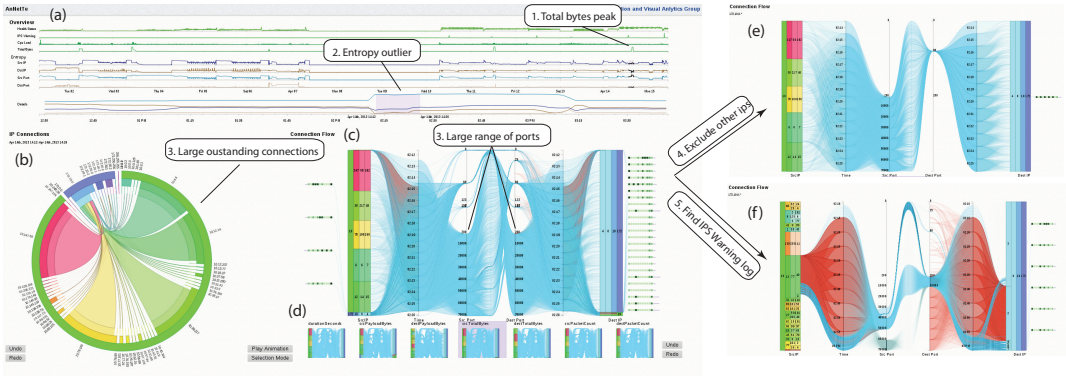


Figure 5.10.: Summary of event 11 using all visualizations of AnNetTe

Step one: Decide time of interest First the detailed time is not adapted to a timeframe and can not be select directly. To find an event we search for peaks in one of the 8 timelines (Figure 5.10 a). In this case we see a high value in the total bytes timeline.

Step two: Choose reasonable attack window In the detailed view we currently use the entropy package as our anomaly detection. We can see a source IP and source port outlier at the same time. We then select this anomaly.

Step three: Choose suspicious IP In the ring graph (Figure 5.10 b) we see five dominant external IPs. We can exclude other IPs to concentrate on this part. Then we select those remaining five IPs as the reference selection for the connection river.

Step four: Verify event with details Looking at the river (Figure 5.10 c) we see the many source ports in the port axis and the high number of bytes displayed as the higher distance between the connection lines. We also see some red lines. Therefore we separate the river. First we select only the five dominant IPs and see the DDOS attack with high unhealthy status of the victim (Figure 5.10 e). On the other hand we select the IP with red lines and see a port scan attack on one server of company part one (Figure 5.10 f). It shows the typical high number of destination ports and the IPS log. Also it has a healthy destination IP which would not be the case in an DOS attack.

Step five: Submit Event and comment After being sure about the event the user can save his results by submitting the event to the database. The event is then shown as a new marker in the timeline. To do this he can click on the submit event button and a submission screen opens. Data about the selected timeframe and IPs as well as a picture of the connection river showing all the details are already included. To finish the submission the user should provide his name, his level of expertise, the type of event as well as some textual description. Other people can later click on the event market in the timeline to instantly choose all the parameters of this event. They then can leave comments and agree or disagree with the chosen event.

6. Evaluation

The main focus of this thesis is the comparison and evaluation of the STL and the Shannon entropy anomaly detection methods. After introducing to those concepts, presenting the process of implementing those methods and giving a overall picture of the benchmark and system those methods are used for, those two approaches can now be analyzed by various parameters. In this chapter all three parameters for evaluating the methods are shortly explained and afterwards all results of those evaluations are given. In general the methods were evaluated by the quantity of anomalies, the quality of those anomalies and the usability of these features integrated into the system AnNetTe.

6.1. Steps of evaluation

The evaluation takes place in three different steps. First the amount of anomalies is determined for each methods. Then those amounts are categorized into attack types and false or true alarms. In the end the usability of both methods is tested in a small study with nine participants.

