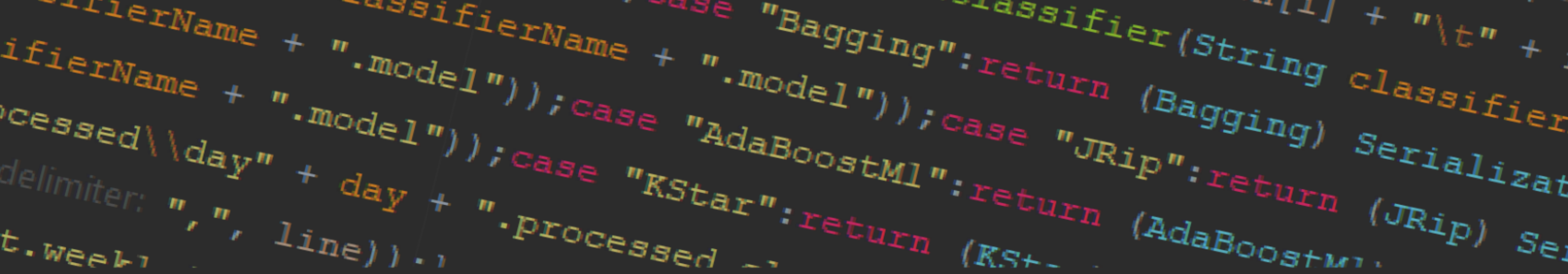


A PERSONALIZED APPROACH FOR COMMUNICATING FOUND ANOMALIES IN NETFLOW DATA TO END-USERS

DION DE HOOG

A personalized approach for communicating found anomalies in Netflow data to end-users

by R.A.D. de Hoog

To obtain the degree of Master of Science
in Computer Science & Science Communication
at the Delft University of Technology

Student Number: 4189485
Supervisor: Andre Smulders, TNO

Thesis committees

Computer Science	Science Communication
Chair: Prof.dr. A. van Deursen	Chair: Prof.dr. M. de Vries
Supervisor: Dr. A. Panichella	First Supervisor: Drs. C. Wehrmann
Member: Dr. R.R. Venkatesha Prasad	Second Supervisor: Dr. E. Kalmar
Member: Drs. C. Wehrmann	Member: Dr. A. Panichella

Preface

I hope that this research inspires researchers to broaden their view and challenge existing research. I think there is a need for people to take a step back and have a critical look at the current and classical approaches. Previous research should inspire us, but not prevent us from looking for new and unexplored roads.

This thesis has two different, although related, parts. In the first part I aim to challenge the current literature on anomaly detection and network security. In the second part I try to take a step into a personalized approach to involve end-users into protecting their network.

I want to thank my supervisors, Caroline, Éva, Annibale, and Andre, for their guidance throughout the project. Their feedback was crucial for this research. Caroline, regular checkups with you helped me stay sharp and focused. When I had a difficult time progressing, talking to you always helped me remotivate myself and rediscover the fun of my research. Éva, whenever I needed it, you could provide me with a different perspective; your feedback allowed me to look at alternatives I did not think of myself. Annibale, there were quite some decisions I had to make, sparring with you allowed me to look at the alternatives and make a grounded decision. You helped me through my research without steering me in a certain direction. Andre, I would like to thank you for the opportunity to be a part of TNO. I have had numerous fun and valuable discussions with you and my colleagues. Your experience and guidance helped me set up my thesis and choose a topic I was personally interested in.

Finally, I would also like the thank my friends and family for supporting me during this research and helping me change my mind when I needed it. Special thanks to Jorick, for creating the visuals I used in the introduction of part 1.

Summary

This thesis consists of two parts. The focus of the first part is computer science, while the focus of the second part is science communication. In the first part, we aim to find anomalies in network data, while in the second part, we want to present these anomalies to end-users.

Part 1: Computer Science

The growth in IoT is more rapid than the growth in protective measures. Attacks such as DDoS and botnets can use or target IoT devices without the owner ever knowing. There are services that monitor a network, but these are often too expensive for personal or small business use. We aim to explore the possibilities for a system that can monitor a small network.

We compare supervised and unsupervised machine learning models to determine the feasibility of such models for monitoring a home network. For this, we need to look at which features we should use, how supervised models perform compared to each other, how unsupervised models perform compared to each other, and how supervised and unsupervised models compare to each other. To answer these questions, we look at different aspects to argue whether a model is feasible; the performance, the speed and the output. We use WEKA to train nine supervised models and four unsupervised models. To train and test the models, we use the UGR'16 dataset, which contains real background data and artificial attacks.

We found that supervised methods perform better and are faster than unsupervised methods. The best performing model is random forest, with an F1-score of 0.9165. However, using the Friedman and Nemenyi test, we found that seven supervised methods are statistically similar. Supervised methods are fast enough to keep up with real-time NetFlow monitoring on a small network and show which attack occurred. Based on the results, it seems feasible to develop a system that monitors NetFlow data. Although the performance needs to be improved to reduce the number of undetected attacks and false alarms.

Part 2: Science communication

The purpose of the system in part 1 is to find anomalies and present them to a user. This part aims to find out how to present these anomalies. Since every person is different, we want to have a personalized approach. However, it would be easier if we can divide users into groups that we can approach similarly. In this part, we aim to answer the question of how to present warnings to users in a way that allows them to take action. Before we can answer this question, we first need to know how a message can influence behaviour and how users

respond to a warning message.

We start by creating a theoretical framework based on behaviour, motivation and uncertainties. The uncertainties of users can prevent them from taking the action we want them to take. We choose to focus on four different uncertainties; complexity, the intention of the attacker, consequences and competence. Based on this framework, we create different versions of warning messages that focus on a single uncertainty. We use four different attacks, combined with the four uncertainties which result in 16 warning messages.

We then interviewed 22 participants and presented them with these formulations. The participants ranked the formulations per attack based on how worrisome they sounded to them. Finally, they assigned which uncertainty was most prevalent for them. Based on these factors, we analyze whether their self-assigned uncertainty allows us to group them in our approach. Next to these rankings, we also asked the users several questions to get an insight into their stance on network security.

The results show us that we can not group users based on uncertainties. However, uncertainties do play a role in motivating users. Most users want to receive warning messages and have the intention to protect their network. We were able to extract some directions for warning messages. The most important aspect might be that users want to have a choice in which information they see.

Contents

1 Introduction

Internet is a concept that is twined into everyday life; whether it is for work, personal use, or communication. This is especially true in Europe and North America where the average internet users exceed 85% of the total population [1].

With more devices and everyday objects being linked to the internet (such as smart thermostats), being connected becomes an even more ingrained part of the lives of any person. The technology in which devices and objects are connected to the internet is called the 'Internet of Things' (IoT). The Internet of Things (IoT) can be defined as "a group of infrastructures interconnecting connected objects and allowing their management, data mining and the access to the data they generate" [2]. These connected objects are the 'Things' in IoT, also called IoT devices. The amount of IoT devices is rapidly increasing. Different sources report a varying number of devices, such as 11.7 billion IoT device connections [3] and 26.66 billion active IoT devices [4]. However, all sources agree that that IoT is here to stay and will keep growing.

The surge in IoT devices brings numeral conveniences, such as a lower need to track data manually, but they also introduce new vulnerabilities. IoT devices for home use often use cloud services; the device collects data and sends it through a local gateway to a cloud, in which data is stored or processed. The interconnected parts leave us with vulnerabilities on three fronts: cloud, gateway and device.

The new vulnerabilities introduced by IoT mean that it is vital to keep security in mind. When a cloud is used to provide services, the user has less control over the security measures. Data is stored and processed externally and the devices can often be controlled from the cloud. Since processing often happens on a cloud, users of IoT devices often do not know what is happening behind the scenes or whether they can trust the device and cloud. Without direct access to the data or communication between device and cloud, it can be unclear how one can monitor IoT devices or a cloud for suspicious behaviour. Next to that, developers can use different implementations, so every device might behave differently. There has been plenty of research on how to encrypt data or protect a cloud or device. It is, however, an ongoing challenge to secure the communication itself from and to devices.

Based on the discussed issues, we bring forth the first subject of this thesis: how can users be protected from attacks on their network? The focus of this research is on a home or small-business environment. It is often not feasible to pay for a service to monitor a small network. In this work, we will create supervised and unsupervised models using network data that contains common attacks. We will then compare the different methods to see which performs best. Finally, We will analyze whether it is feasible to use such methods for network monitoring.

If an actor or system attacks a network, we want to stop the attack and prevent future attacks. However, since each network can be different, it is not feasible to create one solution that protects every user. One possibility would be to notify the user whenever the system detected an attack so that the user can take action to interrupt the attack or prevent future attacks. However, not every user has the same knowledge and know-how about computers, networks and cyber-attacks.

To this end, we split this research into two separate parts. The first part will go into the detection of malicious data on a network using machine learning techniques. The question we want to answer in this part is *Which factors need to be taken into account to create a system to detect anomalies in NetFlow data for home usage?*. To answer our question, we need to take several steps. We start by diving into the related works to see prior research. We then choose a dataset and methods to perform anomaly detection. We will use multiple supervised and unsupervised learning methods to try and detect anomalies in NetFlow. We will then compare the supervised and unsupervised methods and argue which method performs best. Finally, we will have a look at other factors that might influence the feasibility of such a detection system.

The second part will focus on the following question: *In what way can technical vulnerabilities in the context of IoT be presented to users to motivate them to take action when taking personal uncertainties into account.* Since the focus is on small networks without external supervision, the user will have to process the warning and take action him- or herself. Presenting the danger to the users comes with several challenges; what should the warning message look like? How can users be motivated to take action? Do users have enough knowledge and know-how to take action?

To answer these questions, we create a theoretical framework based on motivation and behaviour. Based on the theoretical framework, an interview is constructed and held with multiple users. Finally, we analyze and present their answers. Based on these answers, we take the first step to see how different users react to the same warning message containing technical information.

These two parts combined are a step towards creating a system that can analyze the network of home users or small businesses and present the information to users in such a way that they are motivated to take action.

Part I
Computer Science

1 Introduction

The Internet of Things (IoT) can be defined as “a group of infrastructures interconnecting connected objects and allowing their management, data mining and the access to the data they generate” [2]. These connected objects are the ‘things’ in IoT, also called IoT devices. The amount of IoT devices is rapidly increasing; the IEEE P2413 work group expects that there will be 50 billion connected devices by 2020 [5].

As stated in the general introduction, IoT is a rapidly growing field. However, the exact definition of IoT is not agreed upon and evolves over time. Minerva et al. [6] provided definitions for IoT from seven different organizations, which all differ from the definition quoted above.
Even though the exact definitions differ, Minerva et al.[6] state there is some common ground between the definitions: IoT is the network of physical things connected to the internet, in which we can uniquely identify a thing.
Just like for IoT, there is no widely accepted definition for ‘thing’. A thing is often described as a device that gathers data, processes data and can provide a service. An example of a thing is smart lighting that can operate based on a schedule. However, a smartphone is not an IoT device since it, instead, allows a human user to connect with the internet. When we combine these different requirements, a more complete definition could be *a device that is connected to the internet and fulfils a clear purpose within the physical world without needing human interaction, after being set up*. However, this definition also presents a problem when looking at, for example, a smart TV, which has some functions similar to other IoT devices but still acts mainly on user input.

The focus of this research will be IoT in home (or small business) use. We chose this environment because there is often a lack of knowledge or capacity for network maintenance and monitoring. According to Statistics Netherlands, the amount of cybersecurity measures a company takes increases as the size of the company increases [7]. IoT devices for personal or small business use are often connected to a local network within a building that does not have its own dedicated IT support and large scale data storage to store and process information. In this case, there are three parts in the architecture: The IoT devices themselves, a network with a gateway in which the IoT devices operate, and a cloud service that handles the data or controls the device. Possible additions to a network are edge devices, which influence data or devices. An example of an edge device is a smartphone that can remotely control an IoT device. Image 1 shows a graphical representation of a network in which IoT is used

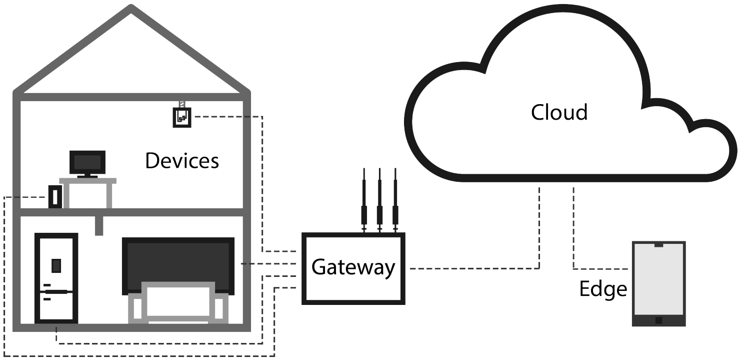


Figure 1: An overview of a network in which IoT functions.

This research aims to take the first steps towards building a tool to monitor incoming and outgoing data between devices and the cloud. There are two types of data when monitoring a network: Normal and malicious data. Malicious data consists of different types of messages that aim to attack a network. Normal data is also called background data. To detect security attacks, we need to assume that it is possible to distinguish malicious data from background data.

Since the architecture shown in Figure 1 contains different services, we need to choose which service our tool will monitor. When monitoring the cloud side, it is not visible if a device receives messages from a different source. Focusing on the IoT devices themselves would require a separate system per device. Since IoT devices are often lacking in power and computing capabilities [8], this is not feasible. We are left with the gateways, where the messages from and to the IoT devices pass through. We can use a gateway to collect all relevant data, after which we can analyze the collected data. Since the focus is on personal or small company use, there will often only be a single gateway from the local network to the outside world.

Whenever a potential attack is detected, the detection system can send a message to share this potential attack with the user. However, to detect suspicious data, it is also necessary to know how normal data looks. The idea is that a monitoring system uses old data to learn what background data is. Once the system knows which data is normal, it can use this to determine when incoming data deviates from this norm. We call these deviating records anomalies or outliers. The act of looking for anomalies in data is called anomaly detection, which we can describe as "finding patterns in data that do not conform to expected behaviour" [9]. An anomaly does not have to be malicious, but an attack, such as when a botnet is used [10] or when a network is flooded [11], should result in anomalous behaviour.

We can look at the packets sent over the internet to find out what happens between cloud and device. However, the amount and total size of these packets can become very large quickly, such as when a constant supply of packets is received when streaming a movie. Monitoring these packets and extracting useful features in real-time can thus be problematic. Another way to look at the data is using NetFlow, which aggregates packets that are part of one stream of data between sender and receiver. Each aggregated collection is called a 'flow'. Each flow contains metadata, such as the source, recipient, flow duration, size, and the amount of packets [12].

Since NetFlow data is continuously generated for all incoming and outgoing data, it includes both background and malicious data. Since newly generated data does not show whether it is malicious, it is desirable to have a separate monitoring system that can indicate when suspicious data is found. We can create such a system by using historical data from which we know which flows are background and malicious to make predictions on new data. A way in which we can do this is by using machine learning techniques. Machine learning models are trained using old data and then predict new data. There are three main techniques in machine learning. First, we have supervised learning, in which each record of the data is labelled, gives the algorithm sample pairs of data and label. The algorithm then attempts to 'recognize' data of a certain label based on what it saw before. Secondly, unsupervised learning, in which data is unlabelled, tries to group similar records. The model can create patterns or groups to see which records are close to each other and, probably, the same type of data. Finally, we have semi-supervised learning, which contains both labelled and unlabelled data. Semi-supervised learning falls in between the other two types of learning. For this research, we focus on supervised and unsupervised learning. The reason for this is that NetFlow data is typically unlabelled. If a third party adds labels, they usually add labels for all records.

However, when looking at these different types of learning, what is the difference in performance between supervised and unsupervised learning in this specific context? Which learning techniques perform best on this type of data? Are the results clear enough to inform a user? Can the model keep up with real-time data generation? All these questions result in the following main research question: *Which factors need to be taken into account to create a system to detect anomalies in NetFlow data in home usage?*. This question can not be answered immediately and requires multiple research question to conclude.

This thesis aims to lay the groundwork for future research. The research questions formulated based on the aim of this paper are as follows:

- RQ1: Which features can be used to find anomalies in NetFlow data?

- RQ2: What is the difference in performance and feasibility between supervised learning algorithms for a home monitoring system?
- RQ3: What is the difference in performance and feasibility between unsupervised learning algorithms for a home monitoring system?
- RQ4: How do supervised and unsupervised algorithms perform compared to each other?

Before we can detect anomalies in NetFlow data, we first need to know what NetFlow data looks like. Which features are in the data and which should we use? Next to that, we also need to look at related works; we do not want to reinvent the wheel. Section 2 goes into the related literature and will act as a basis for the main research question. Based on the related literature, we will create a methodology in section 3. The methodology will describe how we will approach the research of this thesis and how to interpret the results. This section will lay the groundwork for answering the research questions. We will then discuss the results in section 4. We then reflect on the results and the research in section 5. Finally, we will answer the research questions one by one, based on the results and the discussion in section 6. Figure 2 shows the structure of the research questions.

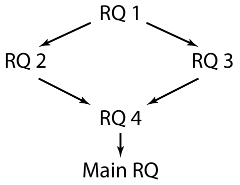


Figure 2: An overview of the relation between research questions.

We will use the UGR'16 dataset, created by Macio et al. [13]. The UGR'16 dataset contains real data with artificial, up-to-date, attacks. We chose this dataset because it is specifically created to train models for network protection. Macia et al. looked at previous datasets and their shortcomings and created the UGR'16 dataset to overcome these shortcomings.
We will use WEKA [14] to perform feature selection and to train various supervised and unsupervised models. Once we trained the models, we will test them on a test set. These tests give us a confusion matrix that shows the correct and incorrect instances. Based on these confusion matrices, we can calculate the F1-score to see how well the model performs. Since the data is highly imbalanced, we will not use the accuracy. Next to these factors, we look at the processing time and working of the models. Based on the analysis, we will argue about the feasibility of a monitoring system for home-usage.

2 Related works

The purpose of this research is to define machine learning-based approaches that we can use to monitor networks for anomalies, which we then present to a user. Techniques such as Machine learning, anomaly detection and NetFlow inspection are not new topics in themselves. In this chapter, we use related works to establish a baseline for the rest of this research.

2.1 Detection system

We are looking for a system that monitors a network and alarm us when suspicious activity is detected. There are different ways to approach this; we will give a brief overview in the following subsections.

2.1.1 Intrusion detection

Hung-Jen Liao et al. describe intrusion detection as “the process of monitoring the events occurring in a computer system or network and analyzing them for intrusions” [15]. The same authors also go into the difference between signature-based detection and anomaly-based detection. Signature-based detection looks for “patterns against captured events for recognizing possible intrusions” [15]. Anomaly-based detection is a “deviation from the normal or expected behaviours derived from monitoring regular activities, network connections, hosts or users over a period of time” [15]. Finally, they specify stateful protocol analysis; this is similar to anomaly-based detection except that the expected behaviour is based on predetermined protocols.

N. Idika et al. [16] focussed on anomaly-based detection, Specification-based detection and signature-based detection. In their case, the focus was malware. According to them, anomaly detection consists of a training and learning phase and can detect zero-day attacks, which are previously unknown attacks. However, previously unseen normal behaviour can also result in the system flagging it as an anomaly, as it is different from the data on which the system is trained. This can lead to false positives, where the system labels normal data as malicious.Specification-based detection is based on the correct behaviour of a system, which the model learns during the training phase. The model will monitor the system and notify us whenever the actual behaviour does not match the correct behaviour. A problem with this type of detection is that it is challenging to create a complete list of valid behaviours.

Signature-based detection does the opposite of specification-based detection. It attempts to create different models of how attacks behave, also called a signature. If the model finds a signature, the system assumes this is an attack. A drawback is that the model can only detect attacks for which it has a signature.

Signature-based detection differs from anomaly detection since anomaly-based detection looks for differences from normal behaviour, while signature-based detection looks for signatures of an attack.

In this research, we focus on supervised and unsupervised classification techniques. Unsupervised techniques are similar to anomaly-based detection as described by Idika et al. and can detect day-zero attacks. Supervised techniques will classify the anomalies into one of the classes in the training data. A drawback of supervised methods is that they require fully labelled data and can not find day-zero attacks. The following section will go deeper into this topic.

2.1.2 Anomaly detection

Anomaly detection is, as the name implies, the process of detecting anomalies in data. Anomaly detection aims to find records or observations that are inconsistent with the majority of data and, as such, raise suspicion [17]. An example could be a set of temperature measurements in a regulated environment. When each temperature measurement is between 10 and 20 degrees, except for one, which is 50 degrees, this is an anomaly. Such an anomaly can mean the measurement itself was faulty or that the system is not working like intended. Most datasets are not as straightforward as temperature measurements. Each record in a dataset can contain a wide variety of features, and an anomaly could be a lot more subtle when deviations are divided over multiple features. The complexity of anomaly detection is why it is often not done manually but by automated systems. Machine learning is one of the systems that we can use when “The application is too complex for people to manually design the algorithm” [18], which is the case when there is a wide variety of features.

2.2 Machine learning

Machine learning encompasses a wide variety of techniques, such as classification, which in turn contains multiple ways to perform, such as decision trees or neural networks [19]. These techniques use known data to train a model, which makes predictions on new, unseen, data. A machine learning algorithm is usually trained and tested with different sets of data, called the training data and test data [20]. A machine learning algorithm uses the training data to create a model that predicts data points of the test data.

For this research, the main focus is using supervised and unsupervised learning techniques to detect attacks related to IoT devices. We choose these techniques because they have been studied extensively and thus have a solid basis. While both methods rely on calculating the distance between an instance and a previously determined baseline there are also some differences, such as supervised methods outputting the type of attack while unsupervised only stating that data is malicious.

One question that remains is whether we can find anomalies in the traffic from and to IoT devices with machine learning. A well-known attack in the field of IoT is a Mirai Botnet attack; this botnet looks for IoT devices using a set of default credentials. C. Kolias et al. state that every device that is found is added to its network and can be used to attack other services through a DDoS attack [21]. The same authors also state that botnets such as these do not focus on avoiding detection. Since a botnet does not focus on avoiding detection, we expect that the attack is sufficiently different from background data. Since the attack should differ from background data, we should be able to use machine learning to distinguish botnet attacks from regular background data.

Supervised and unsupervised learning both have a wide variety of algorithms, such as Naive Bayes or Simple K Means, respectively. These algorithms can have different performances based on the type of data. Gogoi et al. describe that some techniques assign scores to new data to indicate how likely it is that this record belongs to a certain category. Other techniques directly assign labels, which state in which category the instance falls [22]. Gogoi et al. also compared supervised and unsupervised techniques using the KDD-CUP-99 dataset. They found that supervised techniques had a slightly lower attack detection rate, but also a lower false positive rate. However, this only works for datasets with known attacks.

Even within clustering or classification methods, there are different ways to approach anomaly detection. Terzi et al.[23], for example, took partitions of NetFlow data instead of individual records and calculated several features such as the number of unique source ports. Based on these features, the partition was clustered and labelled as anomalous or normal data.

We can also use machine learning to predict the type of background traffic in NetFlow. Carela-Espanol et al.[24] use the C4.5 supervised algorithm to distinguish P2P, HTTP, Network, DNS, FTP, Email, VoIP, Chat, Games, Streaming and other traffic instead of focusing on attacks. They achieve an accuracy of 90%.

The following subsection provides a brief overview of related literature for both supervised and unsupervised approaches.

2.2.1 Supervised learning for network security

Supervised learning trains a system on a set of data that contains labels, after which it classifies new data into one of these labels. An example of supervised learning techniques are classifiers [19]. When used for finding attacks in network data, this method has the advantage that it will return a label, which indicates what type of attack most likely occurred. A disadvantage is that the system needs labels, which are not present in newly generated NetFlow data, next to that it will also not be able to classify new attacks without retraining the model with this new attack. Labelling a dataset can also be a costly process as the dataset grows larger, which requires either a very systematic approach or a high amount of manual labour.

L. Bilge et al. [10] take an approach similar to signature-based detection. They use algorithms that extract patterns from NetFlow and uses these to find botnet command and control servers. They use flow sizes, client access patterns and temporal behaviour to create these signatures. They change the threshold of the classifier, which increases the detection rate and false positive rate. They found that they can detect roughly a third of the command and control centres while having a 0% false positive rate. Their highest detection rate was 90.4%, but this resulted in a false positive rate of 6.0%. While their results seem promising, they take a very narrow approach. They use a single method to detect botnets. The method they used is a method they developed themselves; we do not know if they use an existing supervised method.

In another paper by H. Yao et al. [25], they split up the detection into two parts. First, they check for background vs anomalous data, then they check which type of anomaly occurred. They tested different combinations of k-means, random forest and decision tree models to find DoS, Probe, U2R and R2L attacks. Their detection rate was between 98.6% on DoS and 56.1% on User to Root (U2R) attacks. A shortcoming of this research is that they only compared three different methods. Next to that, the results focus on the accuracy of the classifier. Especially for imbalanced data, accuracy is not always a good metric to judge a model. The accuracy is calculated by adding the total number of correctly predicted instances and dividing that by the total number of instances. The problem with this is that imbalanced data can give a skewed impression. Accuracy does not take into account the cost of misclassification for a certain instance. One could argue that a background instance being labelled malicious is less problematic than a malicious instance being labelled as background. This is why different metrics are often used instead of, or in addition to, accuracy. A big problem in their research is that they use different combinations of models and take the best result. The best performance for DoS was a combination of decision tree and random forest; for U2R, this was a combination of k-means and random forest. The problem with this approach is that this is only feasible when the data is labelled. When data is unlabelled, we can not see which combination performs better. The authors also do not test whether a specific combination works better on different datasets.

N. Williams et al.[26] tested five different classification algorithms on NetFlow data: Naive Bayes (using both discretisation and kernel density estimation), naive Bayes tree, C4.5 and Bayesian network. They found that C4.5 has the best performance and is robust when working with small sets of features. Their research was only based on detecting DDoS attacks. They also used two different feature selection algorithms (correlation based and consistency based), which reduced the dataset to 7 and 9 features. The smaller set of features did impact the accuracy slightly but still produced good results. The authors use accuracy to determine the performance of the classifier. As we stated before, the accuracy alone is often not enough to determine whether a classifier produces good results. The author also only uses supervised methods. The authors sample from three datasets to obtain the data they will use but do not explain why they use these datasets.

M.M. Rathore et al. [27] use 9 features (duration, protocol, service, number of roots, number of packets, packets/second, mean packet size, packet size std and packet size range) with six different classifiers (Naive Bayes, conjunctive rule, SVM, random forest, J48 and Reptree) for intrusion detection. They find that J48 and Reptree perform the best when looking at accuracy and efficiency. They use a "four-layered IDS Hadoop-based architecture". The authors do not explain why they used these classifiers; they also only look at the accuracy of the models.

D. Rossi et al. [28] also look at classifying network traffic using NetFlow. They used a supervised learning technique, a Support Vector Machine (SVM), to classify P2P TV, P2P file-sharing, P2P VoIP, Naming and Other UDP traffic. They only used the count of packets and bytes exchanged with other hosts. When looking at the byte-wise accuracy they obtain results above 90%. When they look at signatures, however, the results are less positive and range from 12.5% to 87.6%. Their research is limited and only tries to distinguish P2P data; we do not know how well their method would work for anomaly detection.

Hou et al. [29] created a random forest model to detect DDoS attacks in NetFlow data. They claim that *DDoS attacks have become one of the most popular and dangerous cyber-attacks.* In the case of IoT, it works slightly different. A single IoT device will most likely not be the target of a DDoS attack, but it can be used in a botnet to perform DDoS attacks. The authors used Flow-based features and Pattern-based features to perform the classification. They achieved a 99.5% accuracy and 0.4% false positive rate on their data set and a 97.4% accuracy and 1.7% false positive rate on a public dataset.

2.2.2 Unsupervised learning for network security

Unsupervised learning techniques, such as clustering, train a system on a set of data without labels after which it determines to which cluster the new record likely belongs [30]. The advantage of this method is that it can handle any type of data that differentiates enough from background data. The drawback is that it must be sure the training data does not contain attacks. Next to that, this type of system decided whether the occurrence is background data or an anomaly; it is thus not clear which type of anomaly occurred.

J. Zhang et al. [31] used two unsupervised techniques, iForest and Local Outlier Factor (LOF). They claim that LOF is a state-of-the-art technique.

Table 1: Overview of the related literature, with the amount of supervised methods, unsupervised methods, the dataset and performance metrics.

Author	# Supervised	# Unsupervised	dataset	Metrics
L. Bilge et al. [10]	1	0	Original	accuracy & True positive rate
H. Yao et al. [25]	1	2	KDDCUP99	accuracy
N. Williams et al.[26]	5	0	Original	accuracy
M.M. Rathore et al. [27]	6	0	KDDCUP99 & NSL-KDD	accuracy
D. Rossi et al. [28]	1	0	Original	accuracy
J. Zhang et al. [31]	0	2	Original	none
D.S. Terzi et al. [23]	0	1	CTU-13	accuracy
Hou et al. [29]	1	0	Original & CIC-IDS-2017	accuracy & false positives

In their comparison between the two, it seems that iForest has a higher performance. However, since they generated their data set without labels, they cannot validate the actual results. They also only used four numerical values for the instances.

D.S. Terzi et al. [23] focused their research on big data since they claim that the large amounts of internet data nowadays requires a big data approach. They divided NetFlows into one-minute intervals and aggregate them using the source IP. They then use k-means clustering and the euclidean distance of the elements to the centre of the cluster. Based on the distance, they decide whether the flows are malicious or not. They achieved a 96% accuracy. The accuracy does not directly tell the whole story. When we look at the research of Terzi et al, we can see that it yielded a high amount of true negatives, which are included in the accuracy, but not in the precision or recall. If we use their confusion matrix to calculate the precision and recall we get 0.833 and 0.724 respectively, which is far less impressive.

2.2.3 Overview

In this section we will give a short overview of the literature. Table 1 Shows a summary of the literature we used. Our literature search is far from exhaustive, but it does paint a clear picture. Every research has its unique approach, which makes it difficult to compare. Most researches use either supervised or unsupervised methods. The dataset is also not a consistent factor, especially an original dataset without much explanation can make it difficult to compare the results. Another problem is that most results focus on accuracy, which often gives a skewed impression. Multiple papers also state that it is challenging to find a dataset that suits the purpose and is recent enough. Finally, there is not a consensus on the best set of features. Most researchers use their own set of features, which also makes it difficult to compare the results.

We aim to give a comparison of different supervised and unsupervised learning techniques. We use a recent dataset that contains real background data. In this thesis, we use basic learning methods to establish a baseline of the performance of different techniques. From this baseline, further research can then refine the results using more sophisticated methods.

2.3 Datasets

Any machine learning technique requires a dataset. To apply supervised learning the dataset also needs labels for the training phase. Other important aspects of a dataset are features and size. A feature is a dimension to describe an instance in a dataset, such as a length or duration. A full set of features then gives a description of an instance in the dataset. More features mean that each instance is described in more depth, but this does not always lead to better classification. In this subsection, different aspects that are relevant when looking at a dataset are described.

Several vital requirements are already in place for a dataset for this research: It needs to be a NetFlow dataset, it needs to contain both background data and malicious data, and finally, it needs labels. Especially labels are a challenging factor. For an unlabeled dataset, we can simply capture network traffic; a labelled dataset needs a setup where every record has a known label.

2.3.1 Imbalanced data

One of the challenges when it comes to machine learning techniques is an imbalanced dataset. According to R.G. Shaw et al. [32], there are three ways in which a dataset can be imbalanced. The first way is data in which the number of occurrences differs significantly per class. The second way is when records are missing certain parts of the data, which leads to incomplete instances. The third way is when the model does not test every outcome against each instance; this is only relevant when one instance can have multiple labels.

In the case of NetFlow, we can assume that data is not missing, since all data is generated in real time. Each instance can also only have a single label. Each instance is either background data or a specific attack. We are left with the option of the occurrences per class being unequal. If this is the case for a dataset, it has several consequences. First of all, the accuracy of a machine learning algorithm, which is the percentage of correctly classified instances, can become misleading. If a dataset consists of 99% background data, then the accuracy would still be 99% if we predict that every instance is background, even though every attack is mislabeled [33]. A possible solution for this is to look at statistics per class and analyze the cost of misclassification for a class.

The cost of misclassification is the consequence of making a wrong prediction. An example of this can be diagnosing a healthy person with a disease or diagnosing a sick person as healthy. If this is a lethal disease, then the cost of classifying a sick person as healthy is very high. In the case of a common cold, this cost is a lot lower. We know what the consequences are of imbalanced data and how to deal with these consequences. Training on imbalanced data can make the model perform worse; we sample the data to make sure each class is represented sufficiently. Since accuracy can be misleading when the data is imbalanced, we will use different metrics that give a better overview of the results.

