Impact of Virtual Networks on Anomaly Detection with Machine Learning

Daniel Spiekermann
Polizeiakademie Niedersachsen
daniel.spiekermann@polizei.niedersachsen.de

Jörg Keller
FernUniversität in Hagen
joerg.keller@fernuni-hagen.de

***Abstract*—The enormous number of network packets transferred in modern networks together with the high-speed transmissions hamper the implementation of successful IT security mechanisms. In addition to this, virtual networks create highly dynamic and flexible environments, which differ widely from well-known infrastructures of the past decade. Network forensic investigation aiming at the detection of covert channels, malware usage or anomaly detection is faced with new problems and gets a time-consuming, error-prone and complex process. Machine learning provides advanced techniques to perform this work faster with a lower error rate. Depending on the learning technique, algorithms work nearly without any necessary interaction to detect relevant events in the transferred network packets. Occurring changes are noticed and additional processes might be started. Current algorithms work well in static environments, but the highly-dynamic environments of virtual networks create additional events, which might irritate the anomaly detection algorithms. This paper analyses virtual network protocols like VXLAN, GRE and GENVE and their impact of the detection rate of anomalies in the environment. Our research shows the need for adapted pre-processing of the network data, in the worst case on demand if changes are detected.**

***Index Terms*—virtual networks, anomaly detection, covert channels, machine learning**

I. Introduction

Nowadays IT environments are an important parameter in our modern life. Modern data centers form the basis for our everyday digital life. Digital services play an important role, in the private life (i. e. as a backup for photos, videos and files or as a shared online calender) as well as in many professional areas such as the financial sector, development & research or the office environment. With the evolution of cloud computing, ubiquitous use of computers, digital services and resources become more and more normal in our everyday life and demand for faster connections and higher data rates. To fulfill these demands, modern datacenters require a highly flexible infrastructure, which is adaptable without any further administrative work. The implementation of various virtualisation layers like virtual machines (VM), virtual networks and virtual storage provide such an on-demand infrastructure. This led to the implementation of advanced techniques like container-based environments, which require a dynamic infrastructure as well as the different cloud services like Software-as-a-Service, Platform-as-a-Service or Infrastructure-as-a-Service.

Research for this paper was done while the first author was employed at FernUniversität in Hagen.

978-1-7281-5684-2/20/$31.00 ©2020 IEEE

Cloud service providers (CSP) use the virtualisation inside their datacenters to comply with the demands of their customers. Especially the use of VMs improves the on-demand provision of new systems. But the connection of these VMs with a hardware-based network hamper the necessary adaptability. Only with use of virtual networks (VNs) the datacenter plays to its strength. These VNs work on an additional layer in the environment which led to the designation of an underlay and an overlay network as shown in Figure 1.

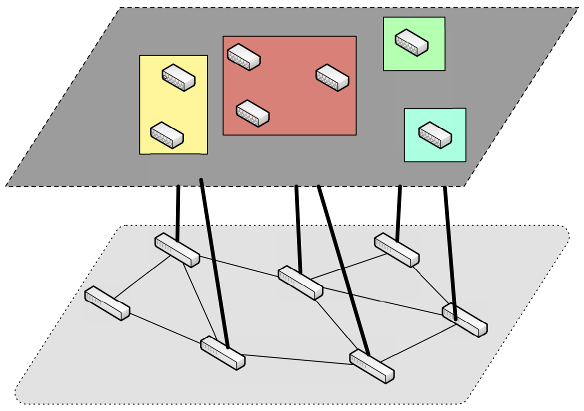


Fig. 1. Underlay and overlay networks.

The overlay networks perform various tasks. On the one hand, a VM needs access to the internet and maybe to different internal or external networks. On the other hand, the same VM has to be separated from VMs of other customers to ensure an isolated environment. Typically this separation is done with virtual networks, which run on top of the hardware-based underlay network.