6.1.1. Quantity of anomalies found

The first step of the evaluation is to measure the quantity of anomalies detected with each methods, when used on the benchmark dataset of the VAST 2013. To measure this the variables to be used in each measurement need to be decide. As described in chapter 5.1 the entropy method is used on source and destination IP and ports which are then combined in one visual timeline package. For the STL methods the connection variables for bytes, payload, packages and duration are used for source and destination each. Another distinction is between the overall STL which delivers trend and remainder timelines as well as the STL method used on each subnet of the network which can then also be split into trend and remainder. For each of the packages described in chapter 5.1 the number of anomalies is determined. The anomalies are grouped into positive outliers, which are mostly peaks in the remainders and negative outliers, which appear in the remainders, but could also be a negative trend. Each of the two weeks is analyzed separately, so get more detailed comparisons between the methods, since accumulation often hides more subtle differences. The results of this analysis are discussed below in section 6.2.1.

6.1.2. Categorization of anomalies found

The second step of evaluating the efficiency of both anomaly detection methods is to categorize the anomalies and evaluate their type and quality. All packages of the first evaluation are again analyzed in this step. The anomalies are first distinguished between false alarms and real attacks. This should either emphasize the methods with a higher number of results or

nullify their advantage by only providing additional false events. Apart from this verification of previous results the type of attack is noted for each event. By doing so reasons for different anomalies or different numbers of results are investigated. Effects such as a high reliability in one type of attacks but low results in another type could explain better results for the methods that by chance focuses on the attack types in the given benchmark. The types differentiated in this evaluation are DOS, DDOS, Port Scan, Network Scan, Worms and SMTP attacks.

6.1.3. Study about usability of both methods

In a third step a very different characteristic of the methods is evaluated. For an efficient usage the understandability and usability of the concept by the user is essential therefore both approaches have been integrated into a small user study. The study consists of two parts, one where the participants solve questions with the help of the system AnNetTe by using both methods and one where they evaluate their experience with the system and both methods. For the usability questions part of the "Computer System Usability Questionnaire" was used. For other evaluation questions multiple choice questions and free text questions were asked. The main purpose was to find out, which concept is faster and easier to use by experts and new users. For multiple choice questions the methods were distinguished between entropy method, STL methods and STL methods with subnet distinction. The full questionnaire and study results can be found in the appendix A. An overall conclusion with the major points are presented below in section 6.3.

6.2. Results of anomaly analysis

The results of the analysis of resulting anomalies for each method are divided into two different viewpoints. One analysis compares the amount of anomalies detected for different time section and different anomaly types. The second viewpoint shows the anomalies categorized by false and true alarms as well as by attack types.

6.2.1. Resulting amount of anomalies

The analysis of the quantity of resulting anomalies indicates a very similar efficiency between the concepts. While the entropy method detected more outliers in the first week, the STL detection found more in the second week. The different relation from negative and positive outliers and between both weeks are further analyzed in the qualitative analysis. For the different variables used in the STL methods large differences can be detected. Especially the analysis of duration value detects less than half as many outliers as the others. A differentiation into the three subnets brought two different results. For the remainders of all subnets the anomalies were mostly identical with the overall variables. Only the Packages Variable was getting more effective by using all three subnets. For the trend values less than half of the overall anomalies could still be detected. Whether this loss of information is due to less efficiency or less false alarms are discussed in the qualitative analysis.

Quantity	**Negative Anomalies Week 1**	**Positive Anomalies Week 1**	**All Anomalies Week 1**	**Negative Anomalies Week 2**	**Positive Anomalies Week 2**	**All Anomalies Week 2**	**All negative Anomalies**	**All positive Anomalies**	**All Anomalies**
Entropy	5	9	14	7	9	16	12	18	30
STL - Bytes	5	6	11	9	8	17	14	14	28
STL - Payload	4	6	10	8	11	19	12	17	29
STL - Packages	4	6	10	8	11	19	12	17	29
STL - Overview and Duration	5	1	6	3	2	5	8	3	11
STL - Bytes Subnet Remainder	5	6	11	9	8	17	14	14	28
STL - Bytes Subnet Trend	2	3	5	3	3	6	5	6	11
STL - Duration Subnet Remainder	5	1	6	3	2	5	8	3	11
STL - Duration Subnet Trend	2	2	4	3	3	6	5	5	10
STL - Packages Subnet Remainder	7	7	14	7	10	17	14	17	31