2.3.2 Features

A dataset contains a certain amount of features; based on these features, a model can be trained and used to make predictions. However, we can extract additional features. In this subsection, we discuss different approaches and datasets, together with their features. More features can improve the performance of the model. However, too many features can reduce the performance of the model. We want the smallest set of features that gives the most amount of information.

J. Dromard et al. [30] created a clustering method that uses a sliding window to extract 17 features. They then use the clustering method on the features. Their method can run in real-time and find anomalies quickly so that the network can be protected. However, they still have the problem of finding changes over time in network data; if the data already contains anomalies during the training, the model might see those anomalies as background data.

In the paper of J. Zhang et al. [31], they compare two different unsupervised techniques, Local outlier factor and Isolation forest, to detect network intrusions. They find that Isolation forest performs better based on their set of features, which included: Source address, destination address, source port, destination port, bytes, packets, duration and transfer rate.

2.3.3 Existing features

Datasets already contain certain features. However, the included features can differ per dataset. In this subsection, we take a look at the literature and at which features are used.

M. Ring et al. [34] performed a literature survey and reported that the following attributes are typical of flow-based data: Start time, duration, transport protocol, source IP, source port, destination IP, destination port, number of bytes, number of packets, TCP flags.

S. Garcia et al. [35] provided a dataset for botnet attack detection that has been used as a benchmark for anomaly detection techniques. The authors show a method that aggregates the NetFlows in windows and uses the following features: Source IP, unique source ports, unique destinations, Amount of flows, Amount of bytes, Amount of packets. The full dataset contains the following features: start time, duration, protocol, source IP, source port, destination IP, destination port, flags, type of service, packets, bytes, flows and a label, which is similar to the findings of Ring et al.
[34].

A more recently created dataset for anomaly detection in NetFlow data is the UGR'16 dataset [13]. This dataset contains the following features: Start time, duration, source IP, destination IP, source port, destination port, protocol, flags, forwarding status, type of service, number of packets, number of bytes and a label. These features are yet again in line with the research of Ring et al. [34]

2.3.4 Feature extraction and selection

Sometimes we can improve the performance of the model by adding or removing features. If a feature only adds noise to the rest of the model, it is advisable to remove them. Sometimes, a feature can also skew a dataset because of how the dataset is build up.

V. Carela et al. [24] used the following features for supervised learning: source port, destination port, protocol, type of service, flags, duration, packets, bytes, average packet size, average packet interval. They use the existing features to calculate the average packet size and average packet interval. The authors do, however, not explain why they added these specific features. On the other side, they exclude IP information to make the system more robust on other networks.

In this research, we will add some features, after which we perform feature selection to see which are relevant; subsection 3 describes this process. Feature selection looks at the impact of specific features and then results in a list of features that have a high enough impact to be included in the model.

2.4 Countermeasures

As we stated before, a detected anomaly does not necessarily have to be malicious. A flow from a different source could mean the device provider made a change in cloud infrastructure; a larger flow at an irregular time could be an update. In this part of the thesis, we will not be looking at countermeasures. The reason is that the intention is to create a system that we can deploy at home networks where end-users are notified when we find an anomaly. In part II of this thesis, we will go into the warning messages we send to users. We will hold interviews to ask if participants want a system like this and how we should present the warning message to them. However, we will still give a short overview of a possible automated solution as well.

One way to directly protect a network is by using Software Defined Networking (SDN). The idea behind SDN is that we can decouple the control plane and data plane. The central gateway takes care of the data while the SDN system has the control [8]. Dividing the data and control plane means that we can dynamically change the rules, regardless of the type of gateway. Bull et al. [8]

created a system that analyzes the data throughput rate and interval of transmission of IoT devices on a network. They then use this information to make flow-based rules in the SDN system. When flows occur at odd intervals or the flow is larger than normal, the flow rules can dynamically be changed to keep the network performance from decreasing. This system works for flows originating from IoT devices as well as flows send to IoT devices. The authors then show that a flow that threatens to flood the network can dynamically be blocked.

Bera et al. [36] performed a survey on SDN for IoT. They reinforce that SDN can be used to manage resource utilization based on Flow-rule-based traffic forwarding. They also add that SDN can balance network load and minimize network delay when the network consists of a large amount of IoT devices. According to Bera et al. SDN can be deployed on an edge device for the network.

3 Methodology

This chapter describes the methodology. The main question of this research is *Which factors need to be taken into account to create a system to detect anomalies in NetFlow data in home usage.* In other words, the aim is to provide building blocks to implement a concrete system. Since the intention is to monitor the data on a day to day basis, our system should process the data in real time. The result should be a system that uses existing data to monitor present data. If all of this is combined, we can deduce several requirements.

The following research questions steer out research:

RQ1: Which features can be used to find anomalies in Netflow data?
We will base the answer to this question on the dataset, literature and feature selection. The data present in the dataset will determine which features are already present and which we can extract. The related literature will serve as guidance for which features are classically used. Finally, we will perform feature selection on the total set of features to see which have a significant impact on the classification or clustering.

RQ2: What is the difference in performance and feasibility between supervised learning algorithms for a home monitoring system?
We will use different classification methods after which we will compare them. Metrics such as recall and precision will serve as a way to compare the results. The time it took to complete and the flexibility of the algorithm determines the feasibility for a home monitoring system.

RQ3: What is the difference in performance and feasibility between unsupervised learning algorithms for a home monitoring system?
We will use the same method as we use for RQ2 to answer this research question.

RQ4: How do supervised and unsupervised algorithms perform compared to each other?
We will analyze the answers of RQ2 and RQ3 to compare the performance between supervised and unsupervised algorithms. Based on this comparison, we discuss the feasibility of the two different methods.

The rest of this chapter will describe different parts of the methodology and how they help us answer the research questions.

3.1 Literature review

Literature will form the basis for this research. There has been plenty of research towards classifications, anomaly detection and network protection; subsection 2 shows an overview of these results. In the following sections, literature will keep playing an important role. We found literature by using keywords that are relevant to the topic of this research. The main starting point was using surveys related to IoT, network security, anomaly detection, NetFlow and network protection. From here, we employed a snowball search and citation search.

3.2 Dataset

To answer research question 1, we first need to have a dataset. Our dataset will contain certain features; based on these features, we can remove or add features. We should base the training and test set for both supervised and unsupervised algorithms on the same data. This guarantees that we can fairly evaluate the comparison between the different techniques. The dataset will need to have labels so that we can use it for supervised machine learning methods. Generating and labelling a dataset is out of scope for this project, which means public data will be necessary for the training and testing. Next to these requirements, the dataset should also contain both background and malicious traffic, preferably with attacks that are still relevant at the time of testing.

For this reason, we use the UGR1́6 dataset, created by Macia et al. [13]. This dataset contains "4 months of real background data and 6 weeks of real background data with synthetically generated traffic data that corresponds with several and well-known types of attacks" [13]. Table 2 gives an overview of the attacks in this dataset, together with a brief description. Macia et al. created the UGR'16 dataset to provide researchers with a dataset containing real background data and recent attacks. The authors show that often used datasets lack certain aspects and created the UGR'16 dataset to overcome these shortcomings.

Table 2: Types of attacks in the UGR16 dataset, with a brief description of what the attack entails.

Attack	Description
DoS	A continuous stream of packets is send from attacker to victim, in an attempt to flood the network, so that legitimate data cannot be processed.
Scan11	An attacker scans common ports of a victim to see if there are any open ports, which can be used to attack the victim.
Scan44	Similar to Scan 11, except that there are four attackers that simultaneously scan four victims.
Blacklist	Data is sent from an IP-address that is on a blacklist. This means the source is most likely not trustworthy.
Spam	A large amount of spam emails are sent from a central point to a large amount of users.
SSH Scan	A brute force attack on SSH servers to try and log in using the most common default names and password.
UDP scan	"A malware driven scanning for a specific vulnerability" [13].
Neris botnet	An attack in which an attempt is made to access devices in a network, if they can be accessed, they are added to the network of bots.

The IP addresses in the dataset are anonymized using the Crypto-PAn algorithm. In our research, we use the CSV format since the original NetFlow capture does not include labels. The inclusion of labels is essential since this allows us to use supervised algorithms. We can use the same data without labels for unsupervised models.

A drawback of the dataset is that it is not aimed at IoT devices. Since the purpose of this research is to find out how different algorithms perform on NetFlow data, we expect that the results give a good representation of how the models will perform to protect IoT devices. The dataset does contain both Botnet and DDoS attacks, which are both relevant to IoT, as we can see in the paper of Kolias et al. [21].

3.2.1 Dataset analysis

To answer research questions 2 and 3, we need to split the dataset into a training and test set. We need to make sure there is enough data to create these two sets. The amount of data in the UGR'16 dataset is no concern as it is large. The full dataset[1] contains 23 weeks of NetFlow capture, which are around 80 GB per week. The 23 weeks of capture are split up into a training set and a test set, the main difference being that the test set has a higher density of attacks. It is not feasible to process all this data due to processing constraints, so we take a single week as a starting point. We will use the first week of the test set. Table 3 shows an overview of the size and amount of the records.

Table 3: Size and amount of instances per day, together with the total size and amount of records.

Day	**Size** (in KB)	**Amount of records**
1	11.768.977	123.471.442
2	12.110.644	127.184.818
3	11.970.576	125.664.463
4	11.616.498	121.963.192
5	11.384.408	119.552.216
6	10.683.799	111.990.672
7	11.590.561	121.008.019
Total	81.125.463	850.834.822

From Table 1, we can observe that each day is comparable in size and number of records. A single day of data is still a large subset on its own. We expect that we can split up the dataset into individual days to make smaller subsets that we can process easier. Just the size and amount of records are not enough information to see if it constitutes a valid subset. Another important aspect is the number of labels there are per day, which we can see in tables 4 and 5.

[1] https://nesg.ugr.es/nesg-ugr16/august_week2.php#INI

Table 4: Amount of instances, that are labelled background, dos, nerisbotnet, scan11 or scan44, per day.

Day	Background	dos	nerisbotnet	scan11	scan44
1	120.779.166	783.640	151.525	76.284	406.077
2	125.204.583	784.186	151.640	78.282	370.198
3	136.254.446	391.527	151.964	48.139	188.554
4	120.263.352	783.842	151.490	83.310	376.128
5	117.867.837	782.356	151.368	68.278	367.007
6	104.462.147	783.901	152.419	92.234	366.955
7	97.517.366	783.680	82.168	92.491	402.284

Table 5: Amount of instances, that are labelled blacklist, anomaly-spam, anomaly-sshscan or animaly-udpscan, per day.

Day	blacklist	anomaly-spam	anomaly-sshscan	anomaly-udpscan
1	284.823	47	8	989.872
2	347.276	248.650	2	0
3	631.069	7.863	4	0
4	305.066	0	0	0
5	315.368	0	2	0
6	294.635	5.838.381	0	0
7	322.301	21.807.728	0	0

From Tables 4 and 5 we can notice that the data is not balanced across the different types of the attacks. Anomaly-udpscan only occurs in a single day, and anomaly-sshscan has so few occurrences that it is not feasible to include them. Day 7 has a large bias towards anomaly-spam attacks, which make up roughly 18.02% of the total data. We are left with day 2, 3 and 6 with the following attacks: dos, nerisbotnet, scan11, scan44, blacklist, anomaly-spam. We will use these three days as a basis for classification and clustering. However, before we can use them, they need to be pre-processed to ensure every record is relevant and usable. In the following section, we will explain how we processed the dataset.

3.2.2 Dataset pre-processing

We need data to create models for research question 2 and 3. We already chose a dataset, but we need to have a critical look at the useability of this set. Not every record is necessarily complete and formatted correctly; the data also contains labels that we do not use. These issues, among others, need to be taken care of before the data is usable. Due to processing restraints, we can not train the classifiers and clustering algorithms on the complete dataset. To this end, we will need to create representative subsets of the full dataset. Certain aspects could cause the model to be biased. We took the following steps:

1. Remove anomaly-sshscan: Since machine learning uses old data to predict new data, both the training and test set need to contains records with the same labels. There is only one day that contains anomaly-sshscan labels, and since this is also the first day, we can only train the model on this data and not test it. Because of this, we removed anomaly-sshscan records from the dataset.

2. Removed timestamp: The subsets contain records from a single day and the attacks are artificial, this can lead to a model connecting an attack to a certain day or time. If there is, for example, a set delay between attacks, it can lead to a model learning this pattern instead of being able to detect these attacks. To ensure this is not a problem, we removed the timestamp.

3. Removed source and destination IP: After analyzing the data it became clear that some IPs, almost exclusively, send different attacks. It is not the purpose to find a single IP and conclude that all or most messages from this IP are attacks. Especially when it comes to monitoring IoT devices and cloud services since the IPs of those should be part of normal background data. While IPs can be useful data, especially for attacks such as blacklist, it could create a big bias that we should avoid. Next to that, if the intention is to create a system that can work on different networks the IP loses some meaning since not every network communicates with the same devices.

4. Parse all data: Not every record was correctly formatted; some lacked a separation between values. We attempted to correct these records, but this turned out the require too much manual work and processing time since the problems were not consistent. In the end, we chose to go through each record and try to parse each value into the correct format using java code; if this resulted in an error, we discarded the instance.

5. Add features: Based on the existing data, we added two features. We will test the relevance using the built-in feature selection of WEKA. The features we added are bytes per second and bytes per package. We included these features to add data that gives a summary of what is happening in the flow.

6. Normalize all numeric data: To ensure that the data is less skewed by larger or smaller numbers all numeric numbers (including ports) are normalized between 0 and 1. We use min-max normalization for this..

7. Sample for training: To train the classifier, we selected random instances with a chance of
10000/occurrences per label. In the end, we created a dataset that contains roughly 10.000 occurrences of each label. This resulted in the inclusion of every anomaly-spam record of day 2 since there were less than 10.000 in total. For clustering, we only included background instances with a chance of 70000/occurrences of background.

8. Feature selection: After we create the training sets, we use feature selection to select which features are relevant. We use different methods in WEKA to determine whether we can remove certain features without impacting the performance. More information about this can be found in section 3.3

Once we have taken all these steps, there will be a total of 9 different datasets. For each day, there is a full set of data, a set of data containing 10.000 instances of each label, and a set of data containing 70.000 background instances.

Information about the full sets of data for day 2, 3 and 6 can be seen in Tables 6 and 7. Information about the sampled training sets can be found in Table 8. Information for the background instances is straightforward and contains roughly 70.000 instances of background data.

Table 6: Size and amount of instances per day after processing

Day	**Size** (in KB)	**Amount of records**
2	20.885.552	127.184.655
3	20.547.724	125.664.270
6	18.403.374	111.990.465

Table 7: Amount of instances that belong to a certain label per day after processing.

Day	**Background**	**dos**	**nerisbotnet**	**scan11**	**scan44**	**blacklist**	**anomaly-spam**
2	125.204.432	784.186	151.641	78.282	370.198	347.276	248.650
3	124.285.460	391.527	151.964	48.139	188.554	593.796	4.830
6	104.461.940	783.901	152.419	92.234	366.955	294.635	5.838.381

Table 8: Amount of instances that belong to a certain label per day after sampling.

Day	**Background**	**dos**	**nerisbotnet**	**scan11**	**scan44**	**blacklist**	**anomaly-spam**
2	10.090	10.140	10.075	10.011	10.033	10.122	9.951
3	10.143	10.189	10.021	9.856	10.206	10.123	4.830
6	9.871	10.031	10.011	10.184	9.985	10.020	10.140

3.2.3 Further changes

After a preliminary analysis, we ran into several issues. Since we do not want these issues to create disappointing results for RQ 2 and 3, we decided to make another two changes. First of all, the models had trouble when it came to recognizing whether an attack was a scan11 or scan44 attack. The two types of scan attacks have a similar impact on a user, and both require the user to check or close his ports as a solution. To this end, we decided to fuse the two attacks into one. Combining the two attacks resulted in a single scan attack instead of separate scan11 and scan44 attacks.

Secondly, the models had a bad performance on blacklist attacks. After careful consideration, we decided to remove blacklist attacks; these types of attacks do not need to have a signature outside of being sent from a blacklisted IP. Since these IPs are blacklisted, it seems easier to compare incoming IPs with actual blacklists rather than trying to recognize a signature.

Next to that, we found out that the normalization was also not completely fair. The models are trained on the training set of one day and then tested on a test set of another day. We then normalize the test set and training set that with different values. Since we needed to re-sample the training sets, we decided to also re-normalize the sets. Instead of min-max normalizing each set based on the minimum and maximum in the set, we normalized the test set on the values of the training set. If we use day 2 as the training set and day 3 as the test set, the sampled set of day 2 and the full set of day 3 are both min-max normalized using the values of the sampled set of day 2.

We now have 9 datasets, of which 6 new sets; for each day we have a full set of data, a set containing roughly 10.000 instances of 5 labels and a set containing 50.000 background instances.

Since the purpose is to use known data to predict future data, we will use the older days as training data and the ‘future’ days as test data. Using this approach, we have two different models; Model M2-3, which is trained on data of day 2 and tested with the data of day 3, and model M3-6 which is trained on data of day 3 and tested on data of day 6. Now that all the preparations are complete, we can start working towards asnwering our research questions.

3.3 Feature selection

To answer research question 1, we want a set of features that can represent the dataset. If features are not necessary, we prefer to remove them, since this reduces the size and dimensionality of the data. Using fewer features can speed up the training and testing of the model and reduce the complexity. Feature selection can also reduce overfitting, which improves the general performance of the model.

We will use WEKA to perform feature selection. We chose three feature selection methods that use a different approach and compare the results to determine a final set of features. We will briefly describe the methods we chose below.

Correlation Attribute Evaluation, which "Evaluates the worth of an attribute by measuring the correlation (Pearson's) between it and the class. Nominal attributes are considered on a value by value basis by treating each value as an indicator. An overall correlation for a nominal attribute is arrived at via a weighted average". This means that for each nominal attribute, the model determines the correlation against the class attribute. The model then combines these correlations into a weighted average. If two attributes are strongly correlated, it indicates that one of them might be unnecessary.

One R Attribute Evaluation, which "Evaluates the worth of an attribute by using the OneR classifier", in which the oneR classifiers creates a single rule for each predictor. For each attribute, a oneR classifier is trained. The worth of the attribute is then determined, based on the oneR classifiers. If the attribute adds sufficient information, it is included in the feature selection.

Relief F Attribute Evaluation, which "Evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class". This method is based on the nearest neighbour method, as it compares nearby instances. If two neighbouring instances have a similar value for a feature but are of a different class, the feature score decreases. If two neighbouring instances have similar values for a feature and are of the same class the feature score increases.

3.4 Algorithms

For research question 2 and 3, we need a set of supervised and unsupervised methods which we can use to detect anomalies. The idea is to use a multitude of models and compare their performance. We will use WEKA [37] to train and test the models. WEKA is a multifunctional tool that we can use for feature selection, training, and testing. Once we have the results from WEKA, we will analyze the outcome and compare them to answer research question 4. In this thesis, we use WEKA version 3.8.

Originally, we used a virtual machine (VM) with 16GB of memory but after testing, it became clear that this was not enough for all algorithms. Linear regression already ran out of memory when using 50.000 training instances. After this, we used a VM with 32GB of memory which could handle the number of training instances. In the end, we ended up using a single computer instead of a VM. We chose a computer so that we did not have to upload large files when we made a small change.

WEKA contains different categories for classification; we will choose at least one model per category based on literature. Not every model is easy to understand or interpret, especially for people that have not seen such models before. However, since this model is part of the back-end, users do not have to interact with these models, which means that the complexity of the model is not relevant for this research. We will base our choice on literature and experience. In the following subsection, we will give a brief overview of the categories and which models we chose to use. There are only several different unsupervised methods, so we will use the ones we can apply to our dataset.

3.4.1 Naive Bayes

A Bayesian network is "a probabilistic model based on a directed acyclic graph" [38]. These classifiers are based on Bayes' Theorem and assume that each feature is independent. This classifier calculates the probability that an instance belongs to a certain class based on a set of feature values. The model calculates this probability for each class; the class with the highest probability is assigned to the instance.

In Weka, the choice is between BayesNet and Naive Bayes. According to Vijayaran et al. [39] the BayesNet classifier performs better and will thus also be used in our research. There are different implementation, but these either do not work on the dataset or assume conditional dependencies [40].

3.4.2 Functions

Logistic regression and multilayer perceptron are both often used models when it comes to classification. We will use both of these classifiers. Logistic regression in itself is not a classifier but can be used for one. WEKA uses a multinomial logistic regression model. This model determines how important a feature is and assigns a weight to represent this importance. Based on these weights, a predictor function is created, which is used to classify the instances.

Multilayer perceptron is a neural network with at least three layers; an input layer, a hidden layer and an output. Each neuron in a layer is connected to every neuron in the next layer. Every connection has a certain weight. Training the network sets up the weights, which allows the test instances to go through the network based on these weights.

3.4.3 Lazy

Lazy methods do not build a model a-priori but instead groups the test data based on their properties. In the same paper used for the Naive Bayes [39], it seems that IBk has the best performance. IBk stands for instance based k, where k is the number of instances. It implements a k-nearest neighbour classifier, which is an often-used classifier. A nearest neighbour algorithm looks at which training instance is most similar to the current test instance. If the model finds the nearest instance, it assigns the same class to the test instance. for k-nearest neighbour, it does not just look at the nearest neighbour but the k nearest neighbours and determines the class based on these instances.

3.4.4 Meta

There is not a clear description for this category; it often involves using a combination of classifiers. Two types of classifiers that can be interesting are boosting and bagging. These methods resample the training data and train multiple weaker classifiers, which should increase stability. While bagging trains different classifiers which each get equal weight for the prediction, Boosting works sequentially and gives each model a weight, which also adjusts the new models [41]. Bagging can help when data is noisy, which is the case when looking at NetFlow data while boosting is prone to overfitting. Bal et al. [42] found that bagging performs better than Adaboost. Because of the nature of the dataset, Bagging will be used in this category.

3.4.5 Rules

Rule-based classifiers create a set of rules which determine the class of new instances. There is not a clear advantage for one of the rule-based classifiers so we will use both JRip and PART. JRip is an implementation of RIPPER (Repeated Incremental Pruning to Produce Error Reduction). JRip consist of two phases; a building phase and an optimization phase. In the building phase, the rules are set up by greedily adding conditions until the rule performs well. In the optimization phase, different heuristics are used to optimize the previously established rules.

PART is based on partial decision trees. In this method, the model creates multiple decision trees. For each decision tree, the model makes a rule based on the leaf with the largest coverage. After the model used a tree to create a rule, it discards that tree.

Next to these two classifiers, we will use ZeroR. ZeroR classifies every single instance as a single class. Even though this is rather useless as a classifier, it creates a baseline. If a classifier performs worse than ZeroR, it indicates that this classifier might not be worth using.

3.4.6 Trees

A tree classifier uses a decision tree to come to a result. Based on the training data, the model builds a tree; it then uses this tree to assign a class to each instance of the test set. For this category, we will use random forest. The advantage of this method is that it does not use a single tree but a multitude of trees, hence the name forest. This often yields better results, which is supported by the findings of Borges et al. [43]. In random forest classification, the model uses multiple decision trees to reach a conclusion. Each decision tree yields a classification, these classifications are then combined to reach the final classification.

3.5 Comparison

Now that we have determined which models we will use, we can determine how we will compare the different models. After all, the comparison is a critical aspect in answering research questions 2, 3 and 4. To ensure that we can compare the models, we will run each model on the same test and training set. Factors to keep in mind are that some models might be better at detecting certain types of attacks. One of the most well-known metrics to see how well a classification works is accuracy, which is the number of correct predictions divided by the total number of predictions. However, since the used dataset is highly imbalanced and more than 99% of the data has the same label, accuracy is not a trustworthy metric. To prevent the imbalanced data from giving us a misleading overview, we will use the F1-score. Before explaining what the F1-score is, we will have to go over the types of classification results.

As an example, we will use a situation in which the only labels are 'attack' and 'background'. In this case, the purpose is to find attacks. An attack will be called a positive and a background a negative. An attack that is classified as an attack is a true positive (TP), while a background that is classified as background is a true negative (TN). In both of these cases, the classifier is working correctly. However, the classifier can also have two types of errors. A background record classified as an attack is called a Type 1 error, or false positive (FP). An attack classified as background is a Type 2 error, or a false negative (FN).

Based on these statistics, multiple metrics can be calculated, such as accuracy, precision or recall. The accuracy shows the overall success rate of the classifier, which in the case of imbalanced data can give a false sense of performance. Precision is the relation between true positives and false positives; the higher the precision, the more true positives there are in relation to the false positives. Recall is similar to precision but shows the relationship between true positives and false negatives; a high recall means that there are relatively few cases of false negatives.

We can combine the precision and recall into the F1-score, which indicates the performance of a classifier. A high F1-score means there are relatively few false positives and false negatives compared to the true positives. A low F1-score indicates that either the false positives, the false negatives or both occur relatively frequently.

When looking at multi-label classification, this becomes slightly more complicated. The F1-score can be calculated for each label but should then be combined. We can do this with or without weighing each score by the number of instances with that label. In the case of our research, we chose to not use weights. Not using weights means that the F1-score for each label is equally important, no matter how frequently the label occurs. We made this choice to

prevent the imbalance of the dataset to skew the performance metrics.
Based on the F1-score, we can compare the different classifiers, which will be a vital part in answering research questions 2, 3 and 4.

3.5.1 Classification

Once we have the results, we can work towards an answer to research question 2. The comparison for classification will be rather straightforward; the results will be a confusion matrix with the classifications. From this confusion matrix, we will first calculate the precision and recall and then the F1-score. Based on the F1-score, we can rank the classifiers. Next to ranking, the classifiers based on the F1-score, we will also use the Friedman and Nemenyi tests, which show us whether there is a statistical difference between the classifiers.

Next to the performance of the model, we will also look at the runtime. If a model is significantly slower with only a slight improvement, it might not be worth the extra time.

3.5.2 Clustering

We need more preliminary work before we can answer research question 3. It is only possible to cluster results in two categories, malicious or background. This binary detection can be a problem when we want to present the anomaly to users. If a user does not know which anomaly occurred, they might not know how to solve it. The model creates the clusters using background data. Each cluster has a centre point and a radius; these spaces represent background data. Whenever a new instance falls outside one of these clusters, it is seen as an anomaly.

A challenge for clustering is that we need to choose the number of clusters per model. We use the silhouette coefficient to find a good amount of clusters. The silhouette coefficient is a cluster validity measure and is calculated as follows: s(i) = (b(i)-a(i))/Max(b(i),a(i)). In this calculation, s(i) is the silhouette coefficient for instance i; a(i) is the average distance between object i and all other objects in its cluster; b(i) is the minimum average distance between object i and all other clusters that do not contain i [44].

To calculate an average silhouette coefficient for a cluster, the silhouette coefficients of each instance are summed up and then divided by the number of total instances. The number of clusters with the lowest average silhouette coefficient should then have the best structural integrity. To make sure this process is not too costly, the silhouette coefficient will not be calculated for every cluster but rather in steps of 10 until the value starts decreasing. If value x has the highest silhouette coefficient, it will then be calculated in steps of 1 for value x-9 up till x+9. This leads to a final highest value.

When we calculated the correct amount of clusters for each of the models we can start detecting anomalies. The results will not show us in which class each instance falls, but rather whether it falls within a cluster or not. Based on this information, we can again make a confusion matrix and calculate the recall, precision and F1-score. From here, we can answer research question 3 by comparing these results similar to the classification results. We will, once again, also look at the runtime of the models.

Once we have the results and comparisons of all the models, we can compare the supervised models with the unsupervised models. With this comparison, we can answer research question 4 and work towards answering our main research question.

4 Results

In this section, we will present the results of the research. We begin by showing the features we chose to use in the final data. Next are the classification results, then the clustering results follow and finally, we will compare the different models.

4.1 Results for research question 1

This section answers RQ1, which is *Which features can be used to find anomalies in NetFlow data.* Based on the literature described in subsection 2.3, it seems that there is no one best set of features. Different researchers use different groups of features and often do not clearly explain why they choose those features. In this thesis, we use the features in the dataset as a basis. We remove the timestamp, source IP and destination IP to prevent possible problems, which we described in section 3.2.2. We added two features; bytes per second and bytes per package.

After this, we used WEKA to perform feature selection to see if all features are relevant for the performance. Section 3.3 shows an overview of the methods we used in WEKA.

Each of these methods used the default threshold, which is an extremely small negative number so that 0 is still included. The three methods all conclude that each of the features is relevant. Because of this, we did not remove any of the features. The final list of features are as follows:

- Duration
- Source port
- Destination port
- Protocol
- Flags
- Forwarding status
- Type of service
- Number of packets
- Number of bytes
- Bytes per packet
- Bytes per second

4.2 Results for research question 2

This section aims to answer RQ2 *What is the difference in performance and feasibility between supervised learning algorithms for a home monitoring system.* We will first look at the performance and then at the feasibility.

We excluded some classifiers from our research due to constraints in our setup or because of other problems. KStar was unable to complete due to running out of memory. Naive Bayes could not complete because there were too many values close to each other, which resulted in the following error: “A duplicate bin range was detected. Try increasing the bin range precision”. We attempted to increase the bin range precision but were not successful. The number of values caused there to be too many values that were too close to each other. Adaboost did not work correctly and classified everything as either scan or nerisbotnet; this resulted in a divide by zero when calculating precision or recall.

As explained in section 3.5 the F1-score will be used to analyze the performance of the classifiers. After we calculate the F1-score, we will also calculate the average of the F1-scores. This average will be unweighted since the datasets are similar in size, and we already took precautions against the imbalance of the dataset. In some cases, we could not calculate the precision since both the true positives and false positives are 0. If we could not calculate the precision, we assign a value of 0, which also results in an F1-score of 0. An overview of F1-scores for each classifier can be seen in figure 3.

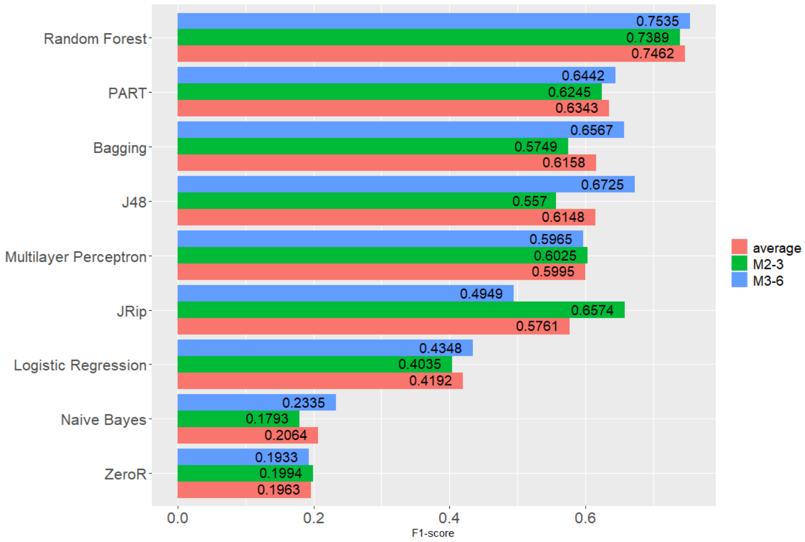


Figure 3: F1 score per classifier for model M2-3 and model M3-6, together with their average. Random Forest has the highest performance for each model.

We can immediately see that ZeroR has, as expected, a bad performance. ZeroR assigns each instance to the same label, which means that the F1-score for four out of five labels is equal to zero. Random Forest has the best performance based on the F1-score, followed by PART and Bagging. However, the fact that the highest F1-score is 0.75348 means that there are still a significant amount of errors.

Random Forest has the highest F1-score. However, an F1-score of 0.74618 indicates there are still a high amount of incorrectly classified instances.

Table 9: Precision, Recall and F1-score for random forest models M2-3 and M3-6

Label	**Precision**		**Recall**		**F1-score**	
	M2-3	M3-6	M2-3	M3-6	M2-3	M3-6
Scan	0.9302	0.9714	0.9575	0.9999	0.9437	0.9854
Neris Botnet	0.5545	0.6834	0.9935	0.9973	0.7118	0.8110
Background	0.9999	0.9470	0.9971	0.9990	0.9985	0.9723
DoS	0.9990	0.9973	1.0	1.0	0.9995	0.9987
Anomaly-Spam	0.0209	0.0009	0.9590	0.0000	0.0409	0.0000

To get a deeper insight into which type of errors there are, we will have a look at the F1-score, precision and recall per label, which table 9 shows.

In table 9, we can easily see that Anomaly-Spam is the biggest problem. With an F1-score of 0.0409 for model M2-3 and a score of 0.0000 for model M3-6, this classification drags down the overall performance. The confusion matrices, from which we calculate these statistics, can be found in appendix C. We can see that Model M2-3 had a high recall but low precision. Model M3-6 has a low score for both precision and recall.
If we take a look back at table 6, we might be able to explain this performance. Day 3 only contains a small amount of anomaly-spam records; this means that the absolute numbers for true positives and false negatives will be lower than normal, which results in a lower precision. Model M3-6 had the same lower amount of anomaly-spam records as a part of the training set. The lack of anomaly-spam could result in the model being unable to tell when something is an anomaly-spam attack. Next to the number of records being lower, it also resulted in records not being randomly chosen. If the 4830 anomaly-spam records only capture a small variety, it leads to the classifier being unable to classify other types of the same attack.