Protocols that create the overlay network are so-called virtual network protocols, which are used for the interconnection of the different VMs in modern infrastructures. With these protocols, VMs of the same customer are connected together in a logical subnet, which is separated from other subnets of different customers. The hardware-based underlay network and its addressing scheme, routing rule-set or security features could be implemented in a static manner, the virtual network provides the flexibility and dynamic needed to interconnect the VMs. Various protocols exist to implement a VN, each of them with a different focus. The first way to implement a virtual subnet was the use of Virtual LAN (VLAN) [1]. The increasing demand in a datacenter led to the development of adapted protocols Generic Network Virtualization Encapsulation (GENEVE) [2], Network Virtualization using Generic Routing Encapsulation (NVGRE) [3], Stateless Transport Tunneling (STT) [4] or Virtual Extensible LAN (VXLAN) [5].

Due to the evolution of these dynamic environments the protection of the network and the internal services gains more and more in importance. Therefore the detection of attacks or the occurrence of anomalies in the environment is a relevant part of the IT security implementations. A modern network is target of various attacks, either from inside or from outside attackers. The types of attacks vary, from used misconfigurations like in 2018, when Amazon S3 buckets with more than 70 million records were leaked due to poor configuration [6], to ransomware attacks like 2019, when one of the biggest US data center providers *CyrusOne* was attacked [7]. By using different security mechanisms like firewalls or intrusion detection systems providers try to increase their overall IT security and counter these kinds of attacks. But the increasing number of network packets transferred inside the environment in combination with the high-speed connections as well as the huge amount of attacks make this task complex and expensive.

Advanced techniques like machine learning (ML) try to support the cloud service provider (CSP) by the detection of anomalies in the network traffic which might be an indicator for unknown attacks against their infrastructure.

ML and its impact on cyber security is a fast growing research area, which results in the definition of different algorithms and an improved analysis of unknown data. One of the most important parts for ML in networks is the detection of anomalies, which [8] defines as

> *Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior.*

The detection of anomalies in the network is part of classification problems [9]. These problems are described by a data point, which has to be classified to a given category [10]. An easy classification in IT security is the classification of SPAM [11]; here an incoming email is checked against a set of features. To analyze an email correctly, the classifier has to be trained with benign and malicious data. The classifier uses this training data to learn parameters, which indicate SPAM mails.

Anomaly detection in networks tries to find changes according the typical network packets in this network. This might be an indicator for the beginning of an attack or a current data leakage [12]. The detection of covert channels with the help of ML is a recent research area [13]. Modern malware uses covert channels to transfer their payload or to exfiltrate sensitive data [14]. The detection of such attacks require a good knowledge of the traffic which is typical in this environment. Each outlier of this known traffic is an indicator for an issue, therefore it need additional investigation of those traffic. Common classifiers use network flows or parts of protocol headers to create a benchmark data-set, which is used to train the algorithm.

Because of this, machine learning algorithms heavily depend on an environment with some mostly static parameters. If the threshold of changes is reached, the classifier will detect an anomaly and start the pre-defined process. Unfortunately, changes are inherent in virtual networks, so an algorithm will produce various false-positive messages, which therefore led to a reconfiguration of the training sets. The changes in a virtual network are mostly unpredictable, because the administrator as well as the customer can adapt this part of the network. The administrator might change some parameters of the virtual protocols like a change of the used protocol, or an update of some protocol fields. The customer might change the internal ip-addresses used in the own logical subnet. Because of the separation of the layers, both changes get active without any impact to the others. Hence, we investigate the influence of the use of virtual networks on the detection capabilities of machine learning algorithms for malware detection by finding network anomalies. We do this by implementing various virtual network protocols, which transfer pre-defined information in a test environment, capturing the traffic and analyzing it with different algorithms for clustering and classification.

The main contribution of our research are:

- We identify the role of different protocols for network virtualization on detection algorithms.
- We experimentally demonstrate the differences between physical and virtual networks with respect to malware use and malware detection.
- We explore the difficulties that machine learning algorithms encounter when being trained and applied on virtual instead of physical networks.