Quantity	**Negative Anomalies Week 1**	**Positive Anomalies Week 1**	**All Anomalies Week 1**	**Negative Anomalies Week 2**	**Positive Anomalies Week 2**	**All Anomalies Week 2**	**All negative Anomalies**	**All positive Anomalies**	**All Anomalies**
STL - Packages Subnet Trend	2	3	5	2	2	4	4	5	9
STL - Payload Subnet Remainder	4	6	10	8	11	19	12	17	29
STL - Payload Subnet Trend	2	2	4	3	3	6	5	5	10

Table 6.2.: Results for quantitative evaluation

6.2.3. Results of categorization of anomalies

The analysis of the quality of results gives a clearer insight into the differences between the methods. STL and Entropy both produce false alarms for a third of the anomalies. With ten instead of eight false alarms the entropy method is even less reliable than less accurate variables of the STL method. On the other hand the entropy method is the only one, that found a hidden SMTP activity in two different cases. In case of events that do not belong to the standardized attack types, the entropy methods is also more thorough than the STL methods. Only the bytes variable and the subnet SZL of the packages recognize as many undefined events as the entropy methods. The same situation occurs for the worm event which can only be found by the entropy, the packages Subnet and the bytes SLT methods. Neither of the methods found a port scan attack, which could simply be indicating that such an event was not contained in the dataset. Excluding the duration variable and the trend lines, the STL method found more DDOS attacks and network scans than the entropy method. Still this result is not certain. Some events are only indicated at their start point in the entropy dataset, while they appear multiple times in the STL datasets. Which of these behaviors is better cannot be directly judged. If a network analyst has to find events with the least possible uncertainty in the anomalies only one outlier might be enough. If on the other hand a lot of uncertainty is integrated in any case, multiple reminders on the same event might be helpful not to overlook anything. The last attack type is a single DOS attack. In each of these methods only one DOS was found, which indicates that there is only one such event hidden, but also confirms the ability of all methods to show this attack type.

6.2.4. Results for qualitative evaluation

Quality	**DOS Attack**	**DDOS Attacks**	**Network Scans**	**Port Scans**	**Worms**	**Others**	**SMTP Activity**	**False alarms**
Entropy	1	3	6	0	1	7	2	10
STL - Bytes	1	5	6	0	1	7	0	8
STL - Payload	1	5	10	0	0	5	0	8
STL - Packages	1	5	10	0	0	5	0	8
STL - Overview and Duration	1	1	5	0	0	2	0	2
STL - Bytes Subnet Remainder	1	5	6	0	1	7	0	8
STL - Bytes Subnet Trend	1	1	5	0	0	2	0	2
STL - Duration Subnet Remainder	1	1	5	0	0	2	0	2
STL - Duration Subnet Trend	1	2	5	0	0	2	0	0
STL - Packages Subnet Remainder	1	4	9	0	1	7	0	9
STL - Packages Subnet Trend	1	2	5	0	0	0	0	1
STL - Payload Subnet Remainder	1	5	10	0	0	5	0	8
STL - Payload Subnet Trend	1	1	6	0	0	1	0	1

Table 6.4.: Results for qualitative evaluation

6.3. Results interaction analysis

The usability of the different methods as well as of the whole system was evaluated by a small study. The details of that study can be seen in the Appendix A. All results are presented in the following chapter.