The lack of Anomaly-Spam instances in day 3 is most likely the reason for the low F-1 Score.

To combat this problem we created model M2-6; this model is thus trained on day 2 and tested on day 6. This new model removes the small sample size and might lead to better results. The F1-score for each classifier can be seen in table 4.

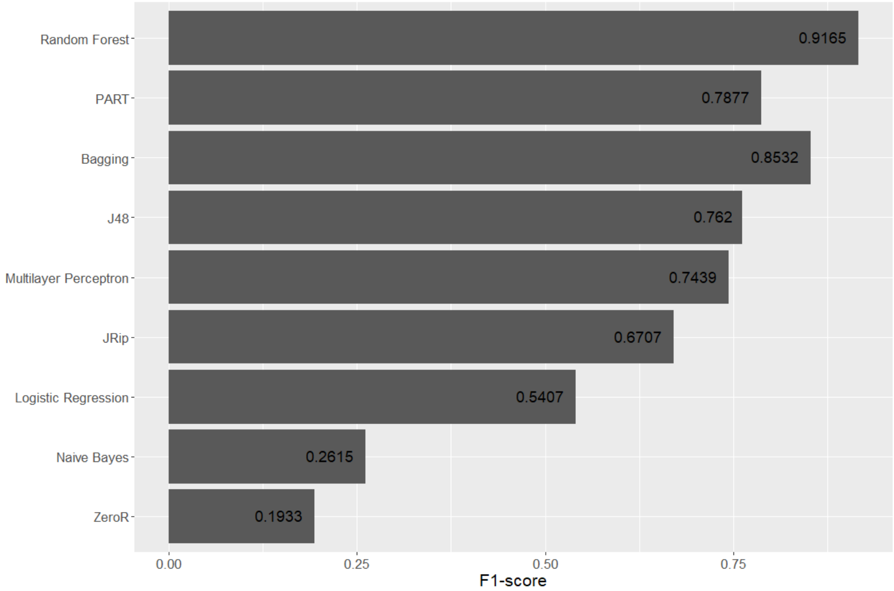


Figure 4: F1 score per classifier for model M2-6, together with their average.

When comparing figure 3 and figure 4 it is clear that the F1-score of model M2-6 is higher than the average for every classifier. The ranking of classifiers does change slightly, but random forest still has the best performance. The next step is to look at the F1-scores to see if this model yields an overall improvement or if there are trade-offs.

Table 10 shows the F1-score for each type of attack for Random Forest on model M2-6. When we compare this to table 9, we can see that the models perform similarly for scan and background; model M3-6 has a slightly higher performance for scan, while model M2-3 has a slightly higher performance for background. The performance on botnet is slightly worse compared to M2-3 and noticeably worse than M3-6. The performance on Dos is slightly better, and the performance on anomaly-spam is a lot better. Overall, it seems like model M2-6 is an improvement with minimal trade-offs.

Table 10: F1-score per label for Random Forest on Model M2-6, with the difference compared to models M2-3 and M3-6.

label	F1-score	Difference M2-3	Difference M3-6
Scan	0.9751	+0.0314	-0.0103
Neris Botnet	0.6951	-0.0167	-0.1159
Background	0.9948	-0.0038	+0.0225
DoS	0.9997	+0.0002	+0.0010
Anomaly-Spam	0.9177	+0.8768	+0.9177

Appendix D shows the F1-scores for all the models and clustering algorithms. It turns out that, with two exceptions, each model combined with Random Forest has the highest F1-score compared to the same model with different algorithms. Only Bagging has a better performance on background and anomaly-spam labels for model M3-6. All in all, model M2-6 has a higher F1-score than models M2-3 and M3-6 and will be used for further analysis.

Model M2-6 yields a higher performance for all classifiers. It seems that the lack of Anomaly-Spam instances in day 3 was indeed the problem. Random Forest still has the highest F1-score, which is 0.9165.

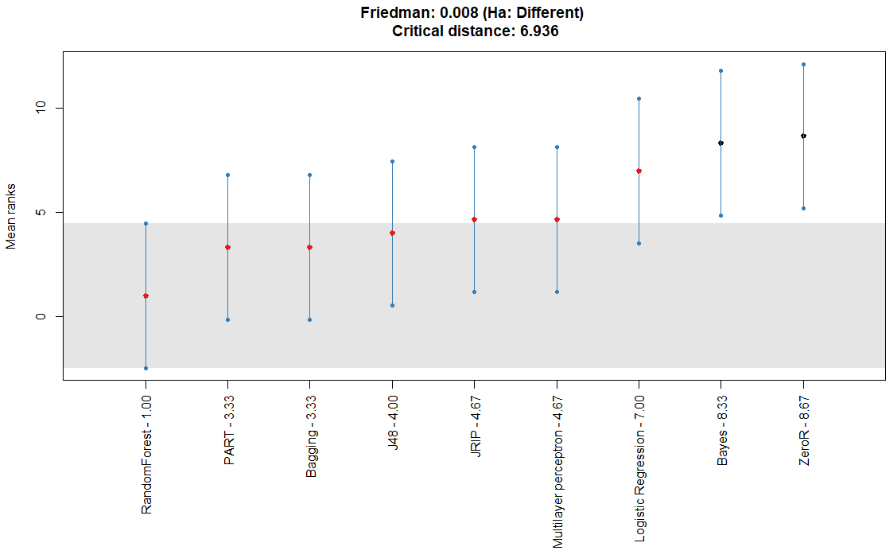


Figure 5: The Friedman and Nemenyi tests, which show that there is a statistical difference between classifiers

Next to this analysis, we use the Nemenyi and Friedman tests to perform a statistical analysis to see if the models are statistically different. Figure 5 shows the results of these tests. These tests show that 7 out of 9 classifiers are statistically similar and should be taken into consideration for further research. The only two classifiers that are statistically different are Naive Bayes and ZeroR.

There is a statistical difference between models. However, 7 out of 9 algorithms are statistically similar.

Table 11: The average amount of instances model M2-6 classified per second.

Classifier	Average instances per second
Random Forest	42359.75
PART	169650.14
Bagging	167716.43
J48	175735.27
Multilayer Perceptron	28665.89
JRip	180289.27
Logistic Regression	107400.70
Naive Bayes	55396.97
ZeroR	176336.28

Next to the performance, the time it takes to complete is also important when looking at the feasibility of a model for a monitoring system. The time for for each classification method can be seen in table 11. We can see that Random Forest is one of the slower methods. However, on average, it can still classify 42359.75 instances per second. The speed of random forest allows the model to go through all instances of day 6 in 44 minutes. Random forest should thus be feasible for real-time monitoring for home or small business environments.

All algorithms can handle real-time monitoring in a home environment. Random Forest was able to process an entire day worth of data in 44 minutes, even though it is one of the slower algorithms.

4.3 Results for research question 3

This section aims to answer RQ3: "What is the difference in performance and feasibility between unsupervised learning algorithms for a home monitoring system". We will first look at the performance and then at the feasibility.

We used four clustering methods: SimpleKMeans, EM, FarthestFirst and Canopy. First, we calculated the silhouette coefficient, using the method in section 3.5.2. The full results are a rather long list of numbers; appendix B shows a plot of the results. The final amounts of clusters are 20 for Canopy, 25 for EM, 115 for FarthestFirst and 113 for SimpleKMeans.

We could use SimpleKMeans and FarthestFirst could with functions of WEKA in java. We used the centroids and radius to see which new instances were inside or outside a cluster. EM and Canopy did not have a function to get the centroids. One idea was to calculate it manually; however, the way nominal values are handled made this complex. The distance between nominal values are treated as 1 if they are not the same and 0 if they are the same; they do not

Table 12: F1-score for SimpleKMeans and FarthestFirst, which was calculated using the center of the cluster a radius equal to the furthest point.

Label	SimpleKMeans			FarthestFirst		
	M2-3	M2-6	M3-6	M2-3	M2-6	M3-6
Scan	0.0073	0.0098	0.0114	0.0002	0.0004	0.0025
Neris Botnet	0.0011	0.0012	0.0014	0.0203	0.0216	0.02332
DoS	0	0	0.5141	0.0670	0.1212	0.3354
Anomaly-Spam	0	0.0000	0.0000	0	0.0001	0.0001
Combined	0.0043	0.0013	0.0996	0.0683	0.0336	0.0919

Table 13: F1-score for Canopy and EM, which was calculated using the manual cluster density method.

Label	Canopy			EM		
	M2-3	M2-6	M3-6	M2-3	M2-6	M3-6
Scan	0.0033	0.0080	0.0081	0.0087	0.0194	0.0207
Neris Botnet	0.0011	0.0014	0.0037	0.0012	0.0014	0.0031
DoS	0.0158	0.0402	0.0411	0.0152	0.0346	0.0349
Anomaly-Spam	0.0001	0.1265	0.1395	0.0002	0.1222	0.1159
Combined	0.0201	0.1600	0.1743	0.0249	0.1611	0.1579

have a value that we can add together.

Instead, we used a different method to determine whether a new instance falls within a cluster. The pairwise distance between every point within a cluster was added together and divided by the number of pairs; this number represents the ‘cluster density’. We then test a new instance is for every cluster in the following way: The distance between the new instance and each instance within a cluster is added together and then divided by the number of instances in the cluster. This yields the average distance between the new instance and each point in the cluster; if this distance is smaller than the cluster density it is treated as being inside the cluster, and thus background data. A disadvantage of this method is that it takes a lot longer; where the built-in methods take hours to complete, the custom method can take days to complete.

We once again take a look at the F1-scores of the different methods. Tables 12 and 13 show the results of the unsupervised methods. Next to the F1-score per class, we calculated the combined F1-score by adding up all the correctly and incorrectly classified instances.

It is clear that these results do not look very promising and are far lower than the supervised methods. Next to that, the custom function with Canopy and EM perform better, even though it is still not good, with the highest combined F1-Score being 0.1743. If we look at the results of simpleKMeans, which can be seen in appendix E, we see that most instances fall within a cluster. Both malicious data and background data is thus largely seen as background data.

Table 14: F1-score for SimpleKMeans and FarthestFirst, which was calculated using the manual cluster density method.

Label	**SimpleKMeans**			**FarthestFirst**		
	M2-3	M2-6	M3-6	M2-3	M2-6	M3-6
Scan	0.0038	0.0091	0.0210	0.0050	0.0136	0.0147
Neris Botnet	0.0018	0.0022	0.0020	0.0035	0.0041	0.0049
DoS	0.0187	0.0445	0.0361	0.0102	0.0258	0.0364
Anomaly-Spam	0.0002	0.0711	0.0940	0.0002	0.2225	0.1203
Combined	0.0242	0.1123	0.1356	0.0185	0.2464	0.1595

The performance is lower than classification. To see if the difference lies in the algorithm or whether the custom function performs better, we used the custom function for SimpleKMeans and FarthestFirst. Table 14 shows the results of using the custom function on SimpleKMeans and FarthestFirst.

This table shows that the custom method has a higher performance. The performance is, however, still not good, with the highest F1-score being 0.2464.

The performance of unsupervised learning methods is low; the highest F1-score is 0.2464 for model M2-6 and still requires custom validation to check whether clustered instances belong to the cluster or not.

Next to the F1-scores, we used the Friedman and Nemenyi tests again to see if there are statistical differences between the clustering algorithms. Figure 6 shows the results from these tests.

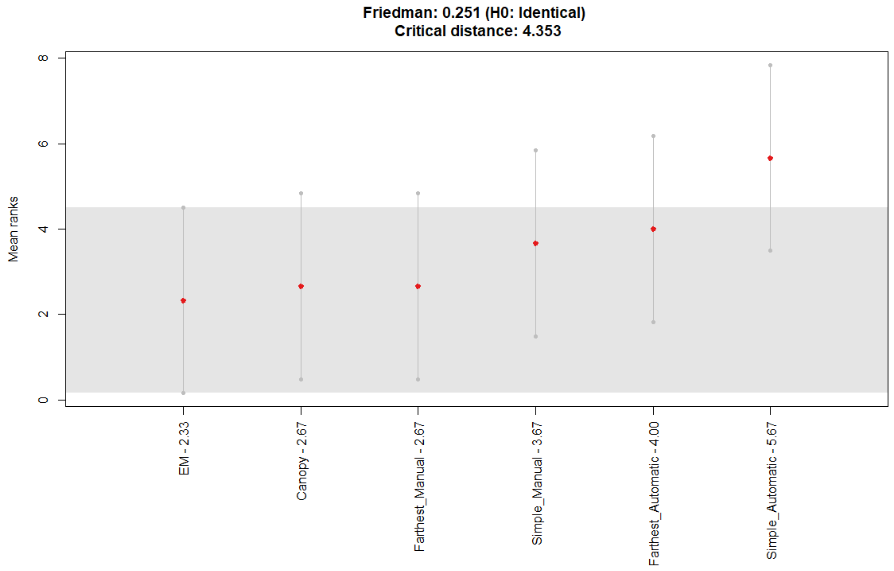


Figure 6: The Friedman and Nemenyi tests, which show that there is not a statistical difference between clusterers.

The results of the Friedman and Nemenyi tests show there is not a statistical difference between the clusterers. While further analysis would be interesting, it is not relevant to answering the research questions for this research. We decided to focus on classification instead of clustering. Further analysis can shed more light on the differences but is not required for answering the research questions in this thesis.

The performance of unsupervised algorithms is low compared to classifiers. There is not a statistical difference between clusterers. A straightforward implementation of clusterers does not seem feasible to use in real-time monitoring of networks.

4.4 Results for research question 4

This section will compare the supervised and unsupervised methods based on performance and feasibility for a home monitoring system. This comparison aims to answer research question 4.

4.4.1 Performance

Since the clustering algorithms performance was so low, it leads to a simple conclusion; this method of straight forward clustering does not perform well on NetFlow data and is not feasible for creating a system that can detect attacks.

Perhaps more sophisticated methods can increase the performance, but to make up for the difference with classification seems like a tall order. It is clear that classification is the preferred method to finding malicious data in NetFlow data.

4.4.2 Feasibility

Just the performance is not enough to determine the feasibility of classification methods for a home monitoring system. The time to complete and how the method works are also important. As we have seen, the run time does not seem problematic and using it on real-time incoming NetFlow data seems feasible.

One problem with classification is that it is unable to find new, unseen, attacks. We train the classifier on background data and certain attacks; if a new attack emerges, it will still classify it to one of the known classes. A classifier would need regular updates to keep up-to-date with current attacks. This also means that there should be a third party that works on developing a constantly updated version to guarantee that the monitoring system does not degrade over time.

5 Discussion

In this section, we will look back on the research. We will briefly recap the findings and argue what the consequences are. We will also reflect on what could be improved and possibilities for future research. In part II of this research, we write a global reflection on both parts.

5.1 Findings in this research

The main finding in this research is that supervised methods perform better than unsupervised methods. This is the case for both performance and runtime. However, when looking at the feasibility of using supervised and unsupervised methods for a home monitoring system, both of them have a key strength over the other. Supervised methods can tell the user what exactly is going on, which is a desirable function. Unsupervised methods can handle attacks that were not part of the training set, which causes it to be more future proof for new attacks. Based on the performance we found, unsupervised methods are not feasible when using a straight forward implementation, as we did in this research.

When looking at the F1-score of supervised methods, the winner is Random Forest. According to the Friedman and Nemenyi tests, however, most classifiers are statistically similar. It seems feasible to use these algorithms to create a monitoring system for home usage based on runtime and memory usage. We did not look at the complexity of the models, since we want to present end-users with little technical information.

We will need a third party to maintain the system and update it whenever possible. This third party would need an expert that can understand and work with machine learning models. If the purpose is to let users deal with the models themselves we would most likely want an easy to understand model, in which case random forest might not be suitable. In this case, a classifier, such as PART would be more suitable. PART is statistically similar to the top performers and is based on a set of rules. These rules are often easier to understand and apply.

5.2 Consequences of these findings

This research is a first step in presenting a broader overview of possibilities for network monitoring in a home environment. We can conclude that it seems feasible to create a real-time monitoring system for home-usage. A third party is necessary to monitor the monitoring system and keep it up to date when new attacks emerge.

We aim to create a system that can help users protect their network. There are services that monitor networks, but these are often expensive and not feasible in a small company or home environment. One challenge is that with the current results, there are still a high amount of false positives and false negatives. These errors are both problematic, as one leads to an attack the user does not know about and the other leads to the user worrying about an attack that is not happening.

Based on the runtime, performance and type of results, we can conclude that supervised methods perform better than unsupervised methods. Since supervised methods classify the instances, users know which exact attack they need to protect themselves from.

Compared to previous research, we used a broad approach on a recent dataset. We would also like to challenge the approach of current researches. As we showed, most classifiers are statistically similar. The similarity of our models means that 7 out of 9 classifiers are worth looking at. Perhaps we should stop searching for the highest number and start looking at other factors.

We conclude that there is a need for a third party, but would it not be better to create a system that the average user can understand? If we can create a set of rules to classify instances, we can then explain the model to the end-user. A set of rules is easy to interpret and can empower the user since it is easier to work with and understand. Empowering users does not mean there is no longer a need for a third party since rules still need to be created and updated for new or changing attacks. However, a user that understands the rules and gets a warning is more easily able to work with the system than a user that does not understand the model and receives a warning from a model such as random forest. Even though the F1-score of random forest might be higher, the system using PART might be more understandable and effective.

Of course, not every user wants to understand their network better, so we would need to research the effectiveness of this strategy. However, we hope to shift the perspective from chasing the highest accuracy to creating a system that offers the best support to its users.

5.3 Limitations and possible improvements

In this section, we will reflect on our methodology and results. We will look at possible improvements or alternatives.

5.3.1 Dataset

We chose to use a recent dataset that represents current attacks. The dataset in itself does not seem to be a limiting factor for our research, but there were certain limitations and choices we will look back on.

First of all, our processing power was limited. We used a single computer for both supervised and unsupervised learning methods. Due to the limited processing power, we only used three days of the entire dataset. This limitation caused us to be unable to use underrepresented attacks. It also made us remove certain features, such as the incoming and outgoing IP addresses and the timestamp. A possibility would have been to sample the training set from multiple days instead of a single day.

We also only use a single dataset. We would have to evaluate multiple datasets to ensure a monitoring system performs well on different networks. It would be ideal to test the model on real-world examples of home networks. Testing the model on home networks would massively increase the scope since we would need to check whether the instances are classified correctly.

5.3.2 Feature selection

We could have been more elaborate in feature selection. Due to the processing restraints and scope, we did not want to add too many features or spend too much time optimizing the features. Right now, we used three different feature selection methods included in WEKA. Our set of features should give a good representation of the data. However, since both added features also added enough information to be valuable, it might be worth it to try adding more features. Finding the ideal set of features for performance and processing power could also be a follow-up research.

5.3.3 Algorithms

Due to the scope of this thesis, we were unable to use every single method, so we had to choose a subset of methods. We based our choice of algorithms on literature. Our research does not give a full overview of the differences in performance, and it might be worth it to try more models. We also only used a straight forward implementation of both supervised and unsupervised methods. We made this choice so that our results serve as a solid baseline. However, when it comes to optimizing the performance, each model can be fine-tuned. A drawback is that fine-tuning the model on our dataset could decrease the performance on another dataset.

Unsupervised methods had a disappointing performance; the cause is most likely that the clusters are too large. Perhaps, we chose a value based on a local maximum of the silhouette coefficient, but based on the silhouette coefficient plots, it seems unlikely we chose a local maximum. The results clearly show that clustering methods need more sophisticated methods.

5.3.4 Comparison

We made a thorough comparison of the different models; it is, however, always possible to take another step. Right now, we looked at the confusion matrices, recall, precision, F1-score, and processing time. A further improvement would be looking at which exact instances were classified correctly by which model. If one model can correctly classify the instances another model can not, these two models might be combined to improve the performance. This would require an additional phase in which the focus is not on the overall results of the classifiers but the results per instance. Since we are working with a large number of instances, this would require significant work.

5.3.5 Reproducibility of the research

The results of this research are largely reproducible. We describe the method in the thesis, the dataset itself is publicly available, and there were no external users involved. However, since we randomly sampled the training data, it is possible that re-sampling the data might lead to a difference in performance. Another aspect that might differ is the time that models need to finish. We used a personal computer to execute the code, and better or worse hardware can influence the runtime.

We used our code to perform the training and testing of models, however, since we used WEKA as the basis, the results should be reproducible. The main addition of our code was to cut the data up into smaller pieces so that memory would not be a problem.

5.4 Future research

The original idea of this research was to focus on IoT. At this moment, it does not seem as if there is a dataset that focuses on only IoT devices. As discussed before, it is not even completely clear what an IoT device exactly is. This might make focusing on a network entirely the preferred option. One could argue that one of the most dangerous attacks on IoT devices are botnets since they give the attackers access to the device. An IoT device that is part of a botnet can also be used in a DDoS attack. If we can thus detect botnet and DoS attacks on a network, this might already be the first step towards protecting IoT devices without focusing on them.

Not every attack is as valuable to include in a monitoring system. When a person is getting a DoS attack, it is obvious that the network is slow or not working. On the other side, an outgoing DoS signature could imply that an IoT device is part of a botnet and is used for a DDoS attack. If we can detect outgoing DoS attacks, we might protect IoT devices against botnet attacks.

Similar to DoS, it might be better to teach users how to recognize and deal with spam. If a user gets a message that they received spam, it could lead to the user misjudging which message was spam. A false positive could also let the user see a legitimate message as spam. Yet again, outgoing spam messages could imply that someone else is using your devices to send spam. Additional research could find out whether we can use NetFlow for this purpose.

Another option for further research would be to investigate which exact instances are labelled correctly by which algorithm. Right now, Random Forest has the highest F1-score. It could be possible that some of the errors are made on a specific signature that another algorithm can detect. In this case, a combination of different algorithms working together might perform better. For this, the classified labels for the test set would have to be stored, analyzed and compared between models.

In an ideal world, the system could be deployed at a user's network and is self-sufficient. In real life, a model will need to be retrained based on, for example, the emergence of new attacks or changes in infrastructure.

In this research, we used different algorithms for both supervised and unsupervised learning methods. A possibility for further research could be combining different algorithms and methods.

It is vital to find an effective way to communicate technical warnings to end-users. We want users to be motivated to protect themselves from possible dangers. To achieve this, we should look at how we can present information and how it is processed. Next to that, motivation and behaviour are important factors when it comes to effectively reaching end-users. These are key points in part II of this research.

6 Conclusion

In this section the conclusion of this research will be presented based on the results. Each research question will be answered in the same order as they were presented.

RQ1: Which features can be used to find anomalies in NetFlow data
From the literature it becomes clear that there is not one clear cut group of best features. Using feature selection also shows that each feature used in this research is relevant. With the obtained results it seems that this set of features offer enough information to perform machine learning. Including the source and destination IP could be a good follow up, but this would need a dataset in which certain attacks are not all sent from the same IP.

RQ2: What is the difference in performance and feasibility between supervised learning algorithms for a home monitoring system?
Based on the F1-score we can see that each classifier performs better than the baseline set by the ZeroR classifier. We can also see that there is a difference in performance between the different classifiers. However, when we look at the Friedman and Nemenyi tests, we can see that 7 out of the 9 classifiers are statistically similar. If these classifiers are statistically similar, it seems like Random Forest has the best performance based on the F1-score. Each classifier is able to keep up with real-time data monitoring based on the data in the dataset.

One problem with these supervised algorithms is that they will classify everything into one of the learned labels. This means that if a new attack emerges it will still be classified into one of the existing labels. A monitoring system based on a supervised learning algorithm will thus require regular updates from a a third party to stay up to date.

RQ3: What is the difference in performance and feasibility between unsupervised learning algorithms for a home monitoring system?
Based on the results it seems like using the cluster density yields better results than using centroids with a range equal to the farthest instance. This might be caused by the clusters being very large, which causes too many instances to be within a cluster.

Another problem was that WEKA could not give the centroids for EM and Canopy, this means a custom method was necessary to calculate whether an instance falls within or outside a cluster. This method is based on the cluster density. The average pairwise distance between each combination of instances of each cluster is calculated first. When a instance is assigned a certain cluster, the average pairwise instance between this instance and each instance in the cluster is calculated. If this average is lower or equal than the cluster density, then it falls within the cluster, otherwise it falls outside of the cluster. This means that the training instances still need to be used when performing the actual clustering, which requires more memory.

The performance between different unsupervised models are also defined; there is not one model that performs best for all algorithms, however, the custom method yields higher F1-scores than the automatic method. The best performing combination is Model M2-6 with Farthest First clustering using the custom method, which still only yields an F1-score of 0.2464. The Friedman and Nemenyi tests show that the six different methods are statistically similar as well.

The runtime of the cluster density method is also long. To compute the density, the distance between each pair of instances within a cluster is needed. Then, for each instance that needs to be clustered, the average pairwise distance between this instance and each instance in the assigned cluster needs to be calculated. This causes the computations to be heavier and slower.
Next to these points, the custom method requires keeping a lot more data in memory, which means it also needs workarounds or more memory.

One advantage that unsupervised learning has is that, theoretically, it should be able to deal with new attacks. However, because of the low performance and longer computation times, it does not seem feasible to use a straight forward unsupervised method.

RQ4: How do supervised and unsupervised algorithms perform compared to each other
The best performing supervised method has a F1-score of 0.91648 while the best performing unsupervised method has a F1-score of 0.24638. Based on this performance, classification is heavily preferred over clustering when using NetFlow data. More sophisticated techniques could possible elevate the performance of classification slightly and the performance of clustering significantly.

When it comes to runtime, classification is also preferred. Training the models take seconds to minutes and even though it is not relevant to the end-users it is still a positive aspect. The actual classification is also rather fast. For the custom method of clustering, each test instance requires calculating the pairwise distance between the test instance and each training instance in the assigned cluster. Since the custom method requires using the training instances to calculate the pairwise distance when testing, clustering also requires more memory, which classification does not suffer from. However, since all the clustering methods are statistically similar we could also use the non-custom method, which does not suffer from the increased memory usage.

The only advantage that clustering has it that it should be able to deal with new attacks. However, with all the disadvantages that come with clustering, it does not seem worthwhile to use it. Based on the results in this study, the best solution would be to use supervised learning, specifically Random Forest.

Outside of the results, we argue that the focus of research like this should shift. We should not try to chase the highest number, but rather look at different factors to create a system that is usable for the intended end-users. If a model has a slightly lower performance but is easy to understand, it might result in more willingness to work with the model.

Computer Science References

[1] *INTERNET USAGE STATISTICS.* `https://www.internetworldstats.com/stats.htm`. Accessed: 21-10-2019.

[2] Bruno Dorsemaine et al. "Internet of Things: a definition & taxonomy". In: *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies.* IEEE. 2015, pp. 72–77.

[3] Knud Lasse Lueth. *IoT 2020 in Review: The 10 Most Relevant IoT Developments of the Year.* URL: `https://iot-analytics.com/iot-2020-in-review/`. (accessed: 20.01.2021).

[4] Ana Bera. *80 Insightful Internet of Things Statistics (Infographic).* URL: `https://safeatlast.co/blog/iot-statistics/`. (accessed: 20.01.2021).

[5] Oleg Logvinov et al. *Standard for an architectural framework for the internet of things (iot) ieee p2413.* 2016.

[6] Roberto Minerva, Abyi Biru, and Domenico Rotondi. "Towards a definition of the Internet of Things (IoT)". In: *IEEE Internet Initiative* 1 (2015), pp. 1–86.

[7] Centraal Bureau Statistiek. "Cybersecuritymonitor 2019". In: (2019).

[8] Peter Bull et al. "Flow based security for IoT devices using an SDN gateway". In: *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud).* IEEE. 2016, pp. 157–163.

[9] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3 (2009), p. 15.

[10] Leyla Bilge et al. "Disclosure: detecting botnet command and control servers through large-scale netflow analysis". In: *Proceedings of the 28th Annual Computer Security Applications Conference.* ACM. 2012, pp. 129–138.

[11] Rick Hofstede et al. "Towards real-time intrusion detection for NetFlow and IPFIX". In: *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013).* IEEE. 2013, pp. 227–234.

[12] Benoit Claise. "Cisco systems netflow services export version 9". In: (2004).

[13] Gabriel Maciá-Fernández et al. "UGR '16: A new dataset for the evaluation of cyclostationarity-based network IDSs". In: *Computers & Security* 73 (2018), pp. 411–424.

[14] Mark Hall et al. "The WEKA data mining software: an update". In: *SIGKDD Explorations* 11.1 (2009), pp. 10–18.

[15] Hung-Jen Liao et al. "Intrusion detection system: A comprehensive review". In: *Journal of Network and Computer Applications* 36.1 (2013), pp. 16–24.

[16] Nwokedi Idika and Aditya P Mathur. "A survey of malware detection techniques". In: *Purdue University* 48 (2007), pp. 2007–2.

[17] Arthur Zimek and Erich Schubert. "Outlier Detection". In: *Encyclopedia of Database Systems*. Ed. by Ling Liu and M. Tamer Özsu. New York, NY: Springer New York, 2017, pp. 1–5. ISBN: 978-1-4899-7993-3. DOI: `10.1007/978-1-4899-7993-3_80719-1`. URL: `https://doi.org/10.1007/978-1-4899-7993-3_80719-1`.

[18] Tom Michael Mitchell. *The discipline of machine learning*. Vol. 9. Carnegie Mellon University, School of Computer Science, Machine Learning . . ., 2006.

[19] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. "Supervised machine learning: A review of classification techniques". In: *Emerging artificial intelligence applications in computer engineering* 160 (2007), pp. 3–24.

[20] Miroslav Kubat. *An introduction to machine learning*. Springer, 2017.

[21] Constantinos Kolias et al. "DDoS in the IoT: Mirai and other botnets". In: *Computer* 50.7 (2017), pp. 80–84.

[22] Prasanta Gogoi, Bhogeswar Borah, and Dhruba K Bhattacharyya. "Anomaly detection analysis of intrusion data using supervised & unsupervised approach". In: *Journal of Convergence Information Technology* 5.1 (2010), pp. 95–110.

[23] Duygu Sinanc Terzi, Ramazan Terzi, and Seref Sagiroglu. "Big data analytics for network anomaly detection from netflow data". In: *2017 International Conference on Computer Science and Engineering (UBMK)*. IEEE. 2017, pp. 592–597.

[24] Valentín Carela-Español et al. "Analysis of the impact of sampling on NetFlow traffic classification". In: *Computer Networks* 55.5 (2011), pp. 1083–1099.

[25] Haipeng Yao, Yiqing Liu, and Chao Fang. "An abnormal network traffic detection algorithm based on big data analysis". In: *International Journal of Computers Communications & Control* 11.4 (2016), pp. 567–579.

[26] Nigel Williams, Sebastian Zander, and Grenville Armitage. "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification". In: *ACM SIGCOMM Computer Communication Review* 36.5 (2006), pp. 5–16.

[27] M Mazhar Rathore, Awais Ahmad, and Anand Paul. "Real time intrusion detection system for ultra-high-speed big data environments". In: *The Journal of Supercomputing* 72.9 (2016), pp. 3489–3510.

[28] Dario Rossi and Silvio Valenti. "Fine-grained traffic classification with netflow data". In: *Proceedings of the 6th international wireless communications and mobile computing conference*. 2010, pp. 479–483.

[29] Jiangpan Hou et al. "Machine Learning Based DDos Detection Through NetFlow Analysis". In: *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE. 2018, pp. 1–6.

[30] Juliette Dromard, Gilles Roudière, and Philippe Owezarski. “Online and scalable unsupervised network anomaly detection method”. In: *IEEE Transactions on Network and Service Management* 14.1 (2016), pp. 34–47.

[31] Julina Zhang et al. “Comparing unsupervised learning approaches to detect network intrusion using NetFlow data”. In: *2017 Systems and Information Engineering Design Symposium (SIEDS)*. IEEE. 2017, pp. 122–127.

[32] Ruth G Shaw and Thomas Mitchell-Olds. “ANOVA for unbalanced data: an overview”. In: *Ecology* 74.6 (1993), pp. 1638–1645.

[33] Foster Provost. “Machine learning from imbalanced data sets 101”. In: *Proceedings of the AAAI’2000 workshop on imbalanced data sets*. Vol. 68. AAAI Press. 2000, pp. 1–3.

[34] Markus Ring et al. “A survey of network-based intrusion detection data sets”. In: *Computers & Security* (2019).

[35] Sebastian Garcia et al. “An empirical comparison of botnet detection methods”. In: *computers & security* 45 (2014), pp. 100–123.