The reminder of this paper is structured as follows: In Section II we list related work of the fields from virtual networks and machine learning for anomaly detection. Section III describes the methodology and the data creation. The implementation of the used algorithm is discussed in Section IV. In Section V we discuss the results, Section VI concludes this paper and gives an outlook to our future research.

II. Related work

Virtual networks and modern datacenters change the well-known methods of digital investigation in hardware-based networks as discussed in [15], [16]. [17] describes the arising problems of network forensic investigation in virtual networks. [18] defines an SDN-model, which is usable to perform secure network forensic investigation in nowadays datacenters, especially when they are distributed over different locations. The need for a special process to implement valid investigation in modern environments is discussed in [19]. A further discussion about the problems of packet captures for law enforcement in modern datacenters is discussed in [20]. An important task in a datacenter's security strategy is the detection of abnormal behavior inside the transmitted network packets [21]. So the anomaly detection is a part of forensic investigation, but in some cases the results of such a monitoring has to be accessible much faster. [22] discusses the use of machine learning aspects as a implementation of *automated network forensics*. [23] discuss problems and countermeasures of anomaly detection in big data networks. The most notable protocol in modern datacenters is apart from ethernet the *Internet Protocol*. [24] analyses various sources of network data like routing or management protocols and special network probes. The research of anomaly detection in virtual networks is thin, [25] describes the detection of DDoS in virtual networks with the help of the network analyzer *Bro*, which results are used to configure parts of the virtual network with the help of OpenFlow.

III. Architecture

Anomaly detection is a critical task in modern networks, either to detect advanced attacks like covert channels or DDoS. Occurring anomalies in the network might be an indicator for malicious behavior, so using ML to achieve this is a common technique. Whereas anomaly detection in traditional networks is well researched, the algorithms in virtual networks are faced with new challenges. The risk of of changes, that might affect the used classifier, is high, so our research focuses on the impact of such changes to ML algorithm. Such changes arise at various position inside the network and cover the underlay physical network as well as the virtual networks which reside above. Algorithms trained for the detection of anomalies in the network might recognize such changes and therefore might calculate misleading results.

A. Process model

A successful use of ML for network forensic investigation or anomaly detection depends on various parameters like valid packet captures, correct data extraction and the implementation of a suitable algorithm. A proven method in digital investigation to ensure a correct process is the use of so-called process models or frameworks, which define the necessary steps, mostly separated in different phases. Whereas different framework for anomaly detection in networks exist [26], [27], there is no specific framework with a special view regarding the dynamic of a virtual network. As shown in [28], digital investigations in a virtual network require adapted frameworks which are able to manage the flexibility of the environment. Because of this we use the process model defined in [29]. The authors propose six steps to implement ML for network analysis:

- **Problem formulation**
  In this phase the investigator defines various parameters of his analysis. Algorithm of ML are often time-consuming, so a detailed definition of needed data and ML category is quite relevant for the subsequent steps.
- **Data collection**
  In this phase the relevant packets are captured. In traditional networks, this step is easy to implement [15], [17], but virtual network increase the complexity of network packet capture processes [30].
- **Data analysis**
  This phase summarizes all necessary steps to transform the captured packets into a usable format for the following steps. The captured data might be stored in raw, pcap or pcap-ng-formats[1] and transformed in formats like *Netflow*, *sFlow*, *csv*, *json* or other user-defined formats. These techniques do not store the the entire network packet, but different header information of the layer of the OSI-model. Whereas Netflow and sFlow use layer 3 and layer 4 protocols, the other formats might extract information from all other layers. The definition of the needed data depends on the planned analysis.
- **Model construction**
  The construction involves the training, testing and tuning of the learning model.
- **Model validation**
  In this phase the model is validated to guarantee the quality of the working model. If errors occur or improvements are needed, all prior tasks are involved to eradicate this issues.
- **Deployment and inference**
  This phase summarizes all relevant steps to implement the ML process in the operational environment with a focus on resource usage, accuracy and performance.