6.3.1. Reliability of results

The third analysis had a very difficult realization and should be considered carefully in its results. Of the planned twelve participants only nine students took part in the study. This was mostly due to time problems. However the other nine participants all took part in the study for the whole 1,5 hours. The study was originally planned with 5 question in 40 minutes. This time was largely exceeded, since the participants needed much more time to get used to the system and understand the meaning of the visualizations. After 1,5 hours the study was finished with a general assessment of the first three questions and some answers to question four and five. Afterwards the questionnaire was filled in by all participants. Although the whole study was accompanied by three experts of the system many questions were not asked but instead noted in the answers in the questionnaire. This leads to some contradictory results like people saying they found most attacks with the STL anomaly detection, but then writing in another question, that they do not know what the STL packages are and could therefore not use it. The most probable explanation is, that they have just never heard of the name STL while working with it, but only remembered the packages variable or the trend category. Even though this is a reasonable explanation the study itself cannot be verified like this since it is anonymous. This means all results have to be considered unreliable to some extent.

6.3.2. Usability

The usability questionnaire was very useful in order to have a standardized measurement. From the original 19 questions twelve were asked in the study. The average value shows, that people had difficulty in learning to use the system. This notion is also proven by the length of the study which by far exceeded the planned time span even with a reduced number of tasks. The second lower value is the pleasantness of the interface, which should be considered for the overall work, but is not very relevant for this study. The best results had questions about becoming productive in a short time and finding information easily. Both points could be in part considered as positive feedback for the anomaly detection which gives a fast and clear entry point to the system. As seen in the interaction pipeline, the choosing of an appropriate time by considering the timelines is essential to finding an event. The later evaluated ring graph and connection river both have a more confirming and extending character for the information finding. Overall the usability is not judged too well. This might have reasons in the short time and in missing expert knowledge. On the other hand it is also valuable feedback about the interaction in the system and to provide better step by step explanation while using the tool.

Interaction Points from 1 (not given) to 5 (excellent)	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6	Participant 7	Participant 8	Participant 9	Average
Overall Satisfaction	3	2	2	5	2	3	5	3	5	3,33
Simple System Usage	3	2	2	5	3	3	5	3	5	3,44
Comfortable System usage	2	5	2	5	2	2	4	3	5	3,33
Easy learning	4	1	1	na	2	4	na	4	na	2,67
Quick productivity	4	4	3	5	3	4	5	4	4	4,00
Clear tutorial	2	1	4	4	4	5	na	3	5	3,50
Information easy to find	4	3	3	5	4	3	5	3	4	3,78
Information easy to understand	4	1	3	5	4	4	4	3	5	3,67
Information effective for completing task	4	1	3	na	2	3	5	4	4	3,25
Organization of system clear	4	1	4	5	3	3	4	3	5	3,56
Pleasant interface	3	2	2	5	3	2	na	3	na	2,86
Provides all functions and capabilities expected	3	1	4	5	4	3	5	4	4	3,67

Table 6.6.: Results for usability evaluation

6.3.4. Task results

For the evaluation the first three questions of the questionnaire are considered, since they were fully participated in. The questionnaire starts with one task as an introduction to the system and helps people to learn the usage of all three visualizations. The second and third question focus on the entropy anomaly detection and the STL decomposition of the package number. The order of those two questions varies in order to get an impression about the dependence of both variables. The correct percentage for each task and order type is summarized in the following table:

	Introduction task	**Task entropy**	**Task STL**	**All**
Type STL first	89	75	75	79,66
Type Entropy first	89	100	60	83
All	89	87,5	67,5	81,33

Table 6.7.: Percentages of correct task fulfillment

This evaluation already gives the impression, that people are more efficient with the entropy anomaly detection. It is clearly visible, that the entropy tasks are better solved and also the questionnaires starting with the entropy task have a higher overall correctness. Still other factors have to be considered before the entropy can be considered the better anomaly detection method for network security. For example the higher overall correctness if starting with the entropy task, might be a time management result. The third task was the last part of the study and thus had more time pressure than the others. If the STL was not worse but just more difficult to understand, it could have resulted in the same percentages.