[36] Samaresh Bera, Sudip Misra, and Athanasios V Vasilakos. “Software-defined networking for internet of things: A survey”. In: *IEEE Internet of Things Journal* 4.6 (2017), pp. 1994–2008.

[37] Ian H Witten et al. *The WEKA Workbench. Online Appendix for ”Data Mining: Practical Machine Learning Tools and Techniques”*. 4th. Morgan Kaufmann, 2016.

[38] Judea Pearl. “Bayesian networks”. In: (2011).

[39] S Vijayarani and M Muthulakshmi. “Comparative analysis of bayes and lazy classification algorithms”. In: *International Journal of Advanced Research in Computer and Communication Engineering* 2.8 (2013), pp. 3118–3124.

[40] Remco R Bouckaert. “Bayesian network classifiers in weka for version 3-5-7”. In: *Artificial Intelligence Tools* 11.3 (2008), pp. 369–387.

[41] Ian H Witten et al. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[42] Rausheen Bal and Sangeeta Sharma. “Review on Meta Classification Algorithms using WEKA”. In: *International Journal of Computer Trends and Technology (IJCTT)–Volume* 35 (2016).

[43] Fábbio AS Borges et al. “Comparison between random forest algorithm and j48 decision trees applied to the classification of power quality disturbances”. In: *Proceedings of the International Conference on Data Mining (DMIN)*. The Steering Committee of The World Congress in Computer Science, Computer . . . 2015, p. 146.

[44] Hong Bo Zhou and Jun Tao Gao. “Automatic method for determining cluster number based on silhouette coefficient”. In: *Advanced Materials Research*. Vol. 951. Trans Tech Publ. 2014, pp. 227–230.

[45] Deepak Kumar et al. “All things considered: an analysis of IoT devices on home networks”. In: *28th {USENIX} Security Symposium ({USENIX} Security 19)*. 2019, pp. 1169–1185.

[46] Mara Balestrini et al. “Onboarding communities to the iot”. In: *International Conference on Internet Science*. Springer. 2017, pp. 19–27.

[47] *Trend Micro Research Finds Major Lack of IoT Security Awareness.* `https://www.trendmicro.com/en_ie/about/newsroom/press-releases/2018/trend-micro-research-finds-major-lack-of-iot-security-awareness.html`. Accessed: 07-11-2020.

Part II
Science Communication

1 Introduction

In part I of this thesis, different methods were used to detect malicious messages in NetFlow data. The results of part I show that there are still too many false positives and false negatives to rely on such a system entirely, but it does indicate it could be feasible with more specialized techniques. If a system exists to find attacks in NetFlow data accurately, there is still another critical challenge, which is preventing or solving the problems. Since it is not feasible yet to create a system that automatically handles these problems in every case and not every person can afford services to monitor their network, users should protect their network themselves. However, this would mean transferring highly technical information to end users, who might not have the background knowledge to understand the information thoroughly. If a user cannot understand the message, it could lead to confusion or uncertainty about what to do, which means they do not protect themselves. This part of the report will go deeper into how messages can be presented to users to motivate them to act.

1.1 Problem description

Since more and more devices are connected to the internet, and more people have to deal with these devices, it is important that people are aware that these devices can bring new vulnerabilities and how they can protect themselves against these vulnerabilities. In the first quarter of 2020, internet penetration is 62% worldwide. North America has the highest rate with 90.3%, closely followed by Europe with a rate of 87.2%. Africa has the lowest rate with 42.2% [1]. The percentage of households with an IoT device in North America is 71%, in western Europe this is 57.2%, and for Africa, this number is just below 20% [45]. The global median is 40.2% of all households. These statistics make it clear that IoT has a large userbase worldwide which seems only to keep growing. However, IoT is still a relatively new technology, and what exactly it encompasses is also not clear to everyone. Nowadays, More and more devices are ‘smart’, such as a thermostat, which is often pre-installed in a new house. Not every user is even aware that they are a user; they might not know that a new device comes with new security challenges. These developments create a necessity for a user-centred approach in which their needs are prioritised.

The way an average user interacts with IoT or solves problems with IoT devices is not a big research topic. There is some research on how to get users to effectively integrate IoT into their daily lives by making the setup more accessible, such as the paper by Balestrini et al. [46]. However, research on users dealing with the dangers on their network related to IoT is minimal. There is some corporate research towards the awareness surrounding IoT security, such as research by Trend Micro and Vanson Bourne [47]. This research showed that "Only 14% of respondents say they have complete organisational awareness of IoT threats". Even though IoT security is becoming more important, little to no research focuses on individual users.

This research focuses on a system that sends a warning message to a user; so that this user can act as independent as possible. The biggest challenge in this research is that every user is different in preference, knowledge, and know-how. The difference between users requires a personal approach instead of a one-size fits all solution. This introduces the challenge that an automatic system can not 'know' the user.

Some people might not understand the message; others might not be sure about their competence. Worries such as these can be described as uncertainties. These uncertainties might prevent users from acting on the message, so the desired behaviour does not occur. The absence of the desired behaviour means the system is not successful. Some practical models will be used to concretise the link between uncertainties and behaviour.

Another important aspect in reaching the desired behaviour is the structure and content of a message. The message should contain all necessary information for a user without causing confusion or frustration. Sending the message should avoid evoking negative feelings or emotions. If a user can not act on a warning message, it can lead to a feeling of the problem being out of their control. These feelings can be triggered by factors such as a low level of perceived control. We want to prevent the warning message from inciting these feelings. It could also be possible that this is not possible, in which case it could be better to not show the warning at all but take another approach.

This thesis does not aim to provide a complete solution to the challenges described above since that would require a much larger scope. Instead, we will take a first qualitative at how users react to messages from a system. The following section shows a more in-depth overview of the goal.

1.2 Goal of the thesis

This research aims to find a way in which different users can be approached in a personal way to prevent negative emotions. These negative feelings, such as a feeling of incompetence, could prevent the user from taking action. It would be ideal to find a way in which data can be presented in a personalized way. A personal approach should lower the feeling of uncertainty or danger and stimulate the user to take action. If a user is, for example, unsure about his or her knowledge, it will most likely not help to give them a highly technical description of the problem.

The scope of this thesis does not allow for the creation of the entire system; instead, the purpose is to take the first step into creating a personalized approach in approaching users about their network, with a focus on behaviour and uncertainties. The idea is that users with similar worries can be approached similarly. If such groupings are possible, it allows for a semi-personal construction in which different user groups are approached in a way that fits their preference. This approach would require an initial setup to analyze to which group a person belongs. Another thing that could change how people look at attacks is whether they have a technical background.

1.3 Research questions

Based on section 1.1 and 1.2 the following main research question was formulated: *In what way can technical vulnerabilities in the context of IoT be presented to users to motivate them to take action, when taking personal uncertainties into account.* The answer to this question requires multiple sub-questions.

1. Which information of the previous part of this research will be used for this part?
2. How can a message be used to influence the motivation and behaviour of users so that they take action?
3. How do users respond to a different presentation of the same vulnerability when there is a focus on a different uncertainty?
4. Based on RQ3, can users be grouped based on their uncertainties.

RQ 1 will be based on part I of the report and a literature review. RQ 2 will be answered through a literature review. Based on the literature, a theoretical framework will be created. RQ 3 will be answered by qualitative research using interviews based on the theoretical framework created for RQ 2. RQ 4 will be based on an analysis of the answers given by participants in the interviews. Finally, these research questions will be combined to answer the main research question.

The rest of this paper will be structured as follows: Section 2 will present the theoretical framework based on literature. Based on this framework, the formulations of the warning messages are set up in section 3. Then the methodology is presented in section 4. The outcome of the interviews will be presented in section 5. Based on the outcome, the results are then presented in 6. Finally, the paper will be finished with a discussion in section 7.

2 Theoretical framework

In this section, we present the results of the literature review. These results will be combined to create a theoretical framework. The theoretical framework serves as the basis for the rest of the project.

This research aims to present possible dangers in a network in such a way that a user can act on it while feeling secure. We thus want users to be motivated to act when they receive a warning message, which means influencing their behaviour.

2.1 Theory on motivation and intention

Since the goal is for people to protect themselves, after reading a message from the system, a good starting point for the theoretical research is motivation and behaviour. One theory especially fits this purpose very well, the protection motivation theory, described in the following subsection.

2.1.1 Protection motivation theory

The protection motivation theory goes into how motivated people are to protect themselves from harm or dangers.Image 7 shows a schematic overview of this theory. This theory was created by R.W. Rogers in the field of psychology to better understand fear appeal [48]. Rogers later developed the theory to focus on persuasive communication instead of only fear appeal [49]. The theory is still actively used in modern research with a variety of applications where the goal is to persuade people to protect themselves, such as personal health.

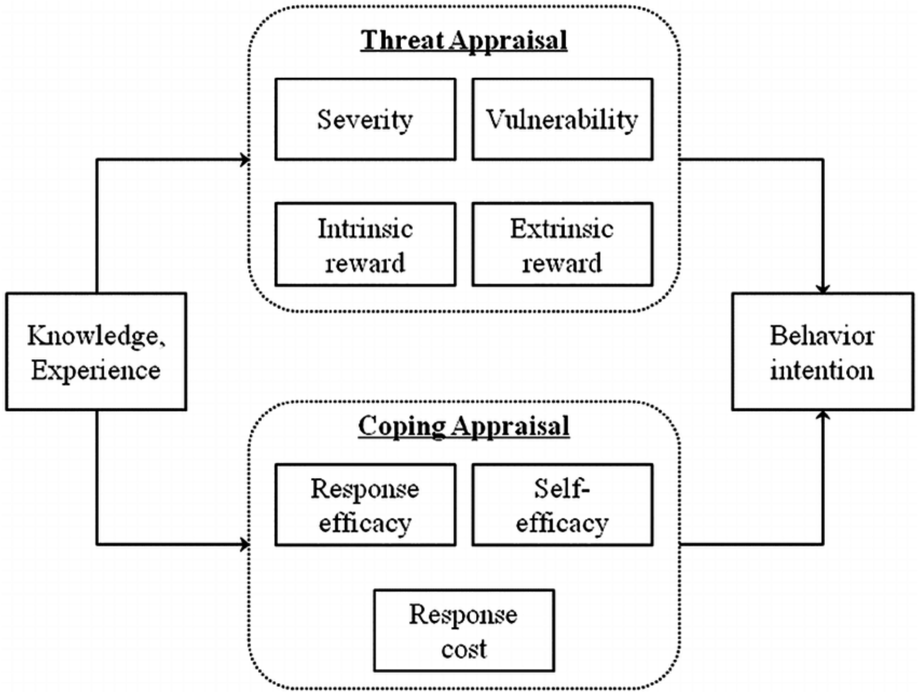


Figure 7: Protection Motivation Theory, which shows that the protection motivation of people is based on the threat appraisal and coping appraisal.

This theory shows that two different factors influence the protection motivation, threat appraisal and coping appraisal. It is important to note that these factors are subjective; they can differ from person to person.

The threat appraisal indicates how threatening the situation or problem is, according to a person. The factors that increase the threat appraisal are severity and vulnerability; these two factors describe how dangerous the situation is and how vulnerable the person feels. The rewards decrease the threat appraisal; both the intrinsic and extrinsic rewards describe the positives for a person not to take action. If the severity and vulnerability outweigh the rewards, there is a positive behavioural intention to take action. Keep in mind that an intention does not necessarily translate to action.

On the other side, there is the coping appraisal, which describes how well a person feels he can handle the situation. In this case, three factors influence the coping appraisal. The first factor, the response efficacy, shows how effective the recommended solution is and the person's belief in this solution. Self-efficacy is the level at which a person thinks he or she is capable of following the recommended solution. These two factors have a positive relationship with the coping appraisal. The response cost describes the cost of following the recommended solution; this could, for example, be the required time. In contrast to the response efficacy and self-efficacy, the response cost has a negative relation with the coping appraisal.

The threat appraisal and coping appraisal are both influenced by a person's knowledge and experience. If, for example, a person has taken care of a problem multiple times before, the self-efficacy will most likely be higher. In short, the theory can be summarized as follows: A higher threat appraisal and coping appraisal increases the behavioural intention to change the situation.

Since we want to motivate people to take action, this theory could imply that users should be given a solution to increase the response efficacy or that an attack should not be marginalized to prevent reducing the severity. However, this model describes how to influence the protection motivation or behavioural intent, but motivation can also be different for each person. We will need to take an in-depth look into motivation to see how we can increase it. The following section will focus on such a theory.

2.1.2 Self-determination theory

The self-determination theory shows different kinds of motivations, what causes them, and how to influence them. Image 8 shows a schematic overview of the theory. Ryan and Deci developed this theory in 1980, also in the field of psychology. The focus of this theory is on intrinsic and extrinsic motivation. Over the years, Ryan and Deci refined the theory and extended it to include the three psycho-social needs [50]. The model is still being worked on and is used in different contexts.

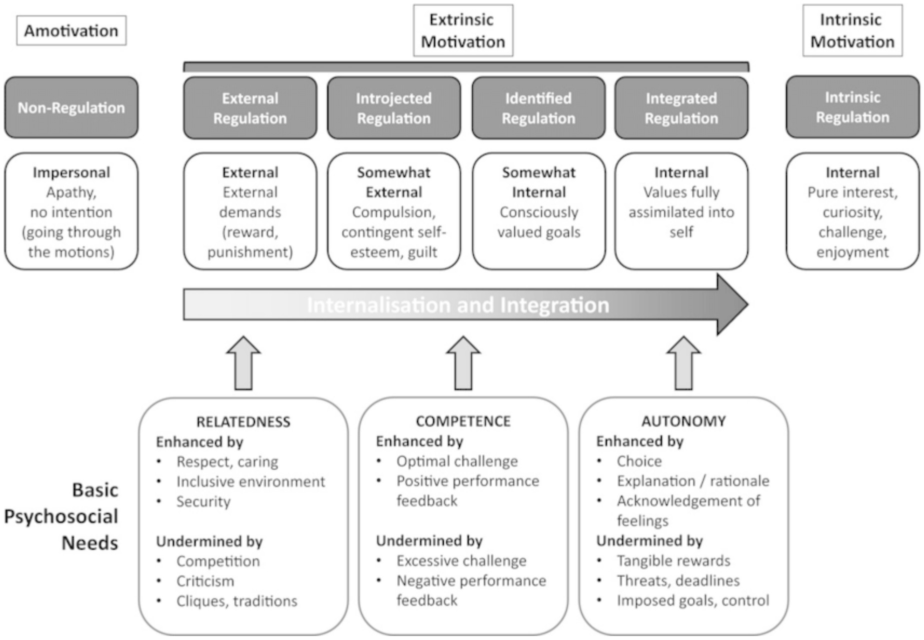


Figure 8: Self-determination theory, which shows the different types of motivation and the factors that influence them.

This theory describes three types of motivation, which we will go through one by one.

The first type of motivation is amotivation; this describes people that can not be motivated. Amotivated people have no intention to perform a specific behaviour. On the other side of the spectrum, there is intrinsic motivation. Intrinsic motivation describes the people that do something just for the joy of doing it; they do not need any external reasons. In the middle of these two types of motivation, there is extrinsic motivation. This type of motivation is more complicated and is a spectrum rather than a single type. There are four types of extrinsic motivation described in the model. The most extrinsic type is external regulation; this type of motivation depends on external demands, such as rewards or punishments. Next is the introjected regulation, which is more internalized but still more external than internal; factors such as compulsion or guilt can lead to this type of motivation. The first somewhat internal type is identified regulation; this means that someone consciously values the goals. Finally, there is the most internal version of extrinsic motivation, which is integrated regulation. This type of motivation happens when the values are fully assimilated into oneself, which means that a person sees it as a part of their own needs.

The model also describes that three basic psycho-social needs that can cause motivation to become more internalized. The three basic psycho-social needs are relatedness, competence, and autonomy. Relatedness describes how connected someone feels with their environment. Respect, security, and inclusivity can enhance the feeling of relatedness, while competition, criticism, cliques, or traditions can undermine it. Competence describes how capable people are in their environment. The difficulty of the challenge influences perceived competence; when the challenge is too complex, a person feels less competent. Performance feedback can also influence perceived competence; positive feedback can increase it, while negative feedback can decrease it. Autonomy is about the amount of control a person has in their environment. If a person understands the situation, has a choice to act, and his or her feelings are acknowledged, it increases perceived autonomy. However, the feeling of autonomy can also be reduced by, for example, tangible rewards, deadlines, or imposed goals.

For this research, self-determination theory gives several helpful pointers, especially the psycho-social needs. Since the intention is to make users take action, we want them to see it as necessary. In the setup of this research, it is not possible nor desirable to try and force people to take action due to external pressure. Since we do not focus on external pressure, we want the motivation to be as internal as possible. Because we focus on internal motivation, we look at factors that can increase the internalization of motivation. Amotivation and intrinsic motivation are of no concern for this research as the assumption is that amotivated people cannot be motivated, and intrinsically motivated people do not need to be motivated.

Based on the previous models, we can see that competence and self-efficacy are important factors for attitude and motivation. The following section will expand on self-efficacy.

2.1.3 Self-efficacy theory

A. Bandura presented a theory on self-efficacy in 1977 [51], yet again in the context of psychology, over the years, he refined and expanded the theory. His latest publication related to self-efficacy was published in 2015, next to that the theory is still often used. Bandura described self-efficacy as “how well one can execute courses of action required to deal with prospective situations” [52]. The main goal of the theory is to understand what influences people to be more or less confident in their ability to deal with a task at hand. Bandura describes four factors that influence self-efficacy, which we will describe one by one.

First, there is mastery experience, which attributes the most to self-efficacy. Mastery experience implies that people succeeded at the required action before; they experienced mastery of the topic at hand. If they succeed, it increases the feeling of self-efficacy, while failure will decrease this feeling. Next up, there are vicarious experiences. Vicarious experiences come from observing others; if a role model or someone similar succeeds, it can give a higher feeling of self-efficacy. Then there is social persuasion. When a person is told that they can succeed by someone they care about, they are more likely to put in the required effort, thus raising self-efficacy. Finally, there are emotional states. Emotional states influence how a person sees himself or challenges and can increase or decrease self-efficacy based on the emotion. Depression, for example, can give people a lowered feeling of self-efficacy.

For this thesis, not every factor is of interest. The premise is that users are at home and receive a message from a system; this means that the mastery experience can be different for each user. Vicarious experiences and social persuasion are also challenging to achieve. The intention is, of course, that self-efficacy increases over time as users gain mastery experience and get positive feedback after solving a problem. The main takeaway is that the problem should be presented in such a way that the problem does not overwhelm the user, as this makes it seem more mastery experience is required and can lower the emotional state. Suppose a user becomes uncertain or worries about their capability of solving a problem. In that case, it could lead to them entering a negative emotional state which negatively impacts self-efficacy.

2.1.4 Technology acceptance model

The technology acceptance model (TAM) was developed by F. Davis [53] in the field of information systems. The model is an extension of the, in 1967, developed theory of reasoned action by I. Ajzen and M. Fishbein [54]. The technology acceptance model goes into the requirements for users to accept new technology. The model was revised twice over the years, which resulted in TAM2 and TAM3, the newest version. The core stayed the same, but new factors gave a deeper insight into the model. Figure 9 shows a visual representation of the model. Venkatesh and Bala [55] proposed the depicted version of the model in 2008.

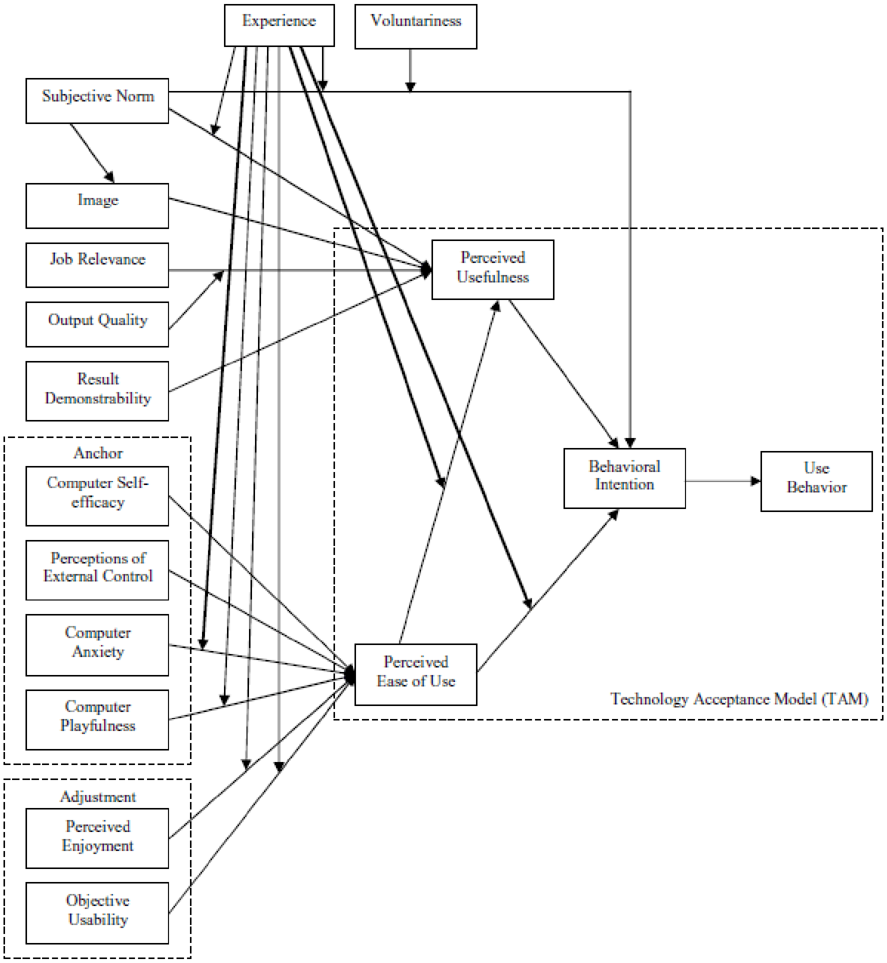


Figure 9: Technology acceptance model, which shows the factors that cause usage intention for a technology.

When we look at the figure, we can how the technology acceptance model evolved. The first extension added subjective norm, image, job relevance, output quality, result demonstrability, experience, and voluntariness to the model. The second extension added the anchor and adjustment. Venkatesh and Bala proposed the three bold arrows as a new addition to the model.

The original TAM shows that perceived usefulness and perceived ease of use make up the behavioural intention, which can then cause people to start using a specific technology. TAM2 gives more insight into what makes up the perceived usefulness while TAM3 expands on the perceived ease of use.

For this thesis, the model is not directly applicable since the point is to see which factors influence the intention to use a new technology. In this thesis, we assume that users already use the technology. However, certain aspects can help when structuring the message. For this, we will need to look at individual factors and determine whether or not they can help. The original TAM has perceived usefulness and perceived ease of use as main factors; this implies that the message we send should also feel useful and easy to use. Increasing the perceived value of a warning message can be achieved by, for example, adding enough relevant information in such a way that users with different backgrounds can understand it.

2.1.5 Elaboration likelihood model

The elaboration likelihood model was developed by R.E. Petty and J. Cacioppo in the field of psychology and published in 1986 [56]. The model focuses on persuasive communication and specifies two paths in which a person can interpret a message; a central route and a peripheral route. The central route is based on careful consideration of a person's system of values and beliefs. If a person changes their attitude through this central route, the change is often long-lasting. The peripheral route arises when a person receives or deduces cues from a message, such as the presentation.

Another aspect of the elaboration likelihood model is that people are often not easily moved by a single message; their previous attitudes and beliefs are essential in deciding whether to act on a message or not.

For this research, we use the central route. We choose for the central route because we do not know how people read and process the messages we want to send. First, we need to determine how well the central route works, which will serve as a baseline for further research in which peripheral strategies can be explored.

The previous models mainly focus on motivation and intention. Motivation and intention are not always enough to translate into behaviour. The following section will present two practical models that go directly into behaviour. We chose these models because they are practical and broad while giving insight into how to approach behaviour.

2.2 Theory on behaviour

Two models aimed at achieving specific behaviour and are practical and easily applicable are Fogg's behaviour model, described in section 2.2.1 and Wendel's CREATE action funnel, describe in section 2.2.2.

2.2.1 Fogg's behaviour model

Fogg's behaviour model claims that behaviour is a simultaneous combination of motivation, ability and a prompt [57]. This model was created in 2007 and updated in 2017. Figure figure 10 shows a visual representation of Fogg's model. The primary purpose of this model is to make behaviour easier to understand, "What was once a fuzzy mass of psychological theories now becomes organized and specific when viewed through the Fogg Behaviour Model" [57].

As shown in figure 10, the model states that "B=MAP at the same time", which means that behaviour occurs when there is a prompt, enough motivation, and enough ability. A prompt can be anything that signals a person to do something, such as an itch leading to a person scratching or a conversation about health leading to a person working out more.

According to the model, a prompt succeeds whenever there is enough of the combination of motivation and ability. Motivation is the measure of how badly a person wants to perform the behaviour. Ability is the measure of how easy it is for the person to perform the behaviour. From this, it follows that an action that is very easy to perform needs less motivation. For most people, it is easy to scratch an itch, so they do not need a high level of motivation. Working on one's health by eating healthier and working out can be more challenging and require more motivation.

In this research, we want to send a message to users that they can then act. Following this model, we can see that the message will act as the prompt, which will only succeed when the ability and motivation are high enough.

Next to the ability to perform the behaviour, another type of ability that should be taken into account is how easy it is to interpret the prompt. If a person is motivated and able enough to perform the behaviour, this person will still not perform the behaviour if he or she cannot understand the prompt.

The advantage of this model is that it is relatively straightforward and very relevant to the challenges in this research. Not everyone has the same technical knowledge and know-how about computers, the internet, and malicious network traffic. Boerman et al. [58] found that "people have little confidence in their own efficacy to protect their online privacy". In their research, they asked 928 people what kind of protective measures they use to protect their online privacy. From the responses, they saw that "the perceived efficacy seems mixed" and that it also differs from person to person. Using Fogg's behaviour model, we can make sure that relevant questions can be included in the design of this research to give us insight into how a prompt can be successful.

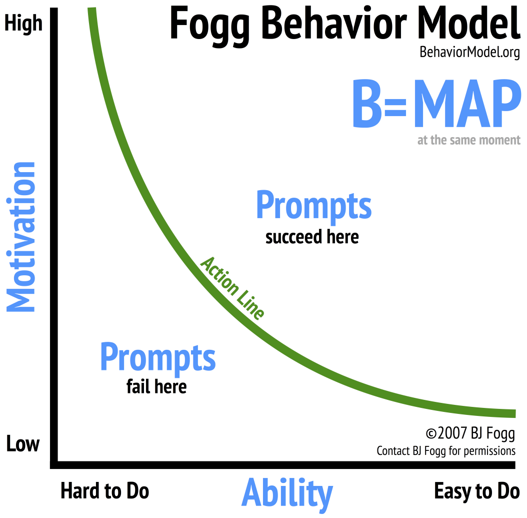


Figure 10: Fogg's B=MAP model, which shows that a combination of motivation and ability is necessary for a prompt to lead to behaviour. Permission for usage of image was obtained by contacting B.J. Fogg.

2.2.2 Wendel's CREATE action funnel

Wendel's action funnel [59], which is visible in figure 11, is a model for creating a service or product to influence behaviour. This model came to be when Wendel was looking at when certain behaviour occurs. In his book *Designing for Behaviour Change* [59] he states that there is an "odd sort of logic to how we decide to take one action instead of another. That logic can't tell us how to force someone to take a different behaviour, but it can help us set up the right conditions for action, if the person chooses to do so". The model describes five different stages which a person goes through before they take action. These steps are as follows:

- Cue: What makes a user think about performing the behaviour?
- Reaction: What is the initial, split-second reaction of the user?
- Evaluation: Do the benefits outweigh the costs?
- Ability: Is it possible for the user to perform the behaviour?
- Timing: How urgent is it? Can the action wait, or should it be done immediately?

Every type of behaviour can be used to create an example for these steps; in this case, we will use drinking water.

- Cue: I feel thirsty; maybe I need to drink something.
- Reaction: When I drink something, my thirst is reduced or removed.
- Evaluation: I have been trying to cut back on calories, so I should rather take water than a soda.
- Ability: I can drink water straight from the tap, and getting there is no problem.
- Timing: I need to submit my paper within the next five minutes, so I should focus on that.

In this case, the person would not execute the behaviour; however, if the feeling of thirst is still there after timing is not a problem anymore, this person will most likely go and drink some water. Wendel does state that for behaviour to occur later, the five steps are still necessary.

Examples of the usage of this model are designing smartphone apps [60], or when looking at behavioural design in tourism [61]. One of the strengths of this model is that it is very generally applicable. No matter how small or large an action may be, this model can serve as a basis for creating a situation where the behaviour is easier to perform.

In this model, there are some parallels to Fogg's behaviour model. The cue of Wendel is the same as the prompt in Fogg's model. Both models include ability, which means the same in both. One of the main differences is that the result of the CREATE action funnel is a state in which the behaviour does or does not occur, while in the case of Fogg, it is more like a spectrum in which it is linked to the amount of ability and motivation. Motivation does not directly play a role in Wendel's model, but it is a part of the evaluation. If a person has a high motivation to perform a certain type of behaviour, the perceived benefit can be higher, or the perceived cost can be lower. When we come back to the example of drinking water: If a person is very thirsty, he or she might be more motivated to drink some water. So while it is not directly a part of the model, it does influence the reaction, evaluation, and timing.

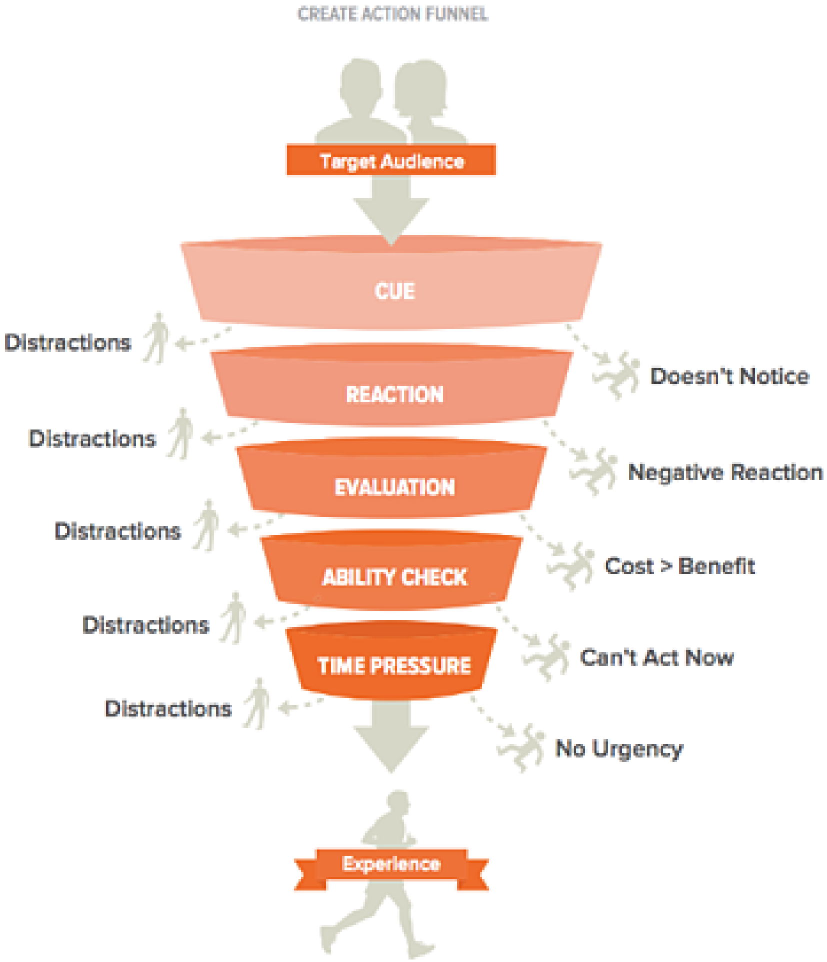


Figure 11: Wendel's CREATE action funnel explains which stages a person has to go through before they take action and which factors can lead to a midway exit. This image is taken from the book D*esigning for behaviour Change* by S. Wendell [59]

.

Popular marketing tactics often try to 'cheat' steps by playing into aspects of this model. A sale that only lasts one more hour, for example, creates artificial urgency and gives potential buyers less time for the evaluation. For this research, out aim is not to cheat since the goal is a consciously chosen action.

One aspect that can have a significant impact on such models is uncertainty. Uncertainty is, logically, something a person is uncertain about. Uncertainties can be caused by, for example, negative consequences or a lack of knowledge about the situation [62].

These uncertainties can cause users not to perform the desired behaviour. Wendel's CREATE action funnel shows that a negative reaction can cause users not to follow the funnel. A lack of knowledge or information, for example, can cause such a negative reaction. The uncertainty about a consequence, especially fear for a negative consequence, can increase the perceived cost in relation to the benefit. The ability check can be influenced by the uncertainty of one's perceived competence.