6.3.5. Anomaly detection questionnaire

The most valuable input was the clearly asked judgment of both anomaly detection methods. The questionnaire held some free text usability questions, which were used for improvement but are not discussed in this thesis. Additionally it held some questions about the preferences of each anomaly detection. Some questions were either differentiated by entropy, STL detection and subnet analysis with STL or they were distinguished between the remainder values and the trend values. Those questions are shown is a small table below. All free text feedback are discussed afterwards:

	Most understandable	**Most useful**	**Most meaningful**	**Most events**	**Fastest results**
Entropy	0,66	-	0,66	0,88	0,88
Overview STL	0,33	-	0,22	0,11	0,22
Subnet STL	-	-	0,11	-	-
Trend	-	0,66	-	-	-
Remainder	-	0,22	-	-	-
No answer	-	0,11	-	-	-

Table 6.8.: Preferences of methods in study

The values confirm the results of the task analysis. Between 66% and 88% prefer the entropy anomaly detection results to the STL method. In the STL method itself they prefer the overview timelines with trend and remainder combined to the subnet distinction with either trend or remainder. They also largely prefer the trend values to the remainders, which could indicate that remainders sometimes do not show the attacks that were searched for, while trend lines at least give a hint to larger events. These results should as before be viewed under the presumption that not all features were understood and some answers are contradictory. However the strong tendency towards entropy in all measurements taken is obvious. The choices in the latest measurements table were further explained by free text questions to all three methods. The full answers can be read in the appendix A. An overview of the overall impression is given now. In reference to the entropy methods most people agree, on a very effective and powerful methods. One person complained, that expert knowledge is necessary, but two other participants find it easy to learn and at least four wrote that it is useful while searching for events. On the general STL methods many participant refer to low understanding and difficult usage. Those who did understand it though also saw its powerfulness. About the subnet distinction many people didn't give comments because they couldn't use it yet or they didn't understand it.

6.3.6. Overall evaluation

The overall evaluation shows that both methods are applicable and can be used to find real events in the given benchmark. In two of three analyses the entropy method had a small advantage over the STL methods. In reference to quantity and quality they are very similar, in the quantity even better on the STL side. In the user study however the entropy method was largely preferred and more easily learned and understood. In the context of the given benchmark and system the entropy methods is by all results a better choice. However the difference is not large and due to unreliability in some parts of the evaluation and especially of the study the STL method might be better than the entropy method in another system or benchmark.

7. Conclusion

This thesis had the goal to determine whether anomaly detection methods are useful for network security analysis in Visual Analytics systems and to compare two different methods for this purpose. The first method chosen for this task was the Shannon-entropy, which is not yet established in Visual Analytics, but has already been used on network security datasets with good results. The second method is the seasonal trend decomposition (STL) which is an older method already proven in some fields, but is not commonly used for network security datasets. In addition to the simple STL decomposition of the given variables, this method was also used on each subnet. Both methods are applied in the form of preprocessing steps, by using simple python programs and in case of the STL preprocessing an R library. The preprocessed datasets are then integrated into the system AnNetTe, which was developed for the VAST 2013 Mini challenge 3 submission. This system and the corresponding dataset are used as a benchmark environment for the evaluation in this thesis. In the evaluation of the resulting anomalies the methods have very similar results. The types of events detected are different for the various variables used in the STL method. Depending on this variable choice, the STL method or the entropy method have better results in the amount of true attacks. The subnet separation has similar results to the general STL anomalies. The most decisive measurement was a small user study which compared both approaches. In this study, the majority of users decided that the entropy method is easier to comprehend, learn and use. Part of this result is, that the entropy method gives a direct hint at the type of event and the meaning of the peaks. For example a DOS attack is shown by a small amount of entropy for either IP and the destination port at the same time as a relatively high amount of source ports is used. Such patterns can instantly be recognized while exploring the anomaly detection results. The STL on the other hand has a clear reference to the variable it is used on, but could in case of the netflow data variables and the given benchmark environment not directly transfer the meaning of an event to the user. Thus the thesis concludes, that in the benchmark dataset and system which were used for the analysis the entropy method is better suited. In further studies this hypothesis might be confirmed on different datasets, but the STL could also be a better candidate in an appropriate dataset.

Bibliography

[CCMT90] R. Cleveland, W. Cleveland, J. McRae, I. Terpenning. STL: A seasonal trend decomposition procedure based on Loess. *Journal of Official Statistics*, pp. 2–73, 1990. (Cited on pages 7, 17, 18 and 21)

[CTB+12] J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. Ebert, T. Ertl. Spatiotemporal Social Media Analytics for Abnormal Event Detection and Examination using Seasonal-Trend Decomposition. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 143 – 152. IEEE, Seattle, WA, 2012. (Cited on page 21)

[HHZ10] W. He, G. Hu, Y. Zhou. *Large-scale IP network behavior anomaly detection and identification using substructure-based approach and multivariate time series mining.* Springer Science+Business Media, 2010. (Cited on page 22)

[JLWX10] B. Jiang, S. Liang, J. Wang, Z. Xiao. Modeling modis lai time series using three statistical methods. *Remote Sensing of Environment*, pp. 1432–1444, 2010. (Cited on page 21)

[LCC05] A. Lakhina, M. Crovella, C.Diot. Mining Anomalies Using Traffic Feature Distributions. *SIGCOMM 2005 Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, 2005. (Cited on pages 19, 22 and 25)

[LLLS13] D. Liu, C.-H. Lung, I. Lambadaris, N. Seddigh. NETWORK TRAFFIC ANOMALY DETECTION USING CLUSTERING TECHNIQUES AND PERFORMANCE COMPARISON. *IEEE Canadian Conference Of Electrical And Computer Engineering (CCECE)*, 26, 2013. (Cited on page 22)

[LX01] W. Lee, D. Xiang. Information-Theoretic Measures for Anomaly Detection. *In Proceedings of the 2001 IEEE Symposium on Security and Privacy*, 2001. (Cited on pages 17 and 22)

[Mer13] F. Merkle. Visual, Interactive Steering of Predictive Techniques during Data Analysis, 2013. (Cited on pages 9, 31, 35 and 36)

[NNS12] N. Naveen, D. S. Natarajan, D. R. Srinivasan. Application of Change Point Outlier Detection Methods in Real Time Intrusion Detection. *International Conference on Advanced Computer Science Applications and Technologies*, 2012. (Cited on page 22)

[SW49] C. Shannon, W. Weaver. In U. of Illinois Press, editor, *The Mathematical Theory of Communication.* 1949. (Cited on pages 17 and 22)

[TC05] J. Thomas, K. Cook. llluminating the path the R&D agenda for visual analytics. *Information-Theoretic Measures for Anomaly Detection*, 2005. (Cited on page 12)

[WHF12] N. Wang, J. Han, J. Fang. An Anomaly Detection Algorithm Based on Lossless Compression. *IEEE International Conference on Networking, Architecture, and Storage*, 7, 2012. (Cited on page 22)

A. Appendix

Study - Information and Tasks A

What is your gender? O male O female

What is your age? [ ]

What is your level of education? O Undergraduate O Graduate O PHD

How familiar are you with the tool?

O I never used the system O I once/twice tried the system O I used the system frequently

Do you agree that all your answers to the tasks and evaluation questions will be anonymously published and accessible to other people?

O Yes I agree O No I don't want my results considered in the study

TASKS - please work in order of the task number

TASK 1 Walkthrough

Note down your steps for finding the network breakdown in day 14 or 15 with using and which timeline you used.

TASK 2 STL comparison

FInd and name as many events in day 14 as possible, by using all three packages timeline (PackagesOverview; PackagesSubnetRemainder; PackagesSubnet Trend)

TASK 3 Entropy insight

Find a DOS or DDOS attack in day 2 and name the starting time of the attack by using the entropy package.

Computer System Usability Questionnaire

Based□n: Lewis, J. R. (1995) *IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use*. **International Journal□f Human-Computer Interaction**, 7:1, 57-78.