The Fogg Behaviour Model shows a combination between motivation and ability. Just like in Wendel's CREATE action funnel, uncertainty about perceived competence can reduce the feeling of ability. A lower ability then requires a higher level of motivation and might lead to the desired behaviour not occurring. It becomes clear that uncertainties can influence or prevent certain behaviour. For this reason, the next section will go into detail about different types of uncertainties.

2.3 Theory on uncertainty

In this research, an ideal scenario would be a user receiving a message and taking action in a short amount of time. Uncertainties can cause users not to take action when presented with a message; they might not understand the information, they might not be able to fix the problem, or the problems might scare them away from using services in general. Uncertainty affects behaviour and will often result in preventing certain behaviour or stimulating behaviour to reduce the uncertainty. In a social context, Kappes et al. [63] found that impact uncertainty can cause people to engage in social behaviour to decrease the uncertainty; outcome uncertainty had the opposite effect.

In the context of attacks on a network, users should act on a notification without uncertainty preventing them from proactive behaviour. The effect of uncertainty in this field has not been studied extensively yet, so while we can draw certain parallels to create a possible link between information and uncertainty, they will have to be tested.

Uncertainties can be caused by various reasons, of which some will be relevant to this specific field. There is a wide variety of taxonomies on uncertainties, so we need to make a choice. We chose four different uncertainties that serve as a common theme within taxonomies. Uncertainty about complexity, technology, or a lack of knowledge about the topic is an essential factor in most taxonomies. Another one is the relationship or diversity of actors. Consequences are also a logical but reoccurring uncertainty. Finally, competence uncertainty is relevant when a person has to perform something him- or herself. The following sections will briefly describe these types of uncertainty based on the literature in which they were found. The full literature review can be found in appendix F.

2.3.1 Uncertainty about the complexity or technology

This type of uncertainty is based on the understanding of the topic at hand. When looking at the literature review of H. Jalonen et al. [62] it comes down to the following: If a person does not understand the topic due to an unfamiliar technology or complexity of the subject, this uncertainty can become more of an issue. This uncertainty is relevant to the topic since the messages try to convey rather technical information that not every user might be familiar with. From now on, when we mention*complexity uncertainty* we imply this uncertainty.

2.3.2 Uncertainty about the intention of the attacker

The more general version would be uncertainty about other actors in the same system. In the context of this research, the other actor is the attacker. An essential point in this type of uncertainty is trust. *Why is a person trying to get into my network?* or *Can I trust this message I got from someone I do not know?* are questions that can arise. People can be unpredictable, and in cyberattacks, an attacker does not have your best interest in mind. If another actor does not seem trustworthy, this uncertainty can increase. This uncertainty is relevant because an attacker is another actor with malicious intentions. This uncertainty will from now on be referenced with *attacker uncertainty.*

2.3.3 Uncertainty about consequences

This type of uncertainty is relatively straightforward; If a person does not know the consequence of a situation, it can increase their uncertainty. The severity of the consequence is also a factor. A more severe consequence adds more uncertainty and doubt to decision making. An attack on a network can have serious consequences, and it follows that this type of uncertainty is applicable. From now on, this will be called *consequence uncertainty*

2.3.4 Uncertainty about competence

The final type of uncertainty is based on the perceived competence of a user. When a person believes that a task is outside their capabilities, it will result in a feeling of uncertainty. The amount of know-how and knowledge of the topic at hand can influence the perceived feeling of competence. Not every user is used to taking care of problems with their computer themselves, let alone their network, which makes this uncertainty relevant. This uncertainty will be referenced as *competence uncertainty*

After showing all these theories, we can now combine them into a single framework. The following section will go into the combination of different theories and their consequences for this research.

2.4 Framework

We can use different theories to tackle different parts of the problem. However, these theories need to be combined to create a robust theoretical foundation for the entire research. Do theories work together or against each other in certain aspects? Do the theories have common ground in which they can support each other or be combined? In this section, we will have a critical look at the theories and how they can be used together to create the underlying theoretical framework of the research.

The first model we looked at was the protection motivation theory. This model has two key factors as a focus, the threat appraisal and the coping appraisal. The threat appraisal is influenced by severity, vulnerability, intrinsic rewards and extrinsic rewards. The intrinsic and extrinsic rewards can be linked to the type of motivation in the self-determination theory. We want people to protect themselves from outside threats just by showing them a warning. Acting on a single message requires a rather intrinsic type of motivation since we are not rewarding or punishing the user for taking action or not. However, the intrinsic values in both models have an inverse relationship. An intrinsic reward lowers the threat appraisal and makes it less likely for the person to take action; this also means there is less intrinsic motivation to take action. Increasing the intrinsic motivation to take action should lower the intrinsic reward for not taking action. For this, we can take a look at the basic psycho-social needs of the self-determination theory.

It would be possible to link the system to rewards or punishments and try to motivate people through extrinsic means. However, this would lead to numerous different options and not a clear baseline for how people react to a warning message. We aim to reach people by using the central route, as described in the elaboration-likelihood model. Using different means would result in trying to use a peripheral route, and although this is interesting in itself, it would result in less clarity in the results.

The other key aspect of the protection motivation theory is the coping appraisal, consisting of response efficacy, self-efficacy and response cost. The self-efficacy here can be explained using the self-efficacy theory from Bandura. Most of the aspects here, such as vicarious experiences and social persuasion, are peripheral approaches. Mastery experience needs to be built over time but can be improved if the problem is made less intimidating. The emotional states can be influenced by a person's uncertainties, leading to a negative emotional state.

According to the technology acceptance model, self-efficacy can also be increased by the perceived ease of use but lowered by computer anxiety. Since computer anxiety is a negative emotion, we can link it to the emotional state described by Bandura.

Fogg's behaviour model shows us that a sufficient combination of ability and motivation is necessary to let a prompt succeed in making someone take action. In our case, the prompt is the warning a user receives. The ability can be linked to self-efficacy; if the self-efficacy is higher, it means the user has less trouble solving the problem. The motivation is the same as in the self-determination theory, which means we aim for intrinsic motivation. However, if a person is not intrinsically motivated, we can still try to let the prompt succeed by making it easier for the user. If the problem seems more manageable, the feeling of ability is higher, which results in a lower need for motivation.

Wendell's CREATE action funnel once again shows us the ability is an essential factor. The cue here is again our warning message. The reaction is linked to the emotional state; when the warning immediately causes a negative reaction, there is a chance the user will not take action. The question of whether the benefits outweigh the costs is influenced by one's motivation, which means that the evaluation is also linked to motivation. If a person is very motivated, he is willing to put more effort into solving a problem. Since the central route would mean showing a problem as soon as it is found, timing is not part of this research.

All in all, the theories point to several important aspects. A higher intrinsic motivation will increase the desire to take action without external rewards or punishments. A higher threat appraisal will increase the behavioural intention to take action. A higher self-efficacy will increase the coping appraisal and thus the behavioural intention. A higher self-efficacy can be stimulated by lowering the negative impact the threat has on the person. This does, however, also risk lowering the perceived vulnerability or severity. This means that the perceived vulnerability and severity can both stimulate or prevent people from acting. Creating a situation in which the user feels the need to act without creating a negative emotional state is an important balance. According to Fogg, A higher level of motivation and self-efficacy also increases the chance for the warning to be successful. The factors mentioned before can also be linked to Wendel's CREATE action funnel and yield a higher chance of action when a message is presented.

The focus in this research, when it comes to emotional states, are uncertainties. Uncertainties can lead to a negative emotional state, which is what we want to prevent. A warning message should show the severity and vulnerability without triggering a user's uncertainties.

2.5 Takeaways for formulations

Based on the previously discussed theories, we can create multiple requirements for presenting a warning message to users. First of all, based on the protection motivation theory, we can see that a user should be aware of the threat since it increases the threat appraisal, increasing the protection motivation. However, as Bandura remarked, a too high threat can lower self-efficacy, which lowers the protection motivation. The threat can be linked to the consequence uncertainty. The threat should thus be clear but not too intimidating.

Self-efficacy is an essential factor in the decision to take action. Since not every user will fully understand the technical information or will be able to take technical steps, they should be supported by the message. A supportive message could mean including a solution and trying to exclude as much technical information as possible. However, including a solution might increase the self-efficacy, but users still need to believe the solution works since the response-efficacy also plays a role in the behavioural intention.

The message should be presented in such a way that the users can understand and solve the problem; this increases the feeling of competence and reduces uncertainties. The message should also only notify the user; if too many parts of the message stress the need to act immediately, it can undermine the perceived autonomy, which reduces the intrinsicness of the motivation.

Based on the technology acceptance model, we want the users to perceive the message as useful and easy to use. Useful could imply including a complete overview and helping the user solve the problem with as little effort as possible. Making a message easy to use most likely implies reducing the amount of technical information since this could be perceived as challenging for someone without prior knowledge.

Wendell and Fog show us that the message can trigger the user to take action. While the specifics on how the message is sent are not part of this research, it does show that it should attract the user's attention.

When looking at the message that we want to be convey to the user, it is thus important to try and structure it so that uncertainties do not prevent the user from acting. However, these uncertainties might be relevant and have a feasible link to the specific topic; there has been no research yet to see how much these uncertainties affect the reaction of users to a message they receive. To get insights on the effect of these uncertainties on the user's reaction, they need to be presented in an encapsulated yet similar way. A fixed structure of the message makes sure that the impact of different uncertainties can be compared without the structure of the message being an influencing factor.

3 Literature for formulations

In this chapter, we will go into the requirements for the formulations. Based on section 2.5 we can see that the theories can serve as a basis for creating the formulations presented to the users. However, these formulations need to be created in a consistent format. For this, we will first take a look at information structures.

3.1 Information structure

Each type of text can be presented in a certain way, shape or form. Similar to structures for a sentence, such as the need for at least a subject and a verb, there are also structures for larger pieces of text. These structures are not a necessity, but they can help maintain consistency and make sure every necessary part is included. Examples are a problem structure or a research structure [64]. Since this research aims to present information to a user, such a structure can act as a solid baseline to ensure consistency of the messages. This consistency is important so that users know what is happening and what they can do about it.

The user will see an overview of anomalies in the data; each anomaly represents a possible attack on a device or network. This makes each anomaly a possible problem that the user might have to deal with. According to Steehouders [64] and Jansen [65], there are different structures to present different types of text, such as a problem structure or a measure structure. Since we do not want the formulations to be too long and complex, we need to determine which structure is the best fit. Since we are presenting a problem to users, the first focus is a problem structure, which is defined as follows:

- What is the problem?
- Why is it a problem?
- What is the cause?
- What is the solution?

This structure allows for a complete overview of what the problem is and what the solution is. Since this structure includes all relevant information, we will not include other information structures.

Now that there is a fixed structure that acts as a basis for presenting the messages, it is important to look at how this structure is used. Different factors of an attack are described by different types of information. We will take a look at which types of information are described in the literature. The type of literature allows for a framework that ensures each part of the structure contains a specific type of information. This will further increase the consistency of presentations between different attacks.

3.2 Type of information

Not all information and knowledge is the same. Some knowledge is gained by studying, while other knowledge is gained by experience. Knowing what a bike looks like is possible without ever seeing a real bike since it is a collection of characteristics. Being able to ride one will take practice.

When it comes to knowledge, different types are defined, such as situational, conceptual, procedural and strategic [66][67]. Knowledge itself is not what we want to include in this research. While knowledge will help a user go through a solution's steps, it is not possible to give a user knowledge with just a message. A message can only convey information; thus, it is more interesting to see which types of information there are.

According to N. Ummelen [68], there is a distinction between procedural and declarative information. Declarative information describes properties or gives information about something. Declarative information answers the question 'what is?'. Procedural information describes how a task can be performed and answers the question 'how to?'. This also works for uncertainties; if a person has uncertainties about a task, he or she is uncertain about **What to do** or **How to perform the task**. In the case of declarative information, a person is uncertain about **What the information means**. When looking further into the literature, it becomes clear that this distinction is still used as a basis for further research, such as in the paper of M.T. Ullman [69] or van Schalkwijk et al. [70].

Using the information structure and type of information makes it possible to present different problems consistently. There is one final part left, which is the text readability. The text's readability should not influence how users interpret the messages when reading about different attacks. When, for example, a sentence contains many sub-sentences, it can cause an increase in difficulty when reading a text, which in turn can lead to a lesser understanding of the text.

3.3 Text readability

To make sure that formulations are not more intimidating simply because the text itself is more difficult to read, we should look at text readability. The formulations should be readable for a general audience and be consistent in their difficulty. If this is not the case, the perceived severity of an attack might be influenced by a difference in the readability of the message. One challenge here is that different parts of a problem statement will yield different readabilities; guiding a user through a solution will require more text than stating the problem.

We made the choice to use two different readability concepts to make the textual readability between attacks as similar as possible. These concepts are the type of a sentence and the Flesch-Kincaid reading ease, which will be described in the following sections.

3.3.1 Type of sentence

There are three types of sentences in the English language; simple, compound, and complex [71]. A simple sentence consists of a single independent clause and has at least a subject and a verb. A compound sentence consists of at least two independent clauses linked to each other, for example, because of relevancy. A complex sentence has at least one independent clause and a dependent clause. A complex sentence should not be confused with complexity as an uncertainty. A dependent clause can not be interpreted without the clause it depends on. This concept ensures that each part of the problem structure is consistent between different attacks. By ensuring this consistency exists, the types of sentences are not a variable to consider when analyzing the answers.

3.3.2 Flesch-Kincaid reading ease

A classical but still often used way to determine the readability of a text is the Flesch-Kincaid readability test [72]. This test takes a piece of text and calculates a score based on the number of words, syllables and sentences. The following formula is used to calculate the readability:
$206.835 - 84.6 * (\textit{total syllables}/\textit{total words}) - 1.015 * (\textit{total words}/\textit{total sentences})$
The higher the score, the easier the sentence is to read. Flesch gave a rough guide to link scores to school levels, shown in table 15.

Table 15: Flesch score related to the school level in which the text should be readable

Score	**School level**
90 to 100	5th grade
80 to 90	6th grade
70 to 80	7th grade
60 to 70	8th and 9th grade
50 to 60	10th to 12th grade
30 to 50	college
0 to 30	college graduate

Flaounas et al. [73] conducted research in which they calculated the Flesch score for different articles in different newspapers. They find that the Flesch score for most articles are between 40 and 50, where only sports achieves a score higher than 50 while business, science, environment and politics score under 40. Since most people should be able to read an average article in a newspaper, the aim for the messages presented to the user will be a Flesch score of 50 or higher. Next to achieving a high Flesch score, it is also important that formulations of different attacks have a similar Flesch score. One problem is that an increase of a single syllable can significantly affect the Flesch of short sentences.

3.4 Conclusions for formulations

Based on the problem structure, type of information and text readability, there are multiple requirements we can extract for the formulations.

- Each vulnerability needs to contain the four parts of a problem structure.
- Each part of the problem structure needs one type (declarative or procedural) of information.
- Each formulation needs an addition for each uncertainty.
- Each formulation needs to be comparable in text readability and type of sentence between attacks.

4 Methodology

This section describes the methods used in this research. It will first briefly describe how literature was found, how the interview and assignment for the interviewees are constructed, and how the interviews are held.

4.1 How was literature found?

In this section, we will describe how we searched for literature. Some literature was recommended by experts in a relevant field. The used models were known beforehand but further investigated by reading literature.

Uncertainty is a rather strange topic to find literature about since it can have different meanings, and the definition is rather broad. Most literature about uncertainty is, for example, about uncertainty in scheduling problems or measurements. The type of uncertainty that is relevant for this research is a more personal uncertainty. While uncertainty about whether or not the internet will work today is also interesting, it is not the focus of this research. Next to that, uncertainty is relevant in various fields such as human-computer interaction, management or health care. To not exclude any relevant fields, the search should be broad in scope.

Martín-Martín et al. [74] found results that suggested that "Google Scholar citation data is essentially a superset of Web of Science and Scopus, with substantial extra coverage". Next to that, the same authors state that "GS has been shown to be reliable and to have good coverage of disciplines and languages, especially in the Humanities and Social Sciences, where WoS and Scopus are known to be weak". Since communication is a social science, we will mainly use Google Scholar for the literature search.

One disadvantage of the inclusivity of Google Scholar is that not every result is as valuable; it can, for example, include older versions of papers or papers that have not been peer-reviewed yet. The inclusion of less reliable results is why it is extra important to look at whether we can use these results. For this reason, we created several criteria on which we judged a result: The title, the publication date, the number of citations and the source it was published in. Depending on the type of literature, some criteria were used differently or were more or less relevant. When looking for classic models, the publication date is less relevant. When looking for statistics, for example, it is essential to have a recent publication. The more classical literature was checked on whether they are still relevant by looking at more recent publications that cite older literature. Another point to keep in mind is that more recent publications often have fewer citations. Next to selecting literature, we used multiple academic methods to find more literature, such as the snowball method, in which we use references to dive deeper into the topic, or a citation search, in which we use citations to see which further research was performed.

Once we used a search term, we took several steps to select literature. The first step is to take a look at the titles to make sure it is relevant. If we found a relevant title, we looked at recency, citations and the source to ensure the found literature is reliable to prevent including sources such as interviews or lectures that can appear on Google Scholar. Once an article seems relevant and reliable, it was opened and inspected further, starting with the abstract.

The first step was a narrative literature review focusing on uncertainty since this gives a broad overview of relevant topics. An example of a search term is 'uncertainty AND systematic literature review'. Following the steps described above, this search resulted in two articles: One by G. Magnani [75], which is very recent, and one by H. Jalonen [62], which coincidentally was also recommended by an expert in the field of Science Communication.

After we found some literature reviews, we used a snowball search based on keywords from previously found literature. An example of this is the search for 'relational uncertainty', which resulted in two articles by L.K. Knobloch [76][77]. A snowball search allows for a deeper dive into the literature.

For the type of information, we also used a snowball search. It started with a recommendation for a paper by N. Ummelen [68] on declarative and procedural information from which citations were used to find more literature. This type of search did often not end up directly in the research and is thus not cited either. It did, however, lead to new keywords and a deeper understanding of the topic at hand.

We knew about B.J. Fogg's behaviour model [57] and Wendels CREATE action funnel [59] beforehand, but performed a literature search to find out whether these models are still relevant. We checked the relevance through a citation search. We search for the original literature and look at which newer papers cite this model. We could see that B.J. Fogg is still actively working on the model and that the model is still being used in recent research. Next to that, we can see that these articles are published in reliable sources such as IEEE or ACM.

4.2 Construction of the interview

Based on the theoretical framework presented in section 2, we need to construct a list of questions to answer the research questions raised in section 1.3. Since uncertainties are a big part of this research, we chose to split the interview into two parts. The first part has a more theoretical approach, where we try to find out which type of uncertainty is most prevalent. This first part consists of a small assignment in which participants rank the different formulations based on the theoretical framework. A second part is more practical and goes into how people approach network security and what they would like to see when it comes to notifications if they want to see any at all. The second part is a semi-structured interview, which means that the questions act as guidance through a conversation in which the interviewee can speak freely. We chose a semi-structured interview since this allows the interviewee to speak more freely, for example, on topics that are not directly related but can still be interesting.

4.2.1 Construction of the assignment

In this part of the interview, the primary purpose is to find out how users react to formulations with a focus on the different uncertainties, as seen in 2.3. The focus of this part is to answer research question 3: *How do users respond to a different presentation of the same vulnerability when there is a focus on a different uncertainty.* To achieve this, we need a vulnerability and several formulations for this vulnerability. Since we specified four types of uncertainties, it follows that we need four formations.

Since we will perform a small scale qualitative research, it is vital to have enough data for an in-depth comparison. Giving participants only four formulations for a single vulnerability would not lead to enough data. Additionally, it would also not show whether a participant is consistent in their rankings for the vulnerabilities. Because of this, we chose to take four vulnerabilities, with each four formulations; one for each of the uncertainties. The interviewees then have to rank each formulation on how worrisome they perceive the formulation. These four different vulnerabilities are taken from part I of this research.

These four vulnerabilities have different challenges. Based on these challenges, we hypothesise that each uncertainty is linked to one of the vulnerabilities. The hypothesis is that if people are more uncertain in a specific field, the corresponding attack will also be ranked higher. To test this, we ask the participants to rank the attacks themselves after ranking the different formulations. We will now give a short overview of each attack, together with the hypothesized link to the uncertainties.

Blacklist
A blacklist attack implies that the source is on a public blacklist and is trying to send you data. This attack has a clear link to attacker intention uncertainty. A blacklist implies that the sender that is sending to your network is determined to be untrustworthy. If a sender can not be trusted, why is this person sending anything to your network?

DoS
A DoS attack means an attacker is flooding your network. When a network is flooded, it can not handle all incoming data, which degrades the performance. Even though a DoS attack is usually not dangerous on a home network, it can be challenging to understand why the network is slow or disabled. It requires knowledge of how messages are sent and received over the internet, how routers work, how devices can work together to attack a network, etc.

Botnet
A botnet attack tries to take over vulnerable devices in your network and use them for personal gains. While most of these attacks have rather tame consequences or ones that only last a relatively short time, a botnet attack can have rather severe consequences. Attackers can hack your devices, steal your data or, among others, use your devices to perform fraudulent activities. We can see that a botnet can have grave consequences, which gives botnet a possible link to consequence uncertainty.

Port Scan
In a port scan, an attacker scans the router for open ports. When a port is open it could mean an attacker can intrude in a network. Even though a port scan attack itself does not have the highest impact, it can be complex to protect yourself against it if you do not have any previous knowledge or experience. The complexity gives a possible link to competence; the steps you have to take to solve this problem are not always clear since every router can have a different way to tackle the problem.

One of the challenges with presenting warning messages is that each vulnerability has its consequences, which might not always be apparent. In a personal network, a DoS attack is often more annoying than dangerous. The consequences of a DoS attack are immediate and noticeable. A botnet attack can have more severe consequences, such as theft of data, but these are not always noticeable for the average user.

When we take the uncertainties and vulnerabilities together, it results in sixteen different formulations; four attacks with four formulations that focus on a different uncertainty. The following subsection will show the different formulations for each attack.

4.2.2 Basic version of formulations

Based on the requirements in section 3.4 we can create a basic version for each attack. These basic versions contain each part of the problem structure. In this chapter, we will show a formulation for one attack. A full overview can be found in appendix G.1. We will use a blacklist attack to illustrate the formulations.

blacklist
Incoming data is suspected to be a blacklist attack. An untrusted source is sending data to your network. There is most likely malicious data on your network. This problem can be solved by blocking the IP address.

4.2.3 Uncertainty focused additions for formulations

Next to the basic version, an addition for each uncertainty was necessary. These additions had to be consistent between different attacks. The type of addition for each uncertainty is explained here:

Complexity
Since complexity concerns the understanding of the entire system it is possible to take two different approaches; make it more or less difficult. If a user is unable to understand the basic version, making it more difficult will most likely not change too much. Making it easier could lead to a better understanding of the dangers. Because of this, we chose to add a metaphor, which makes the dangers easier to understand.

Consequence
The consequence is rather straightforward; the minimal version contains a consequence of the attack, while the addition is a more severe consequence of the same attack. Showing a more severe consequence increases the danger and should have a larger impact on people with a higher consequence uncertainty.

Competence
In this research, we link competence with the solution. The solution consists of several technical steps to solve the problem. The minimal version only contains the solution, while the addition includes the steps to reach this solution. We expect that people who are worried about their competence are more worried about the technical steps they have to take.

Attacker intention
Finally, there is uncertainty about the intention of the attacker. While this link is less clear than the other this fits in the description of 'Why is it a problem'. The answer to this part of the problem structure is always an action of another actor. The minimal version only contains the direct actions and the addition shows more in-depth information about what the attacker is doing.

The combined version will again be illustrated using a blacklist attack; appendix G.2 shows the formulations for all attacks. Keep in mind that users will not see these formulations. Instead, we will show them four formulations that each contain a single uncertainty specific addition. Appendix H shows the structure of these formulations

blacklist
Incoming data is suspected to be a blacklist attack. An untrusted source is sending data to your network. The attacker has been caught performing suspicious activities and is now sending data to your network. There is most likely malicious data on your network, it can however also lead to your network or desktop being infected with a virus. This attack can be compared to a person with a criminal record that is trying to find locations where he or she is not

known. This problem can be solved by blocking the IP address.

1. Open your router settings page, this can often be done by surfing to 192.168.0.1 in your browser
2. Log in with the admin credentials, these can often be found in the manual
3. Search for Firewall or Block Sites configuration page, these might be located under a Filtering or Security page
4. Add the IP address to the blocked addresses
5. Save the settings and reset the router

4.2.4 Readability of formulations

The readability of the formulations needs to be comparable between the different attacks. Otherwise, it might lead to people thinking one attack is more dangerous simply because of its presentation. As explained in the theoretical framework, we analyzed the complexity in two different ways; the type of sentences and the Flesch-Kincaid Score. Table 16 shows which part of the formulation contains which type of sentence.

Table 16: Type of sentence for each part of the problem structure

Question	Minimal version	Addition
What is the problem	Simple	-
Why is it a problem	Simple	Complex, 1 dependent clause
What is the consequence	Simple	Simple
What is the solution	Simple	4 or 5 sentences of which 2 or 3 compound
Metaphor	-	Complex, 1 dependent clause

Each formulation has the same type of sentences in all parts except the addition for the solution. The problem here is that each solution has different steps a person needs to go through. However, since they are all a presentation of technical steps, it is expected that this does not result in a big difference when it comes to impact. If a person is uncertain about his competence, adding four or five steps should both result in a similar feeling.

Next up is the Flesch-Kincaid score for each attack and additions. The Flesch score for the basic formulations can be seen in Table 17, the score for the uncertainty based additions can be seen in Table 18 and the score for the fully combined formulations can be seen in Table 19.

Table 17: Flesch-Kincaid score for the basic formulations

Question	Blacklist	DoS	Botnet	Port Scan
What is the problem	47.3	46.605	47.3	61.325
Why is it a problem	61.24	75.5	61.24	75.5
What is the consequence	66.1	50.665	56.7	57.234
What is the solution	78.245	53.655	71.768	88.905

Table 18: Flesch-Kincaid score for the uncertainty based additions for each attack

Question	Blacklist	DoS	Botnet	Port Scan
Why is it a problem	69.785	71.582	63.017	66.401
What is the consequence	60.705	57.095	55.178	59.067
What is the solution	61.513	64.468	62.446	64.788
Metaphor	66.404	57.605	55.405	69.464

Table 19: Flesch-Kincaid score for the fully combined formulation for each attack

Attack	Flesch-Kincaid Score
Blacklist	64.035
DoS	63.289
Botnet	60.293
Port Scan	65.847

The full analysis can be found in appendix N. We can see that most formulations are rather close to each other and are mostly between 50 and 70. Remember that the average news article in a newspaper has a score of around 40. It thus seems that readability should not be a problem. There is some variation between different vulnerabilities; this is often caused by a single word with a large number of syllables. All in all, it seems that these formulations are acceptably close and readable.

4.3 How was the structure for the interview created

How the interview is structured and which questions we ask are important to get reliable results. The order in which we present the different parts of the interview can influence the participants' answer. Because of this, we decided to start with the assignment. Starting with the assignment guarantees that the user is still open-minded and uninfluenced when ranking the formulations. Self-assigning the most prevalent uncertainty is done at the end. We want the participants to understand what each uncertainty entails before asking them which they see as most prevalent for themselves.

The interviews will start with an introduction to the topic. In this introduction, we do not mention uncertainty. We tell the participants that we will ask them to rank several formulations and answer question afterwards. We then start with the first part of the interview; the assignment. We present the participant with four formulations of the same attack and ask them to rank them based on how worrisome the message is. We repeat this another three times until all four vulnerabilities have a ranking. Finally, we ask them to rank the attacks themselves. We start with this assignment so that prior questions can not influence the participants. The participants do not know yet that the formulations are related to uncertainties, which prevents any unintended nudging.

After this, we start with the semi-structured interview. We first ask for necessary background information; the age and highest finished education of the participant. Next, we ask the participant questions about their view on network security and knowledge of messages on the internet. Then, we ask the participant what their stance is on the warning messages we showed them in the assignment. After these steps, we explain the link between the uncertainties and the formulations and ask which uncertainty they would assign to themself as most prevalent. From now on, we will refer to this choice as the self-assigned uncertainty. Finally, we discuss the answers to the assignment in relation to the uncertainties. The final interview can be seen in appendix J.

4.4 How are participants found?

It is important to have participants of various ages, genders and prior knowledge. As we previously established, almost everyone comes in contact with the internet, regardless of age or experience with computers and internet networks. Next to that, IoT is growing and starting to affect more and more people. Because of these factors, we chose to look for participants that represent the Dutch population when it comes to age and education. We use demographic information released by the CBS (central bureau for statistics) to determine the ratio for education and age. While the distribution of the actual participants does not have to be an exact match, it should be taken into account when looking for participants. The distribution of age and education can be seen in table 20 and 21.

Table 20: Distribution of highest finished education in the Netherlands of inhabitants 25 years or older

Highest Education	Percentage of population
No Bachelor	56.2%
Bachelor	27.0%
Master	16.6%

Table 21: Distribution of age in the Netherlands for inhabitants between 20 and 60

age	Percentage of population
20-30	24.5%
30-40	23.5%
40-50	24.2%
50-60	27.8%

We used different approaches to find participants that fit these distributions. The first approach is to send an email to a company to ask whether they have employees that are interested in taking part in the research. Among the approached companies were an elementary school, an ICT company and a fish farm. These companies have employees with a variety of specializations and age. Next to this method, students were also approached, mainly on the TU Delft and asked whether they or someone they knew would be interested in participating. In this way, we found multiple participants with different backgrounds. Some of these participants are still a student or recently finished their study.

Multiple people in an older age group (> 60) were also approached and asked to be a part of the research. We ran into several problems with the interviews that caused them to be excluded from the final results. These problems are discussed in section 5. Once a possible participant expressed interest in participating, we sent them a short introduction about the research, which can be seen in Appendix I.

4.5 How are interviews held?

Originally the interviews were planned as a face to face interaction. There was a preference for this approach since it often makes it easier to explain yourself and allows both parties to use and pick up on nonverbal communication. Nonverbal communication is a vital component in human interaction; according to D. Phutella [78] “nonverbal communication can become a barrier or break down barriers”. The same author states that people tend to rely on nonverbal cues when there is a conflict between verbal and nonverbal messages.

Due to the emergence of COVID-19, we had to move a part of these interviews to an online format. The original setup is described first with the adaptations for an online interview following.

4.5.1 Original (offline) setup

The interviews consisted of two people, the interviewer and the interviewee. They are in a room together without any others to prevent distractions. We first briefly recap the topic of the research and ask the participant to sign the informed consent form. If they agreed with the informed consent, we asked them to sign it. We then present the interviewee with the different formulations for one attack. The formulations were printed out on separate pieces of paper so that the interviewee can move them. The order in which the formulations were presented was randomized. Once the formulations for the first attack are ranked, the interviewee will be presented with formulations for the second attack, up until the fourth. The order of the attacks is also randomized. If participants had questions about the formulations they were not answered during this part of the interview. Answering questions during this part could influence the answers of the participant, furthermore, there might also not be someone to answer questions in a home environment.

When this part is finished, a more classical interview segment follows. Instead of just questions and answers, the interviewees are asked to talk freely while the questions guide them through the conversation. While the interviewee talks, notes are made of what they say, which are analyzed later. Ten interviews were held in this way, of which one was not included in the results.

4.5.2 Online setup

The online setup is similar to the offline one, except that the different formulations can not be given to the participant on pieces of paper. Instead, we gave them a pdf file that contains the formulations for one attack in a randomized order. After the participant ranked the formulations, we give them the next pdf containing the formulations for the next attack. We used the software the participant is comfortable with, such as Skype. Twelve interviews were held in this way.

4.6 How are answers analyzed?

We code the answers and notes in ATLAS.ti and analyze the rankings in Microsoft Excel and R. In Atlas, each question gets its own code, such as 'Knowledge' or 'Protective measures'. These codes are then given sub-codes to give more insight, such as 'Protective measures: External party' to indicate another party takes care of the network security. Based on these codes, it was easy to see which types of answers there were and how many of each exact answer.

These quantities were then put in Microsoft Excel to further analyze them where necessary. The rankings, for example, were averaged per person and in its totality. We then used R to create a visualization, which provides a more straightforward understanding of the results.

5 Results

We held a total of 22 interviews. One of the interviews had results that, while interesting, could not be used for the analysis. In this interview, the participant had prior knowledge of the attacks and could not rank them since he was not worried about them. The result is a small scale qualitative test with a variety of demographic characteristics. Of the 21 interviewees, there were 13 male participants and 8 female participants. The age of participants varies between 25 to 60. 4 participants are students. 6 participants have an active involvement in the field of IT.

Outside of these participants, we approached three people of a higher age group (>60). However, several problems came up that prevented them from being included in the interviews. First of all, since the interviews had to be online, their skill with a computer can be a problem. There is the setting up of an online call, which might include new software. Secondly, there was a necessity to transfer files and switch between the call and a PDF file containing the formulations that need to be ranked. Finally, language was a barrier, especially when it came to reading technical information. One option could have been to translate the interviews to the native language (Dutch), but this was out of scope because of the complexity check on the formulations. Since English and Dutch are constructed differently, a direct translation can convey a different feeling. A translation that accurately captures the message, emotion, complexity and technicality would require an in-depth set up to ensure the translation does not influence the message within the formulation.

Tables 22 and 23 show the age and highest finished education of participants. Keep in mind that these percentages are only of the intervals included in the relevant group so that it sums up to 100%. It is visible that there is a slight bias towards younger and older people and participants with higher education. Since this research has a small scale, the participants do seem to represent a decent distribution.

Table 22: Distribution of age in the Netherlands

age	% of population	participants	percentage
20-30	24.48%	7	33.33
30-40	23.55%	4	19.05
40-50	24.21%	3	14.29
50-60	27.76%	7	33.33

Table 23: Distribution of highest finished education in the Netherlands

Highest Education	% of population	participants	percentage
No Bachelor	56.52%	8	38.10
Bachelor	27.03%	7	33.33
Master or PhD	16.45%	6	28.57

5.1 Stance on security

We first asked participants which sources they use when solving a problem with their computer or network; figure 12 shows the results of this question. From left to right is the first, second and third source participants used. Some participants had answers where they used two sources simultaneously or used different sources based on the problem. In the cases where a participant used two sources simultaneously, we added both answers to the same bar, which explains why the first bar has more answers than the number of participants. One participant, for example, stated that the second source would be either an Acquaintance or a technical service, depending on if it is a private computer or work computer. There were 21 participants for the first source, 13 participants with a second source and 6 participants for the third source. We did not take into account how or how long someone tried to solve the problem. Advanced problem-solving techniques or simply restarting the computer both add to the same bar. The purpose is not to see if what people do is correct but rather what their experience is.

It is also interesting to note that two-thirds of the participants first try to solve the problem themselves even though they might not be sure of their expertise. One participant stated that “I try it myself, which usually doesn’t work”. Participants are least likely to approach a technical service; four participants remarked that they are more likely to reach out to a help desk at work.

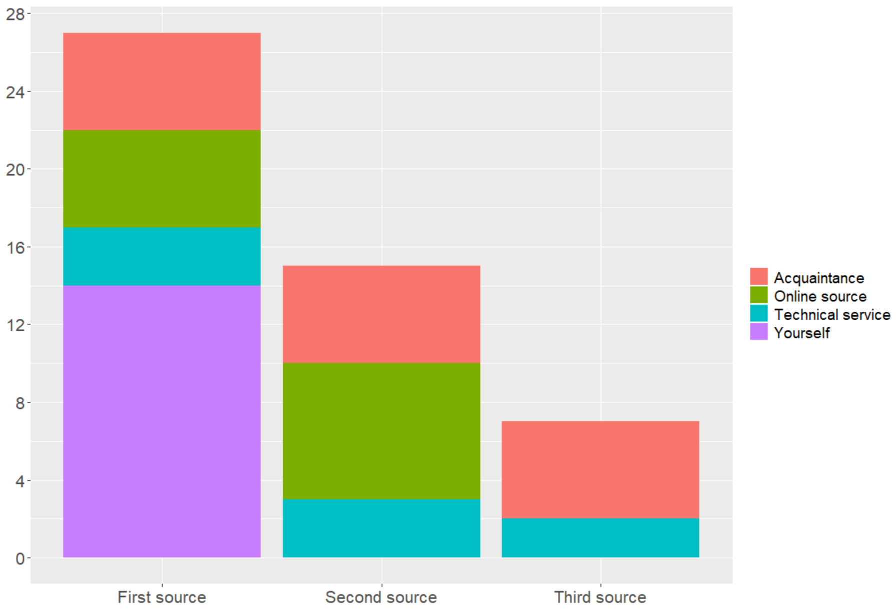


Figure 12: The type and order of sources participants consult to solve a problem.

Next, we asked the participants how important network security is for them. Of the 21 participants, 12 stated it is important, 5 stated it is very important and 4 states it is not important. This shows that most participants do think network security is important. One participant responded to the question 'How important is network security for you?' with the following answer: "Not at all. I know it should be, but I am not worried". Another participant stated that "I don't think about it too much".

When asked which protective measures participants use, there was a wide variety of answers. The most common answer is antivirus software (15 times), followed by protection from an external party (11 times), in third place, there is a firewall (10 times). Some participants use multiple different measures simultaneously, while others don't use anything at all. Three participants, for example, stated they have an Apple computer, which means they don't need to worry about protecting themselves.

Yet again, we did not take the effectiveness of the measures into account. Apple products, for example, are not entirely protected and do not protect your network itself. It does become clear that most people are concerned with the safety of their computer and not the network since most of the measures aim for protecting a single computer. Since participants think about their computer when asked about network security, it could indicate a lack of awareness that a network can be infected even if the computers are safe. IoT devices introduced a whole new playing field for attacks and are not safeguarded by protecting a computer. Next to that, it became clear that the feeling of importance is not directly related to the measures taken. Four participants stated that while they know network security is important, they do not take any or enough precautions.

The final question in this category is why users do or do not use these protective measures. It is interesting to note that there is a wide variety of reasons and that there does not seem to be a consensus between participants. The most common answers were general safety and security being built-in. One participant split the WiFi signal to create a ‘guest network’, this was done for convenience sake and not security. There was one participant that works in cybersecurity and stated that “I know about it, so I do not have an excuse not to do it (take protective measures)”. Another participant used antivirus software “so that I do not have nothing” and protects his router because there are other people on the network that might not be as careful as he is.
It is interesting to note that most people do not have a very concrete reason for using protective measures. The only concrete reasons are Data protection, Privacy and preventing infections; the others are either rather general or are taken care of by others.

Only three participants gave a reason for not using protective measures, of which two don’t use any protection. One of the participants stated: “I am pretty safe without any protective measures since I am careful”. Another participant said that virus scanners slowed down the computer without offering a lot of protection.

5.2 Technical background knowledge

The understanding of how the internet works might impact the way people look at dangers. We asked participants if they know how messages are sent over the internet. Six participants knew how messages are sent over the internet, out of these six, five participants have an IT related study. Out of the 9 other participants, one knew how messages are sent and one knew roughly how they are sent.

Since most people without an IT study do not know how messages are sent we should most likely not expect them to know how to protect themselves. When looking back at the measures people take, this could explain why most measures focus on a computer rather than a network.

We also presented several terms to the participants asked whether they know what they mean. The first term is NetFlow; only one participant roughly knew what it means; the rest hadn’t heard of it before. The other terms we showed the participants are used in NetFlow; if a participant did not know what NetFlow is, we explained it to them before we showed them any other terms. Some of these terms were straight forward, such as duration, number of packets or size. All participants either knew what they meant or could come to a correct definition by connecting the term to NetFlow. The more technical terms were a challenge. Seventeen Participants knew what an IP address is, while four roughly knew what it means. Seven participants knew what a port is, eight knew it roughly and six did not know. When asked about a protocol, 11 participants did not know, three roughly knew, and seven participants knew the term. Out of these seven participants, five of them have a background in IT.

Now that we know the background of the participants, we will look at their responses to the different messages. The following part is the more theoretical part of the interview, where the aim is to find out which uncertainties are relevant for users.

5.3 Ranking of uncertainties

In this section, we present the rankings of the uncertainties. Participants were given four formulations of the same attack and asked to rank them on severity, where 1 is the most worrisome and 4 is the least. The participants did not yet know that these formulations were related to uncertainties. At the end of the interview, we asked participants to choose which uncertainty they see as most prevalent for themselves. The full list of answers can be seen in appendix K

First, we will give a global overview by counting together the scores of all the participants, after which we will take a look at individual responses. We will order the individual responses based on the self-assigned uncertainty. We can then see whether different participants with the same self-assigned uncertainty are consistent in their answers.

Table 24: The amount of times an uncertainty was ranked on a certain place, followed by the average rank of the uncertainty

Uncertainty\Ranking	First	Second	Third	Fourth	Average Rank
Attacker intention	22	26	31	5	2.23
Consequence	33	19	25	7	2.07
Competence	11	16	15	42	3.05
Complexity	18	21	15	30	2.68

Table 24 shows us how often an uncertainty was given a certain ranking. Each ranking has 84 answers, collected by 21 participants ranking four attacks with four formulations. When looking at the individual rankings, we can see that consequence is chosen most often as the top uncertainty. Attacker intention and consequence uncertainty are rarely ranked last (5 and 7 times respectively). Competence and complexity are often chosen as last (42 and 30 times respectively). However, when looking at the average rankings, it becomes clear there is not a sizable difference. The rankings seem to indicate that there is not a one size fits all solution. We will now take a look if participants that assigned the

Table 25: The self assigned most prevalent uncertainty.

Uncertainty	Times self assigned
Attacker intention	1
Consequence	6
Competence	9
Complexity	7

same uncertainty to themselves have similar answers by grouping these participants.

Table 25 shows how often an uncertainty was self-assigned as most prevalent. Keep in mind that two participants answered with a combination of uncertainties; one participant combined complexity and competence and another participant combined consequence with competence. These results paint a different picture than table 24. Competence has a relatively low ranking but is the most self-assigned uncertainty.

In the following sections, we will look at the answers of people grouped on their uncertainty.

5.3.1 Rankings for participants that assigned attacker intention as most prevalent uncertainty

Since only a single participant self-assigned uncertainty, this section will be rather short. The rankings for this participant can be seen in table 26. We can see that attacker intention does indeed have a rather high ranking, however, consequence has an even higher ranking.

Table 26: Average ranking for the participant that self-assigned intention of the attacker as most prevalent uncertainty

Attacker intention	Consequence	Competence	Complexity
1.75	1.5	2.75	4

One of the comments was that the network should be able to fix itself in the case of a DoS attack; this caused the uncertainty about the intention to be ranked lower. The participant also stated that a “metaphor sounded very childish so it gave less pressure”, which explains the low rating for complexity. It seems that a metaphor can work counter-productive for specific participants.

Based on the answers of this participant, it is not possible to say whether or not participants with uncertainty about the intention of the attacker can be approached similarly. More responses are necessary to make any conclusions about this.

5.3.2 Rankings for participants that assigned consequence as most prevalent uncertainty

The focus of this section is the five participants that self-assigned consequence as their most prevalent uncertainty. One other participant gave a double answer, of which consequence was one of the two uncertainties. We will discuss the participant with a double answer in section 5.3.5. Figure 13 shows the results from the participants that self-assigned consequence.

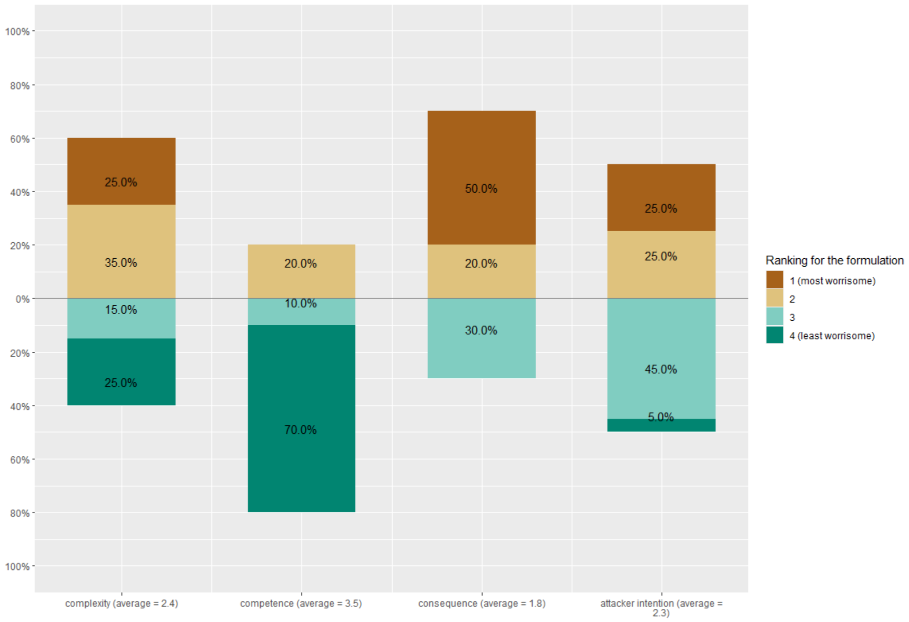


Figure 13: The ranking of attacks for five participants that self-assigned consequence uncertainty, together with the average ranking.

This figure shows the percentage of times a formulation based on an uncertainty got a certain ranking. Complexity, for example, was ranked first in 25% of the cases. The expectation was that participants would consistently rank their self-assigned uncertainty higher than the others. The image shows that 50% of the rankings indeed had consequence at the top. On the other side, 70% of the rankings had competence at the bottom. Based on these responses, it seems like participants that self-assign consequence, have similar worries. When we take a look at the average rankings, we can see that the ranking for consequence is higher (1.8) than the second highest, which is attacker intention (2.3).

These answers indicate that people who self-assign consequence uncertainty are similar and might be reached effectively in a similar way. Appendix L.1 shows a further analysis, where the participants' answers are not combined.

5.3.3 Rankings for participants that assigned competence as most prevalent uncertainty

In this section, we present the results of seven participants that self-assigned competence. Table 65 shows the average score for each uncertainty for participants that chose competence as a self-assigned most prevalent uncertainty. Seven participants answered competence and another two participants answered competence in combination with another uncertainty.

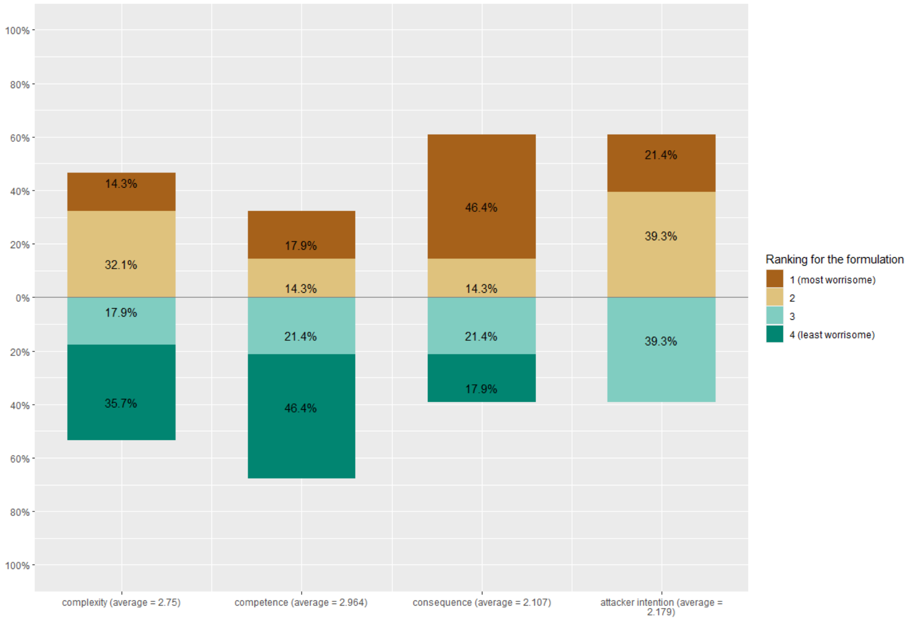


Figure 14: The ranking of attacks for seven participants that self-assigned competence uncertainty, together with the average ranking.

Figure 14 tells us a different story. The expectation is that people that self-assign competence uncertainty would give it a higher score since they need to perform multiple technical steps to solve the problem. The results, however, show us that competence has by far the lowest ranking (3.5 on average). The highest-ranked uncertainty is, once again, consequence with an average rank of 1.8.

The comments made by participants give a better understanding. Three participants stated that a list of steps offers peace of mind. Even if they are worried about their competence and might be unable to perform the necessary steps. These participants are less worried when the message includes a solution, which could indicate a high feeling of response efficacy. It is possible that the lack of confidence in one's competence will result in a higher willingness to accept a solution that is presented by another party. Whether a lower competence results in more willingness to accept a solution could be an interesting topic for future research.

Based on these answers, it does not seem feasible to reach the users in this group in a similar way. Deeper analysis with a look at each unique participant can be found in appendix L.2

5.3.4 Rankings for participants that assigned complexity as most prevalent uncertainty

Finally, there are the six participants that self-assigned complexity. One participant assigned complexity combined with another uncertainty; we will discuss this participant in section 5.3.5.

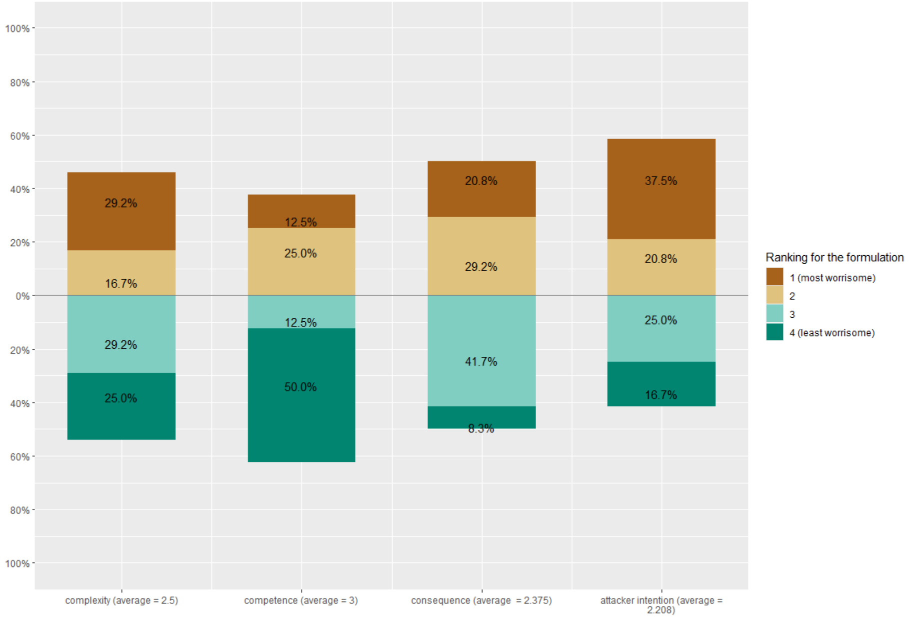


Figure 15:

The ranking of attacks for six participants that self-assigned complexity uncertainty, together with the average ranking.

Once again, we would expect the ranking for complexity to be high for these participants. When we look at figure 15, we can see that this is not the case. All rankings are rather close to each other. The lowest average rank is 3, while the highest average ranking is 2.208. Complexity itself has the second-lowest ranking, but is close to all other uncertainties.

Comments can once again give some more insight into how this ranking came to be. One participant states that complexity is indeed a problem, but the system should not assume it is. The inclusion of a metaphor without the participant asking for more information felt "pedantic" and lead to the participant giving it a low ranking. Another participant stated that a metaphor makes the problem more real and serious. The difference in the answers shows that different people have different preferences. A metaphor can thus be helpful for one while it works counter-productive for the other. These results could imply that a different approach is necessary to make the problem less complex. The results also show that these participants can not be grouped with the current approach.

Another possible problem is that a metaphor can both increase and decrease the feeling of complexity. If a person does not understand the system, it can cause uncertainty. The addition of a metaphor can make the message seem more dangerous and thus increase the uncertainty. However, if the metaphor makes the problem seem less dangerous, it can also lower the feeling of uncertainty. Appendix L.3 shows a deeper analysis with individual responses.

5.3.5 Rankings for self-assigned combination

There were two participants that self-assigned a combination of two uncertainties. In this section, we will analyze the answers of these two participants. The first participant assigned a combination of complexity and competence. The average rankings of this participant can be seen in table 27.

Table 27: Average ranking of the participant with self-assigned complexity and competence uncertainty.

Attacker intention	Consequence	Competence	Complexity
2.25	2	3.75	2

Once again, it seems that the list of steps decreases the feeling of competence uncertainty. In this case, it makes sense that competence has a low ranking even though it one of the self-assigned uncertainties. The self-assigned complexity is not reflected in the rankings. The three other uncertainties are too close to each other to make any conclusions out of this.

We are now left with one more participant that self-assigned two uncertainties, which were consequence and competence. Table 28 shows the average rankings of this participant.

Table 28: Average ranking of the participant with self-assigned consequence and competence uncertainty

Attacker intention	Consequence	Competence	Complexity
2.75	2	1.25	4

We can see that competence is again part of the combined uncertainty. This time, however, the other one is consequence uncertainty instead of complexity uncertainty. This time the rankings are pretty in line with the self-assigned uncertainty; competence and consequence both have a high ranking. Complexity was consistently ranked lowest, which is explained by the comment that "Complexity gives some ease because it explains more" and that is not an "unknown unknown". The comments of this participant make it clear that consequence is something that motivates the participant to take action. Competence seems like it could give some doubt or negative feeling. Even though competence has the highest ranking for this participant, it might be more beneficial to focus on the consequence.

Now that we have seen the results of each participant, it is possible to compare the average ranks between groups to see if there is a significant difference.

5.4 Average rank per self-assigned uncertainty

Table 29 shows the average for each group of self-assigned uncertainties, together with the combined average. Keep in mind that there was only one participant that answered attacker intention.

Table 29: Average ranking per self-assigned uncertainty and the total average

Assigned \Rank	Attacker intention	Consequence	Competence	Complexity
Attacker intention	1.75	**1.5**	2.75	4
Consequence	2.3	**1.8**	3.5	2.4
Competence	2.18	**2.11**	2.96	2.75
Complexity	**2.21**	2.38	3	2.5
Total Average	2.23	**2.07**	3.05	2.68

Here it is visible that the difference in ranking is not very large between groups. Attacker intention uncertainty is difficult to compare since we can not compare a single participant to a group of participants. When we look at the other groups we see that the average rank for attacker intention is very close to the rankings per group. The ranking for consequence does have a slightly higher ranking for people that assigned this uncertainty to themselves. Competence has a slightly lower ranking for people that self-assigned consequence but has a rather low ranking in general. The rankings for complexity are also close to each other and the average.

Next to just comparing the users based on their ranking, an attempt was made to perform a statistical test, such as the chi-square test. However, the group sizes were too different and the total sample size too small to obtain valuable information from a statistical test. These results seem to indicate that we can not effectively group participants with similar self-assigned uncertainties.

5.5 Link between uncertainties and attacks

In section 4.2.1 a hypothesis was made that links uncertainties and attacks on a network. As a brief reminder they will be restated here.

- Blacklist - Attacker intention
- DoS - Complexity
- Botnet - Consequence
- Port Scan - Competence

A first idea of whether these hypothesized links could be interesting for further research is by looking at the self-assigned uncertainty and the average ranking of the corresponding attack. Table 30 shows the average ranking of the attacks.

Table 30: Average rank of each attack.

Attack	Average Rank
Blacklist	2.429
DoS	3.143
Botnet	1.762
Port Scan	2.571

We will use the first participant as an example, this participant self-assigned competence uncertainty. The corresponding attack is port scan, which was ranked the highest for this participant. The average ranking for Port Scan is 2.571. The difference is thus $2.571 - 1 = 1.571$. Table 31 shows the self-assigned uncertainty, together with the ranking of the corresponding attack, for each participant. The participants with a double answer are left out because they could not give formulations the same rank.

Table 31: Ranking of the attack that is hypothesized to be linked to an uncertainty. Together with the difference between the rank and the average rank of the attack.

Participant	**Self-assigned**	**Rank of corresponding attack**	**difference with average rank**
1	Competence	1	1.571
2	Consequence	1	0.761
3	Complexity	2	1.142
4	Complexity	2	1.142
5	Complexity	4	-0.857
6	Consequence	1	0.761
7	Competence	2	0.571
9	Consequence	1	-0.238
10	Complexity	1	2.142
11	Competence	2	0.571
12	Attacker intention	2	0.428
14	Competence	1	1.571
15	consequence	1	0.761
16	complexity	4	-0.857
17	Competence	2	0.571
18	Competence	2	0.571
19	complexity	4	-0.857
20	consequence	2	-0.238
21	Competence	4	-1.428
Average		1.938	0.426

When looking at this table, we can see that the hypothesized link has a lower score than the average six times. Thirteen times the ranking of the corresponding attack has a higher ranking than the average. The average rank of the attack that corresponds to the self-assigned uncertainty is 0.426 higher than the average rank of those attacks. These results could indicate that personal uncertainties are indeed linked to a difference in perception of the severity of an attack. The link between uncertainties and type of attack would be an interesting point for further research.

5.6 Reaction to warning messages

The last category of results is related to the way users look at the warning messages. The first question of this category that we asked participants is whether or not they'd want to receive warning messages. Eighteen out of twenty-one participants answered yes, with only three answering no. Two out of these three participants state that they do not know what to do with the message, so they would rather not get them. One of them even stated that “It would make me panic”. The other participant that answered no would be worried he could not trust the warning message because that in itself might be an attack. These answers show that it is important to make a system that feels trustworthy to the user.

After we asked the participants whether they want to receive messages, we asked them why they want to receive them or not. For this question, participants could also state reasons why they would not want to receive them even if they would want to receive them and vice versa. The main reasons participants do want to receive the messages can be attributed to general security and to act upon the warning. Four participants stated that curiosity plays a role in wanting to receive the messages.

When looking at the reason people do not want to receive messages, there are more concrete reasons. The most common reason was that the participant would not know whether or not to trust the warning message since it could be an attack itself. One participant stated that a high frequency of warnings would lead to annoyance and prevent the participant from wanting any more notifications. Other reasons were a participant not knowing what to do so the warnings would be useless. Another participant stated that warnings such as these only lead to a feeling of uneasiness.

When looking at these answers, it seems like most people don't have a strong intrinsic reason to want to receive messages. It is possible that receiving messages is considered the correct thing to do; this is supported by the idea that people want to receive warnings without understanding them. Curiosity is a more intrinsic reason; this was the case for four of the participants. The reasons for not wanting to receive the messages have a higher intrinsic focus; Trust, competence, annoyance or unease are all personal experiences that influence how you feel about something. Two participants also said that they would rather not receive a warning because it indicates that there is a problem. Although the participants made the statements jokingly, it does show that such messages can give a negative feeling. These remarks show that it is essential to construct a tool that people can trust without inducing negative feelings.

We then asked participants whether they are likely to act immediately when they receive a warning. Fourteen participants said they would, with only two saying they would most likely not. Five of the participants said it depended on the situation; one participant stated that an acquaintance would be necessary, but that the participant would not want to bother someone else if it is late. Another participant stated that it depends on whether or not he knows what to do. The two participants that answered no gave this answer because they would not know what to do. One of them said: "It only makes me uncomfortable since I have to do something I don't understand", while the other stated: "I want a phone number for someone that can fix it remotely".

5.7 Desired structure

The final question that we asked the participants is what they would like a warning message to look like. Participants could freely speak their mind and were not steered in a direction. The answers are thus the parts that participants came up with themselves.

Eleven of the participants stated they would like to know what is causing the warning. Of these ten participants, two stated that more information about the attack would be appreciated, one stated that the severity of the attack is important to include. Another participant, outside of the eleven that want to know what is happening, stated that the type of attack is not important.

Only four participants explicitly stated they would want to know the consequence. Another participant remarked that the consequence can be “scary”.

The most common answer given by thirteen participants is that the message should include a solution. One of these thirteen participants stated he would like the solution to contain more information, such as big ISPs in the Netherlands, so it would be easier for users to find which one is theirs.

When it comes to technical information, the participants are rather divided. Six participants said they would like to see no technical information or as little as possible. Eight participants stated they would like to see more technical information; out of these eight, four have a background in IT. One of the participants that said technical information is not necessary has a background in IT and stated that the type of attack is enough information. The difference between these participants shows that people with a similar background can have different preferences. Two participants that want technical information do not want it for themselves but for someone that helps them solve the problem. One participant stated that the message should contain technical information, but only after pressing a “more information” button.

Four participants remarked that a metaphor is unnecessary. One of these participants stated that a metaphor could be vague or bothersome to people with technical knowledge. Participants also had a different preference for the tone of the message. Four participants want the warnings to be professional, to the point and, serious, while two participants want the problem and solution explained in layman’s terms. Two participants added they would like the warning to be in Dutch.

A rather general preference, given by eight participants, is the possibility of external help. There were different ways participants would like external help; the most common was a phone number for technical assistance. Two participants would prefer the possibility to open a chat for help. One participant prefers a contact person. These participants are still rather focused on solving the problem themselves with some outside help. Two other participants would like a party to remotely solve the problem for them, while one participant would like a technical service to solve it without getting a warning. Interestingly enough, one participant stated they would not want a phone number while another one would not want an email address because they would not trust the people on the other side.

There were other interesting remarks. One participant stated that, whenever possible, a 1-click solution would be preferred. Another participant stated that receiving a warning message is always worrying regardless of the content. Finally, one participant wants a warning to explicitly state the urgency of the problem.

In this regard, there also does not seem to be a solution that suits all participants. Since thirteen participants want the message to include a solution, and no participants are explicitly against it, it seems valuable to include the steps to solve the vulnerability. The same goes for the observation of what is happening; one participant stated it does not matter but did not claim it should be left out. No participants stated they wanted a metaphor to be included while four were against it, so this might be best to exclude from the message. External help also seems like a good inclusion, however, there is no consensus on the type of external help. A possibility would be to let people customize their message.

Now that we presented all the results, we can draw a conclusion and answer the research questions in the following chapter.

6 Conclusion

This section will answer the research questions and the main research question of this paper.

RQ1: Which network vulnerabilities will be taken into account?
Based on part I of this research, there are five different vulnerabilities: DoS, botnet, port scan, blacklist and anomaly-spam. As discussed in the previous part, it is not completely clear whether or not a separate system should look for spam or that users should be informed or educated on how to recognize spam. For that reason, the four vulnerabilities that we use in this research are DoS, botnet, port scan and blacklist.

RQ2: How can a message be used to influence the motivation and behaviour of users so that they take action?
In section 2, we discussed several theories regarding behaviour, motivation, uncertainties and information. Based on Wendel's CREATE action funnel and Fogg's behaviour model, we can see that a message sent by a system can act as a prompt for a user to start taking action. However, this also depends on the motivation and attitude of the user.

Motivation and attitude are factors that are factors within a person; however, the presentation of a message can influence these factors. Section 3 shows the requirements for the message to be as motivating as possible. The main conclusions are:

- Threat should be clear but not too intimidating. We want to improve threat appraisal without decreasing self-efficacy.
- Self-efficacy should be stimulated by supporting the user in the message. The formulation of the warning message should give the user the feeling they can solve the problem.
- A user should be able to understand and use the message to solve the problem; this stimulates competence, as seen in the self-determination theory. These factors can lead to a more internalized motivation to solve the problem.
- The Technology acceptance model shows us that a message should be perceived as useful and easy to use.
- A way to make the message more useful and practical is to make sure each message contains all relevant information. An option is using a fixed structure, such as a problem structure.
- We should prevent triggering uncertainties, as they can prevent users from carrying out the desired behaviour

RQ3: How do users respond to a different presentation of the same vulnerability when there is a focus on a different uncertainty?
Section 2 argues that uncertainties can differ per person and uses Wendel's CREATE action funnel and Fogg's behaviour model to show that these uncertainties can prevent users from taking action. In most cases, the participants were able to rank the different formulations without problems; this shows that each formulation does raise a different amount of concern. Only a single participant answered that the difference in formulation did not matter and that they all invoked the same feeling. Of course, since this was a rather small scale research in the Netherlands, it is not known if these results represent all users. There might be regional or national differences, which would have to be researched in a quantitative follow-up research.

Based on table 24, we can see that, on average, including a list of steps to solve the problem lets users worry less. Including worse consequences resulted in users being more worried about the warning. One of the challenging factors is that the hypothesis was that a list of technical steps would induce a higher feeling of uncertainty about one's competence. In reality, the inclusion of a list of steps often gives users peace of mind, even if they are unable to perform the tasks described. When we take a look at table 25, we do see that competence is the uncertainty most people are worried about. Only a single participant was most worried about other actors but the average rank of this uncertainty is the second highest.

RQ4: Based on RQ2 and RQ3, can users be grouped based on their uncertainties?.
From the collected answers, it becomes clear that users cannot be grouped just on the self-assigned most prevalent uncertainty. A participant that has a background in IT states that a metaphor helps make it more serious, while a participant that does not understand computers thinks a metaphor is pedantic and annoying, even though it would help with understanding the attack. The results show that warning messages should not be constructed based on theory alone but need a practical and personal approach. We can conclude that uncertainties can not be used to group users. We also see that uncertainties can play a part in constructing warning messages. A more personalized approach is necessary.

Based on section 5.7, some generalisations can be made, such as the inclusion of a solution. However, even in participants that would like a similar feature, such as contact information for further help, there is a different preference of the exact implementations. Some participants would prefer a phone number, while others would distrust a phone number.