Try to respond to all the items, with points from 1(strongly disagree) to 5 (strongly agree)
For items that are not applicable, use: NA

1.Overall, I am satisfied with how easy it is to use this system.	□1□2□3□4□5□ NA
2.It was simple to use this system	□1□2□3□4□5□ NA
3.I feel comfortable using this system	□1□2□3□4□5□ NA
4.It was easy to learn to use this system	□1□2□3□4□5□ NA
5.I believe I became productive quickly using this system	□1□2□3□4□5□ NA
6.The information (tutorial screen) provided with this system is clear	□1□2□3□4□5□ NA
7.It is easy to find the information I needed	□1□2□3□4□5□ NA
8.The information provided for the system is easy to understand	□1□2□3□4□5□ NA
9.The information is effective in helping me complete the tasks	□1□2□3□4□5□ NA
10.The organization of information on the system screens is clear	□1□2□3□4□5□ NA
11.The interface of this system is pleasant	□1□2□3□4□5□ NA
12.This system has all the functions and capabilities I expect it to have	□1□2□3□4□5□ NA

List the most **negative** aspect(s):

1.
2.
3.

List the most **positive** aspect(s):

1.
2.
3.

Further questions for usability:

Which timelines were easiest to understand?

□ Entropy □Overview□ Subnet

Which types of anomalies were most useful?

□ Trend lines □ Remainder lines

With which concept was is easies to guess what the outlier means?

□ Entropy □ Overview □ Subnet

With which time line(s) did you find the most events?

-
-

Evaluation Questions

With which time line(s) did you find the events the fastest?

-
-

With which time line(s) did you find most false events?

-
-

What is your impression of the entropy anomaly detection?

What is your impression of the STL anomaly detection?

What is your impression of the STL subnet anomaly detection?

What parts of the system did you not fully understand?

What are your general comments on the tool?

What would you improve in the tool?

What did you like in the tool?

Study 10-11.30 Uhr	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6	Participant 7	Participant 8	Participant 9	Result
Data										
Gender	male	female	male	female	male	male	male	male	male	78% male 22% female
Age	22	25	23	25	22	25	24	23	28	24
Level of Education	undergraduate	undergraduate	graduate	-	undergraduate	PHD	graduate	graduate	PHD	3 Bachelor 3 Master 2
Familiarity with tool	used once	never used the system	never used the system	never used the system	used once	never used the system	used once	never used the system	used once	5 never 4 once
TASKS										
Type	B - entropy	A - STL	A - STL	B - entropy	B - entropy	B - entropy	A - STL	A - STL	B - entropy	
Task1	Select OVerview; Select Deatils; Use Play; Find Day 15 Iam	Select Overview 14-15; select detail outlier 15 0:33-1.12; Breakdown	Select day14/15; Select details; Use animation; find blue at	Select around Day15; Find entropy change at 12; no green in ring; Network	Select day 14 to end; Select day 24 23.45-day 15 00.35	Select Day 14/15; play animation; Find all blue time; refine time range;	Select day 14/15; Event mark at day 15 00am; check this is breakdown	Select high entropy; Use animation; Sleect ip myn connections; River: many	Select overview; Select details; play animation day 15; find only	89% correct
Task2	April 2nd 4.53	2 attacks submitted	Breakdown 23.39-1.39;	DOS day 2 6.25	Day 2 4.53-5.49	High sPort low Dport;	DDOS Day 14 4.30-8.00	Day 14 6.09-7.20; Day 15	Day 2 5.21	2A 75: 2B:100%
Task 3	0.40-1.30 Scan	2 DOS submitted	9am low source IP;	DOS Day 13 22.12- Day 14	-	DOS 4-8	Day 2 21.54-Day 3 1.47	22.22-22.46	Scan: Day 14 12.33-12.50	3A:75% 3B: 60
Task4	-	-	regular anomalies at midnight in entropy; day 11-18	-	-	-	DDOS Packages overview 1-2am	-	Scan 5.36-6.59; Scan 5.42-6.16; Scan 14.04-15.40; Mainly	
Task 5	-	-	-	-	-	-	Flower: many to one; River; destport	-	Entropy and event view useful for	
CSUS	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6	Participant 7	Participant 8	Participant 9	Average
Overall	3	2	2	5	2	3	5	3	5	3,333333333
Simple System	3	2	2	5	3	3	5	3	5	3,444444444
Comfortable System	2	5	2	5	2	2	4	3	5	3,333333333
Easy learnng	4	1	1	na	2	4	na	4	na	2,666666667