Main question: In what way can technical vulnerabilities in the context of IoT be presented to users to motivate them to take action when taking into account personal uncertainties?
From the previous answers, we can see that there is no one clear cut solution. However, we did get valuable information from both the theoretical framework and the qualitative research.

Users did recognize the uncertainties in the context of network security and were able to relate them to attacks. This implies that uncertainties are indeed a possibility to create a more personalized approach. The challenge is that not every person with similar uncertainties wants to be approached in the same way.
We did see that competence is a common worry and that including a list of steps to solve the problem can give users peace of mind, even if they don't understand the steps. The main uncertainty that seems to worry people in the formulation is the consequence, but it was not clear whether this would prevent people from taking action. From the comments of participants, it became clear that more severe consequences would stimulate them more to take action. This could imply that the balance between self-efficacy and threat appraisal should lean more towards threat appraisal. If we take a look at complexity, a metaphor is not always valuable for participants. Keeping the message relatively simple and to the point seemed to work for all participants, which implies that avoiding technical information and jargon is, in general, beneficial.

When it comes to the exact formulation, it seems that choice is an important aspect for most participants. Some information can be useful but should not be shown without the user choosing to opt-in. It seems best to make the formulation customizable to make sure participants can refine the warning to their liking. Multiple participants, for example, would like a way to request external help. Which type of external help varies per participant; one would like a phone number, the other distrusts phone numbers. In this case, it would also help if a user has a basic version and can opt-in to extra information.

After seeing the formulations, only 2 participants stated they would not act on a warning message, which shows that the general formulation of the message is effective and that participants have the intention to act. Based on the results we can conclude that we should send these type of messages to users. Whether their intention to act turns into behaviour can not be concluded from this research.

All in all, this research gives valuable pointers to a successful formulation based on the theoretical framework and interviews. It also shows that uncertainties are a possible way to reach a more personalized approach.

7 Discussion

This chapter will give an overview of a variety of topics. How could the qualitative research have been improved? What are the takeaways for a possible quantitative follow-up? What do the results imply and which results might have been different than expected?

7.1 Findings in this research

The main research question of this paper is *In what way can technical vulnerabilities in the context of IoT be presented to users to motivate them to take action when taking into account personal uncertainties.* We chose four different vulnerabilities from part I of this research: DoS, botnet, port scan and blacklist. Based on the literature, We created several requirements to improve the effectiveness of a warning message and stimulate the user to take action. We performed a small-scale qualitative research, in the form of interviews with 22 participants. Based on the interviews we determined how users respond to different formulations of the same vulnerability. Finally, we analyzed the results to determine if we can group users based on their uncertainties.

Even though there are similarities between users, the results indicate that a general grouping based on uncertainties is not feasible. When it comes to the content of the message there was also not one general preference. Some parts of the warning message seem beneficial for most participants, such as a list of steps to solve the problem. The results indicate a link between a user's uncertainty and the perceived severity of an attack. Choice also seems to be a vital factor, which implies that a user should have a personalizable profile.

7.2 Consequences of these findings

Even though we didn't find a way to group people or a general way to formulate the warning message, there are some takeaways. Each participant is different and has their preferences when it comes to acting on a message. The results indicate that we need to look at individuality, outside of looking at grouping users.

Two responses were unexpected and unique. One participant stated that a system showing a metaphor without the participant asking for it felt 'pedantic'. The participant knew that the information was too difficult, but the system shouldn't assume this. A response like this strongly suggests that some people might want to customize the message themselves. One problem with this is that a user might not know which information is useful and helpful.
Another participant said that a metaphor would be useful if a person explains it, but a computer should stay in its lane and present to the point and factual information. We can thus see that not every user would like a personal, friendly or concerned message from an automated system.

Multiple participants stated they wouldn't know if they can trust a message that tells them there is a problem. Since recent attacks try imitating warning messages, the trust in real warning messages may be lower. To make sure that users trust the message, the system should be designed in such a way that a user can trust its messages.

Looking at the rankings of the participants, they are far less convincing than expected. People with similar worries still have a different preference for the formulation of warning messages. The expectation was that people that worry about a certain topic would respond more strongly to this topic. While this holds for some participants, the opposite was true for others.

Another unexpected result is that a list of steps to solve the problem seems to generally give more peace of mind. Even if participants were unable to understand or perform the steps, they still felt more at ease. The lack of a concrete solution might give them the feeling that they are left to solve it themselves. When we look at the protection motivation model, it could imply that a lack of know-how increases the response efficacy. However, this is something that would need more research.

7.3 Limitations and possible improvements

This section describes the limitations and possible improvements of this thesis. There are always different ways to perform this type of research; different theories can be used, a different interview method can be used, answers can be analyzed differently, etc. Since the theoretical framework serves as the basis for this thesis, we will also start by reflecting on the framework.

7.3.1 Theoretical framework

Based on the theoretical framework, we made multiple assumptions and choices. In this section, we will evaluate these assumptions and choices. Reflecting on our theoretical framework opens up possibilities for further discussion and research.

As seen in the elaboration likelihood model, we can not reach every person using the central route. Some people need a peripheral route. Next to that, this research shows that there is not a clear central route. Even though most participants want to receive warning messages and have the intention to act on them, the preferred formulations vary. When we look at peripheral routes, the options are limitless. We could, for example, try to create a flashy warning message that demands attention instead of a simple warning.

In the Self-determination theory, the focus was on internalizing motivation based on the three basic psychosocial needs. However, this might also not work for every user. Some users might not care about their network security enough for a message to be enough to increase their motivation. We could test different methods that focus on a more external form of motivation.

Self-efficacy is a vital aspect since users that feel like they are not capable are less likely to take action. Based on the Protection motivation theory, we tried to find a balance between severity and self-efficacy. In this research, we presented the consequences and dangers to the user, but they were not made into a spectacle or framed to scare the user. The balance between these two factors might not be in the middle and the results seem to indicate that increasing the perceived severity and vulnerability will prove more important than preventing the self-efficacy from being lowered.

When we look back at Wendell's CREATE action funnel, we tried to let people go through the entire funnel while preventing uncertainties or doubts from disturbing this process. It could prove feasible to find ways in to cheat these steps. By changing the formulation, we can influence the initial reaction and evaluation. The warning can also create a higher urgency, while this will lower the feeling of autonomy, and thus create a more external motivation, it might be more effective.

7.3.2 Literature for formulations

We chose to use a problem structure to present the users with a problem and tell them how to solve it. Some participants had difficulty fully understanding the different parts in the message meant and not every user has a similar response to the same message. A similar experiment could be held using a measure structure. Using a measure structure could shift the formulations from a warning to a more pressing message that tells users to take action.

We used two different measures for the text readability: The Flesch-Kincaid score and the type of sentences. However, we did not take a look at the difficulty of individual words. While longer words can make a sentence more complex to read, it also depends on which longer word we use. The different formulations were similar in readability, but we could try making them more or less difficult to read.

7.3.3 Methodology

The methodology is both an easy topic to look back on but at the same time a very challenging one. Since there are numerous different ways to approach a research like this, it is easy to see various possibilities. It is a difficult to judge whether these possibilities would have been a better.

We created the assignment in the interview based on literature. An alternative approach would be to construct these formulations together with a different set of participants. Creating the formulations with other people, or going through different versions of the formulations, would give better a priori knowledge of how people look at such formulations. The results show multiple difficulties that could have been solved by involving others before holding the interviews. The involvement of other people would expand the scope of the project and require an addiitional phase in the research.

The participants do represent the distribution in age and education rather well. However, there are different factors, which we did not take into account, such as their cultural background. We can use numerous factors when looking at a representation of a population. Since this research is small-scale, it is not possible to include every factor. In a quantitative follow-up study, additional factors can be taken into account, to make sure the representation of participants is as accurate as possible.

Due to Covid-19, interviews were changed from a face-to-face experience to an online version. There was not a clear difference between answers between the online or offline participants. The change could still have resulted in a different experience. In an online experience, it is more difficult to notice the body language from the participant, which can cause less depth in the interview. In an ideal scenario, all the interviews would be held in the same setup, preferably face-to-face.

We chose to focus on the uncertainties in the analysis since this is the focus of this research. There are, however, multiple perspectives we can take when looking at the results. A different analysis of the results might lead to new insights. Participants were now grouped based on their self-assigned most prevalent uncertainty. Instead of using the uncertainties to group participants, we could have also used any other information, such as age, their stance on security, or their background knowledge. Since the focus of this thesis are the uncertainties, we also shaped the questions around this. In a follow-up study, different factors can be taken as the main perspective to look at the impact.

7.3.4 Results of the interviews

In this research, there was a big focus on uncertainties. The reason for this focus is that uncertainties can prevent users from taking action. However, as discussed before, this does not always seem to be the case. For some participants, triggering an uncertainty can instead stimulate them to take action. One of the participants stated that a metaphor “makes the problem seem more real” and that would increase their tendency to take action. In this case, we could utilize the uncertainty to increase the feeling of vulnerability and severity, which increases the protection motivation.

Next to the choices we made based on literature, we made several assumptions. Some of the assumptions we made proved to be false. For example, the assumption that technical steps would increase the feeling of competence uncertainty proved largely untrue. Instead, it often reduces the feeling of uncertainty, even if the person is unable to perform the steps. A possible improvement for this is to involve users in creating the formulations. These were now made based on theory and assumptions. By working together with a different set of participants to create the formulations, we do not have to make these assumptions.

In the version that we used, some words could trigger negative feelings for participants, especially when they had less technical knowledge. Words like 'criminal records' or 'fraud' could influence their feeling of danger without being part of the intended uncertainty.

Some participants had difficulty with the technical information in their non-native language. For these participants, it could be better to localize the formulations to their native language. Localizing the formulations would need further research to ensure the different languages offer the same type of message. If one is more complex to read than the other, it could lead to a difference in results.

In the end, we had results based on 22 interviews. The results show that people can not easily be grouped on their uncertainties. Every person can have a different perspective, even if they worry about similar topics. Based on these results, it does not seem as though more interviews would have given a better result. The expectation is also that there is not a single solution to the problem and that a personalized method is necessary.

7.4 Contributions

This research is not the only research that looks at presenting warning messages to users. Harbach et al. [79], for example, focused on the readability of a warning message. A difference here is that they purely focused on readability without linking it to personal uncertainties.

Maimon et al. [80] looked at system trespassing and how warning messages affected this. They found that users largely ignore warning messages. In comparison to this research, it is then unexpected that most participants stated they would take action when they are presented with a warning message. Their intention to act could be related to their type of motivation, or it could seem like the 'right answer'. Their answers indicate they have the intention to take action. However, as discussed in the theoretical framework, this does not always lead to actual behaviour.

One of the biggest contributions of this research is the focus on a combination between warning messages and personal uncertainties, next to a focus on personal preferences and understanding of the user. While the personal focus and small scale of this research prevent finding clear cut results, it takes the first step towards a more personal approach.

This research also provides an extended theoretic framework in which motivation and behaviour are both taken into account. We link these aspects to uncertainties and the structuring of a warning message. The framework presents multiple angles for further research.

The results of this research do not present a clear and robust conclusion. Social sciences often aim to discover rather than to confirm. The purpose of this research is to take the first step into including uncertainties in research about warning messages. While we found out that uncertainties can not easily be used to group users, we did see they have an impact on them. Our findings allow for further research and exploration into the topic.

7.5 Reliability, validity and ethics

The interviews that we held with the participants are not replicable with the same participants. The semi-structured interview allowed the participants to ask questions themselves. Because the participants could ask questions, their knowledge of networks and attacks has changed during the interview. Their new knowledge might influence their ranking in the assignment and answers in the interview.

All the data was collected and analyzed by a single person this means that there are no inter-rater reliability issues. It does mean that the interpretation of results is prone to be subjective. A peer-review would thus improve the validity of the obtained results.

All participants participated voluntarily and were not compensated. Any personal information is omitted or put into broader categories. The names of participants are only used for the informed consent, which will be stored by the TU Delft.

7.6 Further research

This research gives some handles to start further research. Based on our theoretical framework and results, it seems that perceived severity plays a larger role than self-efficacy. This would be an interesting topic to research in the context of warning messages.

Some of the conclusions in this thesis could also be used for a follow-up research, such as including a list of steps. This would allow for a quantitative research with a focus on a single topic. These results could then confirm or dispute the findings in this research.

The results also indicate that a person's uncertainty influences how they perceive the severity of an attack. Further research into this topic would be an interesting direction to explore. A person's uncertainty is not enough to put them in a generalized group, but it could allow for tweaks aimed at users with this uncertainty.

Since we can not group users based on uncertainties, it could be worthwhile to research more customizable approaches. We will discuss two possibilities here.

A possibility is to give people the ability to customize their messages, either on startup or in a separate menu. The problem with this approach is that people might not know what works for them. If a person does not have in-depth knowledge, how would they know which technical information is useful to them? Even though this approach will not be enough on itself, a separate option to customize which information to present is a necessity when the needs of a user change over time. Another possibility could be to first have a small questionnaire on the first startup, this questionnaire would serve as a way to get to know the preference of the user, which is then used to construct a correct message. However, the questions that should be used for this questionnaire would need a separate research project.

7.7 Reflection

Since this section will be about personal experiences and reflection, it is be written in the 'I'/'me' form. I started this thesis at 'De Nederlandse Organisatie voor toegepast-natuurwetenschappelijk onderzoek' (TNO) and worked here for about a year. Since I have not worked at a big company like this, it gave me a new perspective. In my experience, the structure at the department I was part of was rather flat. I could always walk up to someone and asked them about their viewpoint. Conversations with colleagues allowed me to be more critical and open-minded about my work.

I had two main supervisors at the TU Delft. For most of the time, I had regular meetings with them. Their expertise in the field of my research allowed them to give me critical questions and feedback, which was very helpful. For science communication, I also had a second supervisor. The second supervisor was there for important meetings and gave a different perspective, which caused me to stay open-minded.

This project was not one without its challenges, both personal and practical. I did not start with a pre-established framework of what the thesis would look like. The idea was to research security around IoT. I started this research by exploring literature and research possibilities. I decided to focus on anomaly detection in NetFlow data, since there was not enough data to fully focus on IoT. For the science communication part, the original idea was to create a campaign for awareness around IoT devices. Rather late in the project, I decided to change from a campaign to the current approach. This approach would cost more time but would allow for a practical result.

The emergence of Covid-19 also complicated some aspects of the process. Most of the planned interviews were suddenly cancelled. Companies were busy figuring out how to deal with people working from home. I had to look for new participants and swap to an offline version for the remaining interviews. On a personal level, working from home caused me to turn more into myself. I started talking to others less and sought less contact with my supervisors. This caused me to get stuck in my line of thinking and lose motivation. It took me some time before I realized what I was doing; once I did, I worked to get out of this dip and remotivate myself. I started by contacting my supervisors and talking to them about my situation. After some initial hiccups, I was able to enjoy my research again.

When looking back at my research, I realize that I lost track of the scope. For computer science, there were multiple problems with WEKA and the data that caused me to spend a significant time creating workarounds. Although I learned from this and enjoyed it, it did take time. For science communication, the change from a campaign to the current approach greatly increased the amount of necessary work. Although it increased the scope, it did result in a research that I enjoyed more and results that are, in my opinion, more valuable. Even though it might have taken much more time than originally planned, I wouldn't want to have missed it.

Finally, I would like to reflect on the combination between computer science and science communication. To me, this combination is incredibly valuable, as it lets to take an extra step in research. The combination allows me to create a bridge between experts in different fields or between experts and end-users. If my only goal was to create an anomaly-detection system, I could also get stuck at chasing the best performance. Instead, the combination with science communication provided me wwith the opportunity to have a broader perspective. I personally believe that new, creative solutions are necessary for the future. To achieve that goal, people need to work together and broaden their perspective.

Science Communication References

[48] Ronald W Rogers. “A protection motivation theory of fear appeals and attitude change1”. In: *The journal of psychology* 91.1 (1975), pp. 93–114.

[49] Ronald W Rogers and Steven Prentice-Dunn. “Protection motivation theory.” In: (1997).

[50] Richard M Ryan and Edward L Deci. “Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being.” In: *American psychologist* 55.1 (2000), p. 68.

[51] Albert Bandura. “Self-efficacy: toward a unifying theory of behavioral change.” In: *Psychological review* 84.2 (1977), p. 191.

[52] Albert Bandura. “Self-efficacy mechanism in human agency.” In: *American psychologist* 37.2 (1982), p. 122.

[53] Fred D Davis, Richard P Bagozzi, and Paul R Warshaw. “User acceptance of computer technology: a comparison of two theoretical models”. In: *Management science* 35.8 (1989), pp. 982–1003.

[54] Martin Ed Fishbein. “Readings in attitude theory and measurement.” In: (1967).

[55] Viswanath Venkatesh and Hillol Bala. “Technology acceptance model 3 and a research agenda on interventions”. In: *Decision sciences* 39.2 (2008), pp. 273–315.

[56] Richard E Petty and John T Cacioppo. *Elaboration likelihood model.* 1986.

[57] BJ Fogg. *Fogg Behavior Model.* URL: `https://www.behaviormodel.org/`.

[58] Sophie C Boerman, Sanne Kruikemeier, and Frederik J Zuiderveen Borgesius. “Exploring motivations for online privacy protection behavior: Insights from panel data”. In: *Communication Research* (2018).

[59] Stephen Wendel. *Designing for behavior change: Applying psychology and behavioral economics.* " OŔeilly Media, Inc.", 2013.

[60] David Bakker et al. “Mental health smartphone apps: review and evidence-based recommendations for future developments”. In: *JMIR mental health* 3.1 (2016), e7.

[61] Iis P Tussyadiah. “Technology and behavioral design in tourism”. In: *Design Science in Tourism.* Springer, 2017, pp. 173–191.

[62] Harri Jalonen. “The uncertainty of innovation: a systematic review of the literature”. In: *Journal of Management Research* 4.1 (2012), p. 1.

[63] Andreas Kappes et al. “Uncertainty about the impact of social decisions increases prosocial behaviour”. In: *Nature human behaviour* 2.8 (2018), p. 573.

[64] Michaël Franciscus Steehouder et al. *Leren communiceren.* Noordhoff Uitgevers, 2016.

[65] Carel Jansen. “Taal-en communicatieonderwijs in beweging”. In: *Internationale neerlandistiek* 54.2 (2016), pp. 137–146.

[66] Ton De Jong and Monica GM Ferguson-Hessler. “Types and qualities of knowledge”. In: *Educational psychologist* 31.2 (1996), pp. 105–113.

[67] Joan Josep Solaz-Portolés and V Sanjosé. “Types of knowledge and their relations to problem solving in science: directions for practice”. In: *Sísifo. Educational Sciences Journal* 6 (2008), pp. 105–112.

[68] Nicole Ummelen. “Procedural and declarative information: a closer examination of the distinction”. In: *Steehouder. M., Jansen, C., van der Poort, P., & Verheijen, R.(Eds.), The quality of technical documentation* (1994), pp. 115–131.

[69] Michael T Ullman. “The declarative/procedural model”. In: *Theories in second language acquisition: An introduction* (2015), pp. 135–158.

[70] Frank J van Schalkwijk et al. “The effect of daytime napping and full-night sleep on the consolidation of declarative and procedural information”. In: *Journal of sleep research* 28.1 (2019), e12649.

[71] Rodnry Huddleston and Geqffrry Pullum. “The Cambridge grammar of the English language”. In: *Zeitschrift für Anglistik und Amerikanistik* 53.2 (2005), pp. 193–194.

[72] Rudolph Flesch. “A new readability yardstick.” In: *Journal of applied psychology* 32.3 (1948), p. 221.

[73] Ilias Flaounas et al. “Research methods in the age of digital journalism: Massive-scale automated analysis of news-content—topics, style and gender”. In: *Digital journalism* 1.1 (2013), pp. 102–116.

[74] Alberto Martín-Martín et al. “Google Scholar, Web of Science, and Scopus: A systematic comparison of citations in 252 subject categories”. In: *Journal of informetrics* 12.4 (2018), pp. 1160–1177.

[75] Giovanna Magnani and Antonella Zucchella. “Uncertainty in entrepreneurship and management studies: a systematic literature review”. In: *International Journal of Business and Management* 13.3 (2018), pp. 98–133.

[76] Leanne K Knobloch and Denise Haunani Solomon. “Relational uncertainty and relational information processing: Questions without answers?” In: *Communication Research* 32.3 (2005), pp. 349–388.

[77] Leanne K Knobloch and Kristen L Satterlee. “Relational uncertainty: Theory and application.” In: (2009).

[78] Deepika Phutela. “The importance of non-verbal communication”. In: *IUP Journal of Soft Skills* 9.4 (2015), p. 43.

[79] Marian Harbach et al. “Sorry, I don’t get it: An analysis of warning message texts”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2013, pp. 94–111.

[80] David Maimon et al. “Restrictive deterrent effects of a warning banner in an attacked computer system”. In: *Criminology* 52.1 (2014), pp. 33–59.

[81] Fiona C Saunders, Andrew W Gale, and Andrew H Sherry. “Conceptualising uncertainty in safety-critical projects: A practitioner perspective”. In: *International Journal of Project Management* 33.2 (2015), pp. 467–478.

[82] Paul KJ Han, William MP Klein, and Neeraj K Arora. “Varieties of uncertainty in health care: a conceptual taxonomy”. In: *Medical Decision Making* 31.6 (2011), pp. 828–838.

[83] Marianne Dainton and Brooks Aylor. “A relational uncertainty analysis of jealousy, trust, and maintenance in long-distance versus geographically close relationships”. In: *Communication Quarterly* 49.2 (2001), pp. 172–188.

[84] Magda Osman. “Controlling uncertainty: a review of human behavior in complex dynamic environments.” In: *Psychological bulletin* 136.1 (2010), p. 65.

[85] Berndt Brehmer. “Dynamic decision making: Human control of complex systems”. In: *Acta psychologica* 81.3 (1992), pp. 211–241.

Part III
Appendix

A .arff header

```
@relation

@attribute Duration numeric
@attribute ’Source port’ numeric
@attribute ’Destination port’ numeric
@attribute Protocol TCP,ICMP,UDP,ESP,GRE,IPIP,IPv6,TLSP,144,197,NARP,ISIS4,
178,Leaf2,160,243,SPS,FC,246,148,OSPF,248,224,255
@attribute Flags ......,.....F,....S.,....SF,...R..,...R.F,...RS.,...RSF,
..P...,..P..F,..P.S.,..P.SF,..PR..,..PR.F,..PRS.,..PRSF,.A....,.A...F,.A..S.,.A..SF,
.A.R..,.A.R.F,.A.RS.,.A.RSF,.AP...,.AP..F,.AP.S.,.AP.SF,.APR..,.APR.F,.APRS.,.APRSF,
U.....,U....F,U...S.,U...SF,U..R..,U..R.F,U..RS.,U..RSF,U.P...,U.P..F,U.P.S.,U.P.SF,
U.PR..,U.PR.F,U.PRS.,U.PRSF,UA....,UA...F,UA..S.,UA..SF,UA.R..,UA.R.F,UA.RS.,
UA.RSF,UAP...,UAP..F,UAP.S.,UAP.SF,UAPR..,UAPR.F,UAPRS.,UAPRSF
@attribute ’Forwarding status’ numeric
@attribute ’Type of service’ numeric
@attribute ’Number of packets’ numeric
@attribute ’Number of bytes’ numeric
@attribute ’Bytes per packet’ numeric
@attribute ’Bytes per second’ numeric

@data
```

B Silhouette coefficient

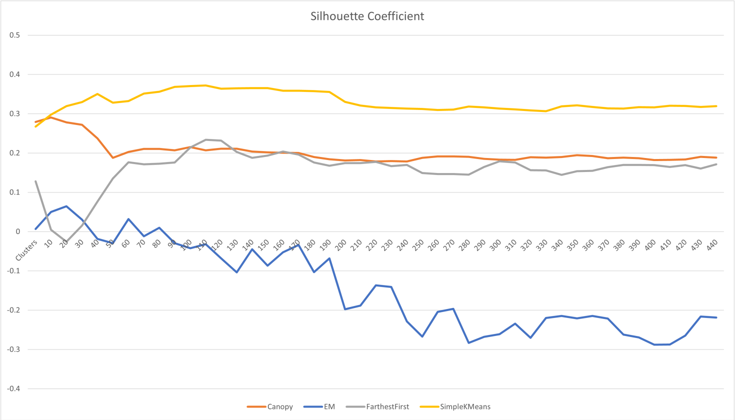
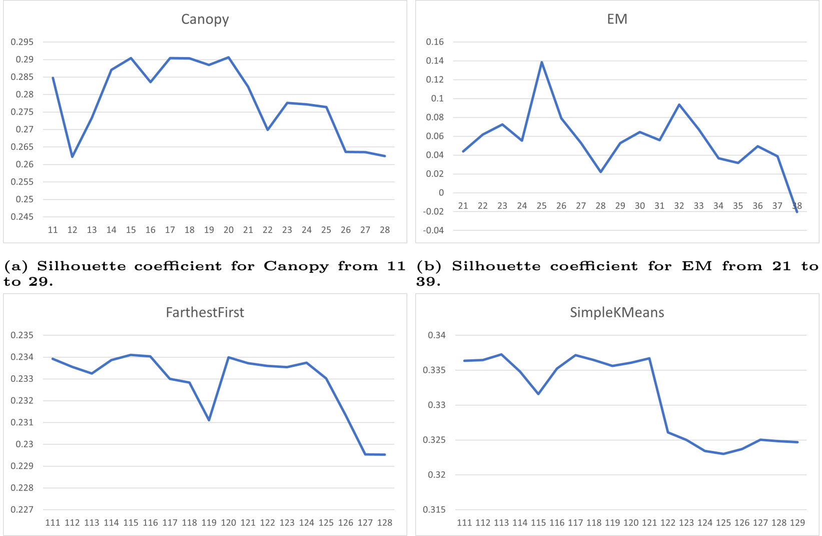


Figure 16: The silhouette coefficient for each of the clustering algorithms calculated from 1 to 440 clusters, using steps of 10.



(a) Silhouette coefficient for Canopy from 11 to 29.

(b) Silhouette coefficient for EM from 21 to 39.

(c) Silhouette coefficient for FarthestFirst from 111 to 129.

(d) Silhouette coefficient for SimpleKMeans from 111 to 129.

Figure 17: The silhouette coefficient for each of the clustering algorithms from x-9 to x+9 where x was the highest value in steps of 10.

C Confusion matrices for random forest

Table 32: Confusion Matrix for RandomForest model M2-3, the actual labels are shows vertically while the classified labels are shown horizontally. Everything on the diagonal is thus classified correctly.

actual\classified	Scan	Neris Botnet	Background	DoS	Anomaly-Spam
Scan	226642	0	10051	0	0
Neris Botnet	0	150976	988	0	0
Background	17006	121277	123929726	410	217041
DoS	0	0	0	391527	0
Anomaly-Spam	0	0	198	0	4632

Table 33: Confusion Matrix for RandomForest model M3-6, the actual labels are shows vertically while the classified labels are shown horizontally. Everything on the diagonal is thus classified correctly.

actual\classified	Scan	Neris Botnet	Background	DoS	Anomaly-Spam
Scan	459120	0	69	0	0
Neris Botnet	0	152007	412	0	0
Background	13523	70423	104352880	2094	23020
DoS	0	0	0	783901	0
Anomaly-Spam	0	0	5838360	0	21

Table 34: Confusion Matrix for RandomForest model M2-6, the actual labels are shows vertically while the classified labels are shown horizontally.

actual\classified	Scan	Neris Botnet	Background	DoS	Anomaly-Spam
Scan	439425	0	19764	0	0
Neris Botnet	0	151931	488	0	0
Background	2669	132795	104007306	450	318720
DoS	0	0	0	783901	0
Anomaly-Spam	0	0	617536	0	5220845

D F1-scores

Table 35: F1-score per model for Random Forest and J48.

Model	Random Forest			J48		
	M2-3	M3-6	M2-6	M2-3	M3-6	M2-6
Scan	0.9437	0.9854	0.9751	0.7724	0.932	0.8278
Neris Botnet	0.7118	0.811	0.6951	0.4439	0.528	0.4349
Background	0.9985	0.9723	0.9948	0.9956	0.9708	0.9813
DoS	0.9995	0.9987	0.9997	0.5576	0.9318	0.9353
Anomaly-Spam	0.0409	0	0.9177	0.0156	0	0.6305

Table 36: F1-score per model for PART and Jrip.

Model	PART			Jrip		
	M2-3	M3-6	M2-6	M2-3	M3-6	M2-6
Scan	0.7833	0.9799	0.9308	0.9333	0.3047	0.9611
Neris Botnet	0.4652	0.3421	0.4679	0.4075	0.5391	0.473
Background	0.9967	0.9693	0.9819	0.9974	0.9674	0.9696
DoS	0.864	0.9299	0.9215	0.9295	0.6631	0.9496
Anomaly-Spam	0.013	0	0.6367	0.0193	0	0

Table 37: F1-score per model for Bagging and ZeroR.

Model	Bagging			ZeroR		
	M2-3	M3-6	M2-6	M2-3	M3-6	M2-6
Scan	0.8004	0.9102	0.9398	0	0	0
Neris Botnet	0.2998	0.4556	0.3878	0	0	0
Background	0.9946	0.9699	0.9953	0.9969	0.9665	0.9665
DoS	0.7518	0.9478	0.9759	0	0	0
Anomaly-Spam	0.028	0	0.9673	0	0	0

Table 38: F1-score per model for Multilayer Perceptron and Logistic Regression.

Model	Multilayer Perceptron			Logistic Regression		
	M2-3	M3-6	M2-6	M2-3	M3-6	M2-6
Scan	0.8919	0.9338	0.9622	0.2096	0.3418	0.3249
Neris Botnet	0.1535	0.1059	0.1564	0.0216	0.0271	0.0291
Background	0.9909	0.957	0.9766	0.9203	0.8966	0.9099
DoS	0.9675	0.9858	0.9923	0.863	0.9085	0.9446
Anomaly-Spam	0.0085	0	0.6322	0.0029	0	0.4947

Table 39: F1-score per model for Naive Bayes.

Model	**Naive Bayes**		
	M2-3	M3-6	M2-6
Scan	0.0989	0.1392	0.1768
Neris Botnet	0.0097	0.0152	0.0118
Background	0.5293	0.685	0.4798
DoS	0.2584	0.328	0.4198
Anomaly-Spam	0.0002	0	0.2193

E Clustering results for SimpleKMeans

Table 40: Clustering results for SimpleKMeans model M2-3

actual\classified	Scan	Neris Botnet	Background	DoS	Anomaly-Spam
Inside	234450	151684	123906606	231527	4830
Outside	2243	280	378854	0	0

Table 41: Clustering results for SimpleKMeans model M2-6

actual\classified	Scan	Neris Botnet	Background	DoS	Anomaly-Spam
Inside	454678	152139	104132621	783901	5838312
Outside	4511	280	329319	0	69

Table 42: Clustering results for SimpleKMeans model M3-6

actual\classified	Scan	Neris Botnet	Background	DoS	Anomaly-Spam
Inside	448342	152094	104131848	398489	5838371
Outside	10847	325	330092	385412	10

F Theory on uncertainties

H. Jalonen et al. [62] performed a systematic literature review of 101 papers. These papers were found systematically and are all based on uncertainty in innovation. They started with a database of 239.843 papers and through different filters, they were left with 101 papers that were relevant to their study. They worked through all these papers and ended up with the following taxonomy of uncertainties: Technological, Market, Regulatory/institutional, Social/Political, Acceptance/legitimacy, Managerial, Timing and Consequence. The power of this literature review is that the authors used a very systematic approach to include a wide variety of papers in which uncertainty play some role.

Another taxonomy is presented by G Magnani et al. [75], which is aimed at uncertainty in entrepreneurship. This paper includes 96 papers in its literature review. The result was four themes of uncertainty: Uncertainty about the outcomes of external environment, uncertainty about other Attacker intentions actions, uncertainty as lack of knowledge and degrees of confidence to deal with uncertainty.

Saunders et al. [81] give five different perspectives on uncertainty in project management. Environmental, individual, complexity, information and temporal. They also link each perspective on concrete sources of uncertainty, of which some have some overlap. Environmental uncertainty describes fAttacker intentions outside of ones own control, such as "competing and conflicting stakeholder demands". Individual uncertainty is the uncertainty that exists within a person such as the "internal state of knowledge or understanding". Complexity uncertainty is caused by how complex a system or product is, such as the complexity of the technology or the "diversity of Attacker intentions and stakeholders". Information uncertainty is caused by a lack of information or understanding. Finally, Temporal uncertainty is related to the timeline in which a project is and the changes in progression. Since this research assumes there is a finished product that can warn the user when necessary, temporal uncertainty is not relevant. The following uncertainties can be interesting for this research: Competing and conflicting stakeholder demands/Diversity of Attacker intentions and stakeholders, Internal state of knowledge and understanding, Technology choice/project complexity, lack of knowledge or understanding/incomplete understanding of cause and effect.

In the medical field, Han et al. [82] splits up uncertainties in three dimensions. The first dimension is the source of the uncertainty, they divide this into three topics: Probability, ambiguity and complexity. The second dimension contains substantive issues and are split up in the following categories: Scientific (data-centered), Practical (system-centered) and Personal (Patient-centered). The last dimension is the locus, this is based on the prior exposure to knowledge about the issue. Does every party know what something means? Do they know whether or not they lack knowledge?

In the case of different Attacker intentions, a parallel can be drawn to interpersonal relationships. L.K. knobloch et al. [76][77] and M. Dainton et al. [83] focus on relational uncertainty, with a strong focus on trust. While in the digital world you might not know who it is that you are communicating with, or that you are even communicating at all, trust is still an important fAttacker intention. Is the Attacker intention sending me messages someone I know and can I trust this source of data?

Osman et al. [84] performed a literature review on the effect of uncertainty on Complex dynamic control tasks, which is a collection of multiple different control tasks in which sequential decisions need to be made to achieve a desired goal [85]. They focused on three different aspects; Task monitoring, self-monitoring and Monitoring & Control Interaction. Four different fields are included, which are Economics, Psychology, Engineering and Human-Computer Interaction. They found that under high uncertainty a person relies more on biases, has a higher persistence of unsuccessful strategies and a poor strategy development when it comes to task monitoring. In regards to self-monitoring, a person has a poorer knowledge of action-outcomes. For the monitoring & control interaction they concluded that a person has a poorer resource allocation. This was all focused on decision making in complex systems so certain types of uncertainty are not touched upon.

G Versions of formulation

G.1 Basic Versions

blacklist
Incoming data is suspected to be a blacklist attack. An untrusted source is sending data to your network. There is most likely malicious data on your network. This problem can be solved by blocking the IP address.

DoS
Incoming data is suspected to be a denial of service (DoS). An attacker is trying to take down your network. Your network performance will be degraded. This problem can be solved by contacting your internet service provider (ISP).

Botnet
Incoming data is suspected to be a botnet attack. An attacker is trying to hack your devices. The attacker can use the hacked devices for fraudulence. This problem can be solved by changing the password for all your network devices.

Port Scan
Incoming data is suspected to be a port scan attack. An attacker is trying to look for open ports. The attacker knows which ports are open or closed. This problem can be solved by checking if port 40 is closed.

G.2 Uncertainty focused additions for formulations

blacklist
Incoming data is suspected to be a blacklist attack. An untrusted source is sending data to your network. The attacker has been caught performing suspicious activities and is now sending data to your network. There is most likely malicious data on your network, it can however also lead to your network or desktop being infected with a virus. This attack can be compared to a person with a criminal record that is trying to find locations where he or she is not known. This problem can be solved by blocking the IP address.

1. Open your router settings page, this can often be done by surfing to 192.168.0.1 in your browser
2. Log in with the admin credentials, these can often be found in the manual
3. Search for Firewall or Block Sites configuration page, these might be located under a Filtering or Security page
4. Add the IP address to the blocked addresses
5. Save the settings and reset the router

DoS
Incoming data is suspected to be a denial of service (DoS). An attacker is trying to take down your network. The attacker is sending you lots of messages, so that your network can not keep up with the incoming data. Your network performance will be degraded, it can however also lead to your network to stop working completely. This attack can be compared to a mob of people blocking a door, which prevents anyone from leaving or entering a building. This problem can be solved by contacting your internet service provider (ISP).

1. First check if your internet is indeed slow or not working
2. If you do not know your ISP, look at a website such as https://whatismyipaddress.com/
3. Find the phone number of your ISP by searching for the customer support contact on the internet
4. Call your ISP
5. Explain to them you are experiencing a denial of service (DoS)

Botnet
Incoming data is suspected to be a botnet attack. An attacker is trying to hack your devices. The attacker is targeting devices in your network and trying to find one that can be infected. The attacker can use the hacked devices for fraudulence, it can however also lead to personal files or data being stolen. This attack can be compared to a someone breaking into a building, to use the facilities or steal the property. This problem can be solved by changing the password for all your network devices.

1. Find which devices are connected to your network, don't forget devices such as smart thermometers
2. Search on the internet for a way to change the password of a device
3. Give the device a strong password, consider using a password generator
4. Disconnect the device if the password cannot be changed

Port Scan
Incoming data is suspected to be a port scan attack. An attacker is trying to look for open ports. The attacker is likely to target any open ports to infect your network or steal your data. This attack can be compared to a burglar trying to find an open window in a building, which they can use to enter. The attacker knows which ports are open or closed, it can however also lead to targeted attacks on the open ports. This problem can be solved by checking if port 40 is closed.

1. Open your router settings page, this can often be done by surfing to 192.168.0.1 in your browser

2. Log in with the admin credentials, these can often be found in the manual
3. Search for Forwarding, this might be located under a Security page
4. If port 40 is forwarded, remove this rule

H Assignment formulations

In the following pages each attack will be presented. First the separate questions of a problem structure will be answered. The structure of each answer is as follows:

What is the problem
Minimal Answer

Why is it a problem
Minimal Answer
Additional Declarative information that is linked to Attacker intention uncertainty
This gives more in depth information about what the attacker is doing or trying to accomplish

What are the consequences
Minimal Answer
Additional Declarative information that is linked to consequence uncertainty
This gives additional dangers or problems that are part of the attack. Going into more detail about the previously stated attack often results in a lot of technical information that is not relevant to most users.

What is the solution
Minimal answer
Additional Procedural information that is linked to competence uncertainty

Metaphor
Additional declarative information that is linked to complexity uncertainty

I Interview introduction

Dear reader,

Thank you for being part of this study! I am Dion de Hoog and I am a student on the TU Delft and I am currently doing research for a thesis in Computer Science and Science Communication. For this research I am working together with the TU Delft and TNO.

The purpose of this study is to find an effective way to communicate technical data in the field of internet security. I am specifically looking at a way to create warning messages in such a way that a user can act upon them.

All data will be anonymized either by omitting traceable data or by putting the raw data into wider bins. Any personal data that is required for the processing of the interviews will not be shared.

This study consists of two parts. In the first part I will ask you to look at different formulations of a problem and rank them based on some criteria. For this part the following scenario should be envisioned: There is a device analyzing the messages that are being sent and received on your gateway/router. You are in a home environment behind your PC. After a suspicious message has been found a notification gets send to your PC/Desktop with the purpose of you taking action to protect your network against the attack without outside help.

The second part will be more like a traditional interview in which I ask some questions and make notes of the answers.

The outcomes of the interviews and surveys will be processed and compared. The conclusions will be presented in the final product. In case of consent, an anonymized version of the answers to the questions will also be included in the appendix.

The final product will be published on the TU Delft educational repository.

J Interview questions

Part 1: Assignment
You will be handed four versions of a message for four different attacks. Please read them in order and rank them based on which message worries you the most. Where the top message is the most worrisome and the bottom message is the least worrisome. After the four messages are ranked, the next attack will be provided.

After all attacks are ranked, please also rank the attacks themselves on the same criteria. The top ranked attack is the gives the most worrisome and the bottom ranked the least worrisome.

Part 2: Questions
1. What is your age?
2. If you have a problem with your PC how do you solve this?
3. How Important is network security for you?
4. What kind of protective measures do you use and why?
5. Why do you or do you not use these?
6. Do you know how messages are send on the internet (if not, give short explanation)?
7. Of which of the following do you know what it means?
NetFlow (if they do not know, explain this), Duration, Port, IP Address, Protocol, Number of packets, Size.
8. Would you like to receive messages as portrayed in part 1 (Show full version of data). And why?
9. Are you likely to act upon messages as portrayed in part 1. And why?
10. Explain link to uncertainties → Which uncertainty do you see most in yourself?
11. In the answers from part 1, it seems uncertainty X is most prevalent, do you recognize this?
12. What would you prefer these messages to look like?

K Interview rankings

In this appendix the answers of each participant will be shown.

Table 43: Answers for participant 1

	DoS	**Botnet**	**Blacklist**	**Port Scan**
1	consequence	consequence	consequence	Attacker intention
2	competence	competence	Attacker intention	consequence
3	Attacker intention	Attacker intention	competence	competence
4	complexity	complexity	complexity	complexity

Table 44: Answers for participant 2

	DoS	**Botnet**	**Blacklist**	**Port Scan**
1	consequence	consequence	consequence	complexity
2	Attacker intention	complexity	complexity	Attacker intention
3	complexity	Attacker intention	Attacker intention	consequence
4	competence	competence	competence	competence

Table 45: Answers for participant 3

	DoS	**Botnet**	**Blacklist**	**Port Scan**
1	Attacker intention	competence	Attacker intention	consequence
2	consequence	consequence	competence	competence
3	complexity	Attacker intention	consequence	Attacker intention
4	competence	complexity	complexity	complexity

Table 46: Answers for participant 4

	DoS	**Botnet**	**Blacklist**	**Port Scan**
1	complexity	complexity	complexity	complexity
2	competence	competence	competence	competence
3	consequence	consequence	consequence	consequence
4	Attacker intention	Attacker intention	Attacker intention	Attacker intention

Table 47: Answers for participant 5

	DoS	**Botnet**	**Blacklist**	**Port Scan**
1	complexity	competence	complexity	Attacker intention
2	Attacker intention	complexity	Attacker intention	consequence
3	consequence	Attacker intention	consequence	complexity
4	competence	consequence	competence	competence

Table 48: Answers for participant 6

	DoS	Botnet	Blacklist	Port Scan
1	complexity	consequence	complexity	Attacker intention
2	consequence	complexity	Attacker intention	consequence
3	Attacker intention	competence	consequence	competence
4	competence	Attacker intention	competence	complexity

Table 49: Answers for participant 7

	DoS	Botnet	Blacklist	Port Scan
1	competence	competence	competence	competence
2	consequence	Attacker intention	Attacker intention	Attacker intention
3	Attacker intention	complexity	consequence	consequence
4	complexity	consequence	complexity	complexity

Table 50: Answers for participant 8

	DoS	Botnet	Blacklist	Port Scan
1	consequence	complexity	consequence	complexity
2	complexity	Attacker intention	Attacker intention	Attacker intention
3	Attacker intention	consequence	competence	consequence
4	competence	competence	complexity	competence

Table 51: Answers for participant 9

	DoS	Botnet	Blacklist	Port Scan
1	consequence	consequence	consequence	Attacker intention
2	competence	competence	competence	competence
3	Attacker intention	Attacker intention	Attacker intention	consequence
4	complexity	complexity	complexity	complexity

Table 52: Answers for participant 10

	DoS	Botnet	Blacklist	Port Scan
1	consequence	competence	Attacker intention	consequence
2	complexity	consequence	complexity	complexity
3	Attacker intention	Attacker intention	consequence	Attacker intention
4	competence	complexity	competence	competence

Table 53: Answers for participant 11

	DoS	Botnet	Blacklist	Port Scan
1	complexity	complexity	competence	complexity
2	competence	competence	complexity	Attacker intention
3	Attacker intention	Attacker intention	Attacker intention	competence
4	consequence	consequence	consequence	consequence

Table 54: Answers for participant 12

	DoS	**Botnet**	**Blacklist**	**Port Scan**
1	consequence	consequence	Attacker intention	Attacker intention
2	competence	Attacker intention	consequence	consequence
3	Attacker intention	competence	competence	competence
4	complexity	complexity	complexity	complexity

Table 55: Answers for participant 13

	DoS	**Botnet**	**Blacklist**	**Port Scan**
1	consequence	competence	competence	competence
2	competence	consequence	Attacker intention	consequence
3	Attacker intention	Attacker intention	consequence	Attacker intention
4	complexity	complexity	complexity	complexity

Table 56: Answers for participant 14

	DoS	**Botnet**	**Blacklist**	**Port Scan**
1	consequence	Attacker intention	consequence	consequence
2	complexity	consequence	complexity	Attacker intention
3	Attacker intention	complexity	Attacker intention	complexity
4	competence	competence	competence	competence

Table 57: Answers for participant 15

	DoS	**Botnet**	**Blacklist**	**Port Scan**
1	Attacker intention	consequence	Attacker intention	consequence
2	complexity	complexity	complexity	complexity
3	consequence	Attacker intention	consequence	Attacker intention
4	competence	competence	competence	competence

Table 58: Answers for participant 16

	DoS	**Botnet**	**Blacklist**	**Port Scan**
1	complexity	consequence	consequence	Attacker intention
2	Attacker intention	Attacker intention	Attacker intention	consequence
3	consequence	complexity	complexity	complexity
4	competence	competence	competence	competence

Table 59: Answers for participant 17

	DoS	**Botnet**	**Blacklist**	**Port Scan**
1	Attacker intention	consequence	consequence	consequence
2	complexity	Attacker intention	complexity	Attacker intention
3	consequence	complexity	Attacker intention	complexity
4	competence	competence	competence	competence

Table 60: Answers for participant 18

	DoS	Botnet	Blacklist	Port Scan
1	consequence	consequence	Attacker intention	consequence
2	complexity	Attacker intention	consequence	Attacker intention
3	Attacker intention	competence	competence	competence
4	competence	complexity	complexity	complexity

Table 61: Answers for participant 19

	DoS	Botnet	Blacklist	Port Scan
1	Attacker intention	Attacker intention	Attacker intention	Attacker intention
2	complexity	consequence	consequence	consequence
3	competence	competence	competence	complexity
4	consequence	complexity	complexity	competence

Table 62: Answers for participant 20

	DoS	Botnet	Blacklist	Port Scan
1	complexity	consequence	complexity	Attacker intention
2	consequence	Attacker intention	Attacker intention	consequence
3	Attacker intention	complexity	consequence	complexity
4	competence	competence	competence	competence

Table 63: Answers for participant 21

	DoS	Botnet	Blacklist	Port Scan
1	Attacker intention	complexity	Attacker intention	consequence
2	complexity	Attacker intention	complexity	complexity
3	consequence	consequence	consequence	Attacker intention
4	competence	competence	competence	competence

L Analysis on rankings of formulations

In this appendix a more in depth look will be taken into the ranking of participants.

L.1 Analysis on consequence uncertainty as most prevalent

Table 64 shows the average ranking of participants that self-assigned consequence. We can see that consequence does indeed have a rather high ranking consequence. For participant 2 and 9, it has the highest rank and for participant 6 it is tied for- the highest rank together with complexity. Participant 15 and 20 have a three way tie for the highest ranking of which only competence is ranked lower. It is clear that for these participants the average ranking for consequence is high and the ranking for competence is low.

Table 64: Average ranking per participants that self-assigned consequence as most prevalent uncertainty together with comments that give further insight into the uncertainties.

Participant	Attacker intention	Consequence	Competence	Complexity
2	2.5	1.5	4	2
6	2.5	2	3.5	2
9	2.5	1.5	2	4
15	2	2	4	2
20	2	2	4	2
Average	2.3	1.8	3.5	2.4

The comments are largely in line with the rankings of the participants, which are shown in Appendix M. Participants 2, 9 and 15 simply stated that consequence is what they are most worried about. Participant 6 however, stated "Most of them gave me a feeling of whatever", which can explain why the rankings are a lot closer. Participant 15 had more peculiar answers, The complexity was ranked second every single time however, this was not because the participant was more worried about the complexity but rather because the metaphor lessens the credibility and thus makes it seem like the message can't be trusted. Attacker intention and consequence were both ranked first twice and ranked third twice, however the reason that Attacker intention was ranked first twice was that "the sentences annoyed me". These rankings are thus a bit more difficult to place. Participant 20 did not directly give a reason for self-assigning consequence, however during the interview his financial information came up as a worry multiple times.

The answers given by participants do seem rather consistent with each other. Outside of competence and complexity for participant 9, the average rankings between participants are all within 0.5 points for each uncertainty. There is also a rather large difference between rankings for different uncertainties.

L.2 Analysis on competence uncertainty as most prevalent

Table 65 shows the average ranking of participants that self-assigned competence.

Table 65: Average ranking per participants that self-assigned competence as most prevalent uncertainty together with comments that give further insight into the uncertainties.

Participant	Attacker intention	Consequence	Competence	Complexity
1	2.25	1.25	2.5	4
7	2.25	3	1	3.75
11	2.75	4	2	1.25
14	2.25	1.25	4	2.5
17	2	1.5	4	2.5
18	2	1.25	3.25	3.5
21	1.75	2.5	4	1.75
Average	2.179	2.107	2.964	2.75

It would be expected that competence has a high ranking for these participants, however, competence actually has the lowest average ranking out of the four uncertainties. We will have to look at the comments to see if they give more context to understand these rankings. The comments that are relevant to the score participants gave can be seen in appendix M.

The comments paint a pretty clear picture, Participant 14, 17 and 21 all state that the list of steps give them peace of mind. Even if they are worried about their competence and might not be able to actually perform the necessary steps they are still less worried when the steps are shown. This could indicate that including a list of steps can lower the worries of participants whether it actually helps them or not. Only participant 7 has competence consistently ranked at the top, the reason for this is that the participant "would like to understand it, but lost it with the switch from MS-DOS". The participant stated that he does not know what to do anymore on modern computers. Another participant with a high ranking is participant 11, this participants states that "When there are only a few small steps it can feel scary".

When comparing the rankings of different participants it is clear that there is quite a large variety. The average rankings are also rather close with the difference between the highest and lowest score only being 0.857.

L.3 Analysis on complexity uncertainty as most prevalent

Table 66: Average ranking per participants that self-assigned complexity as most prevalent uncertainty together with comments that give further insight into the uncertainties.

Participant	Attacker intention	Consequence	Competence	Complexity
3	2	2	2.25	3.75
4	4	3	2	1
5	2	3	3.25	1.75
10	2.5	1.75	3.25	2.5
16	1.75	1.75	4	2.5
19	1	2.75	3.25	3.5
Average	2.208	2.375	3	2.5

It would be expected that complexity has a high rank amount the uncertainties but it actually has the second lowest, only above competence. The ranking are also not very consistent between participants. The comments might give more context to understand these rankings, these comments can be seen in table appendix M.

For participant 3 it becomes clear that complexity is indeed a worry, however, if the system assumes this is the case it feels pedantic. This could mean that a metaphor is not a correct way to make the problem easier to understand, this should be researched in a follow-up study.

Participant 4 simply stated that a metaphor makes the problem more serious and real.

Participant 5 made a good point and stated that once complexity is higher than the competence it is a problem. Complexity was also the highest rank in the answers.

Finally, there is participant 10, which had rather varying answers. The self-assigned uncertainty was complexity while a comment was made that consequence is the most worrying, which is in line with the rankings. This participant does have a background in IT and is currently working in a cyber-security company. This might lead to complexity being a worry but not directly in these messages because it was rather easy or straight forward to understand.

Only participant 4 has complexity ranked highest for every attack. This participant states that a metaphor makes the problem more serious and real. Participant 5 also has a high ranking for complexity, this participant did not directly comment on the metaphor but did state that ”when the complexity is higher than the competence I am screwed”. Participant 3 has a rather interesting comment, ”Metaphor was pedantic. I don’t understand it but I don’t want the system to assume that”. This indicates that people with similar worries are not necessarily effectively reached in the same way. One interesting observation is that participant 10 and 19 are both concerned about the complexity while they are both familiar with cyber security. However, when it comes to the ranking the complexity is not ranked very high since they are both familiar with the attacks. This might be a case of ”The more you know, the more you realize you don’t know”. Even though these participants are experienced and can take care of the presented problems, they also realize the developments in the field of cyber attacks are very quick.

M Comments from participants

In this appendix an overview of relevant comments will be presented. These comments can be seen in Table 67 and Table 68

Table 67: Comments made by participants 1-10 that self-assigned consequence as most prevalent uncertainty.

Participant	Comment
1	Metaphor is useless, I do not understand it and I don't have to.
	I do not know what to do.
	I do not understand computers.
2	Consequence is most worrying.
3	Metaphor was pedantic. I don't understand it but I don't want the system to assume that.
	Consequence and Attacker intention are not directly what I'd be worried about.
4	Metaphor makes the problem more serious and real.
5	If Complexity is higher than my competence I am screwed.
6	Most of them gave me a feeling of "whatever".
	Metaphor makes it more tangible
	Can I trust/do the step-by-step?
7	I would like to understand it, but lost it with the switch from MS-DOS.
8	It is easy as long as you know how to do it.
	List of steps makes it easier because you know how to do it.
9	I am very sensitive to consequences.
10	I recognize consequence as the most worrying (in these messages).
	Other Attacker intentions if I see in the context of a security firm.

Table 68: Comments made by participants 11-21 that self-assigned consequence as most prevalent uncertainty.

Participant	Comment
11	Metaphor makes it more real without giving a solution.
	When there are only a few small steps it can feel scary.
	The more steps I have the easier it is to google.
	If the metaphor is something I would take seriously, then I should also take the problem seriously.
12	DoS made it sound like the network should be able to solve it itself, so Attacker intention was ranked lower.
	Metaphor sounded very childish so it gave less pressure.
13	Complexity is not an unknown unknown, I know what to research.
	Attacker intention can be anyone and I do not need this info.
	For consequence, if I know why I should do something it motivates me.
	For competence, am I able to do something about it?
	Competence and consequence are more concrete/urgent.
	Complexity gives some ease because it explains more.
14	The list (of steps) gives me peace of mind so that's why I do not worry about it anymore.
	Consequence would be a good second option.
	botnet has a higher score for Attacker intention because fraud is scarier.
15	Consequence is most prevalent.
	Metaphor lessens credibility.
16	Think can be complex to me because I do not understand them, so am I making the right choice or taking the right action?
	I am mainly worried about the consequences.
	Consequence is not most prevalent but I do react to it in the message.
	Human empathy does not fit a computer.
17	I am not digitally illiterate, but I'm close to it.
	Consequence makes it more dangerous, especially when it depends on my competence.
	List of steps give peace of mind, I wouldn't be able to come up with it myself
18	I am afraid of doing something wrong, which will have consequences
	That means competence is my biggest uncertainty
19	I am worried about the complexity of everything nowadays.
	Consequence is the lowest because whatever happens, happens.
	Data is sold, I distrust Google, Amazon and other Attacker intentions.
20	I am worried about my competence but the steps give support
21	It is easiest for me when I know what to do
	The list gives me peace of mind, I do not know if I can perform the steps but at least it says what to do

N Reading complexity analysis

Table 69: Reading complexity analysis for blacklist attacks

Blacklist	Words	Syllables	Sentences	Extended words	Extended syllables	Extended sentences	Complexity simple	Complexity extended	Type of minimal sentence	Type of extended sentence
What is the problem	9	16	1	X	X	X	47.3		Simple	
Why is it a problem	8	13	1	22	32	2	61.24	69.785	Simple	Complex 1 dependent clause
What are the consequences	9	14	1	15	24	1	66.1	60.705	Simple	Simple
What is the solution	10	14	1	66	105	5	78.245	61.51254	Simple	5 sentences with 3 compound
Metaphor	25	34	1	X	X	X	66.404		Complex 1 dependent clause	

Table 70: Reading complexity analysis for DoS attacks

DoS	Words	Syllables	Sentences	Extended words	Extended syllables		Complexity simple	Complexity extended		
What is the problem	12	21	1	X	X		46.605		Simple	
Why is it a problem	9	13	1	20	30	2	75.5	71.58195	Simple	Complex 1 dependent clause
What are the consequences	8	14	1	12	19	1	50.665	57.095	Simple	Simple
What is the solution	12	20	1	58	88	5	53.655	64.46762	Simple	5 sentences with 3 compound
Metaphor	22	33	1	X	X		57.605		Complex 1 dependent clause	

Table 71: Reading complexity analysis for botnet attacks

Botnet	Words	Syllables	Sentences	Extended words	Extended syllables		Complexity simple	Complexity extended		
What is the problem	9	16	1	X	X		47.3		Simple	
Why is it a problem	8	13	1	17	27	2	61.24	63.01667	Simple	Complex 1 dependent clause
What are the consequences	9	15	1	12	20	1	56.7	55.1775	Simple	Simple
What is the solution	14	20	1	49	78	4	71.76785714	62.446	Simple	4 sentences with 2 compound
Metaphor	20	31	1	X	X		55.405		Complex 1 dependent clause	

Table 72: Reading complexity analysis for port scan attacks

Port scan	Words	Syllables	Sentences	Extended words	Extended syllables		Complexity simple	Complexity extended		
What is the problem	10	16	1	X	X		61.325		Simple	
Why is it a problem	9	13	1	16	26	2	75.5	66.40067	Simple	Complex 1 dependent clause
What are the consequences	11	18	1	12	19	1	57.23363636	59.06685	Simple	Simple
What is the solution	12	15	1	53	84	4	88.905	64.78769	Simple	4 sentences with 3 compound
Metaphor	23	31	1	X	X		69.46391304		Complex 1 dependent clause	

Table 73: Reading complexity for the combined sentences

Attack	**Words**	**Syllables**	**Sentences**	**Complexity**
Blacklist	164	252	13	64.03526266
DoS	153	238	13	63.28923077
Botnet	138	220	12	60.29293478
Port Scan	146	222	12	65.84747717

Table 74: Ranking of formulations for participants 1 through 14

	DoS	**Botnet**	**Blacklist**	**Port Scan**	**Times #**	**1**	**2**	**3**	**4**	**Avg.**	**Self assigned uncertainty**
1	consequence	consequence	consequence	actor	**actor**	1	1	2	0	2.25	competence
2	competence	competence	actor	consequence	**consequence**	3	1	0	0	1.25	
3	actor	actor	competence	competence	**competence**	0	2	2	0	2.5	
4	complexity	complexity	complexity	complexity	**complexity**	0	0	0	4	4	
1	consequence	consequence	consequence	complexity	**actor**	0	2	2	0	2.5	consequence
2	actor	complexity	complexity	actor	**consequence**	3	0	1	0	1.5	
3	complexity	actor	actor	consequence	**competence**	0	0	0	4	4	
4	competence	competence	competence	competence	**complexity**	1	2	1	0	2	
1	N.A	actor	complexity	N.A.	**actor**	1	0	1	0	1	actor
2	N.A	complexity	consequence	N.A.	**consequence**	0	1	1	0	1.25	
3	N.A	consequence	actor	N.A.	**competence**	0	0	0	2	2	
4	N.A	competence	competence	N.A.	**complexity**	1	1	0	0	0.75	
1	actor	competence	actor	consequence	**actor**	2	0	2	0	2	complexity
2	consequence	consequence	competence	competence	**consequence**	1	2	1	0	2	
3	complexity	actor	consequence	actor	**competence**	1	2	0	1	2.25	
4	competence	complexity	complexity	complexity	**complexity**	0	0	1	3	3.75	
1	complexity	complexity	complexity	complexity	**actor**	0	0	0	4	4	Complexity
2	competence	competence	competence	competence	**consequence**	0	0	4	0	3	
3	consequence	consequence	consequence	consequence	**competence**	0	4	0	0	2	
4	actor	actor	actor	actor	**complexity**	4	0	0	0	1	
1	complexity	competence	complexity	actor	**actor**	1	2	1	0	2	Complexity
2	actor	complexity	actor	consequence	**consequence**	0	1	2	1	3	
3	consequence	actor	consequence	complexity	**competence**	1	0	0	3	3.25	
4	competence	consequence	competence	competence	**complexity**	2	1	1	0	1.75	
1	complexity	consequence	complexity	actor	**actor**	1	1	1	1	2.5	Consequence
2	consequence	complexity	actor	consequence	**consequence**	1	2	1	0	2	
3	actor	competence	consequence	competence	**competence**	0	0	2	2	3.5	
4	competence	actor	competence	complexity	**complexity**	2	1	0	1	2	
1	competence	competence	competence	competence	**actor**	0	3	1	0	2.25	Competence
2	consequence	actor	actor	actor	**consequence**	0	1	2	1	3	
3	actor	complexity	consequence	consequence	**competence**	4	0	0	0	1	
4	complexity	consequence	complexity	complexity	**complexity**	0	0	1	3	3.75	
1	consequence	complexity	consequence	complexity	**actor**	0	3	1	0	2.25	Complexity/Competence
2	complexity	actor	actor	actor	**consequence**	2	0	2	0	2	
3	actor	consequence	competence	consequence	**competence**	0	0	1	3	3.75	
4	competence	competence	complexity	competence	**complexity**	2	1	0	1	2	
1	consequence	consequence	consequence	actor	**actor**	1	0	3	0	2.5	Consequence
2	competence	competence	competence	competence	**consequence**	3	0	1	0	1.5	
3	actor	actor	actor	consequence	**competence**	0	4	0	0	2	
4	complexity	complexity	complexity	complexity	**complexity**	0	0	0	4	4	
1	consequence	competence	actor	consequence	**actor**	1	0	3	0	2.5	Complexity
2	complexity	consequence	complexity	complexity	**consequence**	2	1	1	0	1.75	
3	actor	actor	consequence	actor	**competence**	1	0	0	3	3.25	
4	competence	complexity	competence	competence	**complexity**	0	3	0	1	2.5	
1	complexity	complexity	competence	complexity	**actor**	0	1	3	0	2.75	Competence
2	competence	competence	complexity	actor	**consequence**	0	0	0	4	4	
3	actor	actor	actor	competence	**competence**	1	2	1	0	2	
4	consequence	consequence	consequence	consequence	**complexity**	3	1	0	0	1.25	
1	consequence	consequence	actor	actor	**actor**	2	1	1	0	1.75	Actor
2	competence	actor	consequence	consequence	**consequence**	2	2	0	0	1.5	
3	actor	competence	competence	competence	**competence**	0	1	3	0	2.75	
4	complexity	complexity	complexity	complexity	**complexity**	0	0	0	4	4	
1	consequence	competence	competence	competence	**actor**	0	1	3	0	2.75	Consequence/Competence
2	competence	consequence	actor	consequence	**consequence**	1	2	1	0	2	
3	actor	actor	consequence	actor	**competence**	3	1	0	0	1.25	
4	complexity	complexity	complexity	complexity	**complexity**	0	0	0	4	4	

Table 75: Ranking of formulations for participant 16 through 22

	DoS	**Botnet**	**Blacklist**	**Port Scan**	**Times #**	**1**	**2**	**3**	**4**	**Avg.**	**Self assigned uncertainty**
1	consequence	actor	consequence	consequence	**actor**	1	1	2	0	2.25	Competence
2	complexity	consequence	complexity	actor	**consequence**	3	1	0	0	1.25	
3	actor	complexity	actor	complexity	**competence**	0	0	0	4	4	
4	competence	competence	competence	competence	**complexity**	0	2	2	0	2.5	
1	actor	consequence	actor	consequence	**actor**	2	0	2	0	2	Consequence
2	complexity	complexity	complexity	complexity	**consequence**	2	0	2	0	2	
3	consequence	actor	consequence	actor	**competence**	0	0	0	4	4	
4	competence	competence	competence	competence	**complexity**	0	4	0	0	2	
1	complexity	consequence	consequence	actor	**actor**	1	3	0	0	1.75	Complexity
2	actor	actor	actor	consequence	**consequence**	2	1	1	0	1.75	
3	consequence	complexity	complexity	complexity	**competence**	0	0	0	4	4	
4	competence	competence	competence	competence	**complexity**	1	0	3	0	2.5	
1	actor	consequence	consequence	consequence	**actor**	1	2	1	0	2	Competence
2	complexity	actor	complexity	actor	**consequence**	3	0	1	0	1.5	
3	consequence	complexity	actor	complexity	**competence**	0	0	0	4	4	
4	competence	competence	competence	competence	**complexity**	0	2	2	0	2.5	
1	consequence	consequence	actor	consequence	**actor**	1	2	1	0	2	Competence
2	complexity	actor	consequence	actor	**consequence**	3	1	0	0	1.25	
3	actor	competence	competence	competence	**competence**	0	0	3	1	3.25	
4	competence	complexity	complexity	complexity	**complexity**	0	1	0	3	3.5	
1	actor	actor	actor	actor	**actor**	4	0	0	0	1	Complexity
2	complexity	consequence	consequence	consequence	**consequence**	0	2	1	1	2.75	
3	competence	competence	competence	complexity	**competence**	0	0	3	1	3.25	
4	consequence	complexity	complexity	competence	**complexity**	0	0	2	2	3.5	
1	complexity	consequence	complexity	actor	**actor**	1	2	1	0	2	Consequence
2	consequence	actor	actor	consequence	**consequence**	1	2	1	0	2	
3	actor	complexity	consequence	complexity	**competence**	0	0	0	4	4	
4	competence	competence	competence	competence	**complexity**	2	0	2	0	2	
1	actor	complexity	actor	consequence	**actor**	2	1	1	0	1.75	Competence
2	complexity	actor	complexity	complexity	**consequence**	1	0	3	0	2.5	
3	consequence	consequence	consequence	actor	**competence**	0	0	0	4	4	
4	competence	competence	competence	competence	**complexity**	1	3	0	0	1.75	