Quick	4	4	3	5	3	4	5	4	4	4
Clear tutorial	2	1	4	4	4	5	na	3	5	3,5
Information easy to find	4	3	3	5	4	3	5	3	4	3,777777778
Information easy to	4	1	3	5	4	4	4	3	5	3,666666667
Information effective for	4	1	3	na	2	3	5	4	4	3,25
Organization of system	4	1	4	5	3	3	4	3	5	3,555555556
Pleasant	3	2	2	5	3	2	na	3	na	2,857142857
Provides all functions and capabilities	3	1	4	5	4	3	5	4	4	3,666666667
Free text										
Negative Aspects	Slow; Bugs; Unclear introduction	Slow; Overlap; Unclear colors	Slow; Overlap	Label ambiguous; Slow; Details cluttered	Slow; complicated	slow, bugs; dirty interface	Slow; color meaning; Crowded timeline;	Unstable; not understandin g; Timelineswitc	labels overlap in different browsers; slow;	
Positive aspects	Easy events	Easy use; Useful for events;	Good presentation; nice graphs	Flower hoverting; Entropy	beatiful; commentimg interaction	Clear workflow; colaboration;	Event outstanding; Others	Packages; Animation; Collaboration	Interface; color scheme; Linked views	
Which undestandabl e timeline	Overview	Entropy	Entropy	Overview	entropy	entropy	Entropy	Entropy	overview	
Which useful anomaly	Remainder	Trend	Trend	Trend	trend	remainders	trend	-	trend	
Which meaningful	Entropy	Subnet	Entropy	entropy	entropy	entropy	overview	Entropy	overview	
Which most events	Entropy	entropy only	Entropy	entropy; events	entropy	entropy	BytesOvervie w; Entropy	entropy	event; entropy	
Which fastest events	Entropy	entropy	Overall; Health; IPS	entropy; events	entropy	entropy	Bytes; Payload	entropy	event; entropy	
Which false alarms	no false	no false	no false	no false	overview	packages overview	entropy	no false	IPS Warning; Total Bytes	

Impression Entropy	Good; useful	useful;	Expert necessary;	powerful; easy understand;	easy; slow	very effective	-	very effective	Not representing the truth of	
Impression STL	not understood	duration not useful;	-	Complex; Hard to understand;	not understood	many features; difficult to	-	Overview good; not used other	Lables overlapped	
Impression	not	-	-	not used	not used	not used	-	not used	Too may	
Not understandab	Small multiples	overview	Parameter options	Some timelines	IP connections	Small multiples of	Color of details	Many packages for	Time not accurate;	
General comments	Good aspects; Useful for events; interaction not fluent	Easy to use; useful for events	Better instruction for events needed; Zoom in time needed; Too	Poweful; logical; interaction by selection	Slow; submit button bug	Potential high features; bad usability; hard to use	Color of IPs helpful and important	Learning time; then easy to use; need more expertise and learning	slow	
Improvements	Interaction	Hints for feedback	-	Slow; Timeline drawing	Interface show all parts better	Performance; Bug fixing; More instructions	Backend; Insturction for timelines	Crowded interface; Divede components clearly; Color timelines not distighuish;	slow; color explanation	
Like in tool	Love and Hate	Flexible timelines	Graph and River beautiful	Exploration; Flowerview; collaboration	Events of other people shown in interface	Feature extraction (STL/entropy) Connection	Color encoding	Timeline; collaboration	Colors; beautiful curves	

Declaration

All the work contained within this thesis, except where otherwise acknowledged, was solely the effort of the author. At no stage was any collaboration entered into with any other party.

(Hanna Schaefer)