Our research focuses on the impact of virtual network and the inherent changes on ML algorithm, that are implemented to detect anomalies in the network. Therefore we need to capture all packets, that touch this aspect, the process of data collection is described in detail in the next section. The analysis of the data as well as the model-dependent phases are discussed in Section IV. We did not focus on the last phase of deployment and inference in depth, because we limit our approach to the detection and various statistical values. An optimization of our algorithm is not in scope of this paper.

B. Data collection

The virtual environment provides various flexible changes on demand without any further interaction. Therefore the overlay network as well as the internal virtual network may be changed without any impact on the other part of the network. The internal network uses typically a private ip-address from a predefined subnet as described in [31]. The user of this network is able to change this internal addressing scheme without involving the CSP or any administrator of the cloud environment. On the other hand, the CSP is free to change the underlying network whenever needed, which might led to a fully different network behavior without any effect to the virtual network of the users. If the CSP changes the separation of the internal virtual networks from VXLAN to GRE, it only needs a quick change of the internal transfer mechanism.

To analyze the different situation, we created a test environment, that provides the ability of fast changes in both

[1]In addition to this various vendor-dependent file formats exist.

networks. Changes in the user network are distinguished as overlay changes and mention the change of the ip-addresses from 10.0.0.0/24 to 172.16.99.0/24. Manipulations in the underlay network or with the virtual protocols are differentiated in the change of the virtual protocol, each in combination with an additional VLAN-tagging. This led to twelve different scenarios, as shown in Table I.

TABLE I
CHANGES OF VIRTUAL ENVIRONMENT

Testrun	Overlay	Underlay
1	10.0.0.0/24	XVLAN
2	172.16.99.0/24	VXLAN
3	10.0.0.0/24	VXLAN + LAN
4	172.16.99.0/24	VXLAN + VLAN
5	10.0.0.0/24	GRE
6	172.16.99.0/24	GRE
7	10.0.0.0/24	GRE + VLAN
8	172.16.99.0/24	GRE + VLAN
9	10.0.0.0/24	GENEVE
10	172.16.99.0/24	GENEVE
11	10.0.0.0/24	GENEVE + VLAN
12	172.16.99.0/24	GENEVE + VLAN

We used KVM as the hypervisor for the VMs, which are interconnected via two Open vSwitch (OVS) instance as the internal switch as shown in Figure 2. The connection of the VM to the OVS instance was established via the *tap0*-interface.

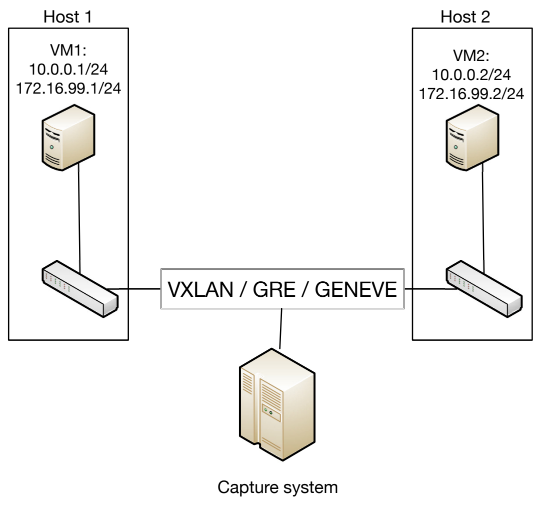


Fig. 2. Architecture of test environment

We used different combination of the protocols to achieve a heterogeneous data-set. The used implementation of Open vSwitch does not provide any support to NVGRE or STT, so we focused on VLAN, VXLAN,GRE and GENEVE.

The easiest way to implement a virtual network is by using the well-known VLAN protocol that is however limited to only 4096[2] subnets, which is not sufficient in modern networks.

[2]VLAN headers use 12 bit VLAN IDs to implement up to $2^{12} = 4096$ subnets.

The most notable protocol to implement these virtual networks is VXLAN, which is similar to the well-known VLAN-protocol, but expands its features and adds some new. VXLAN increases the number of possible virtual separated subnets by using a 24 bit virtual network identifier (VNI) to $2^{24} = 16,777,216$ networks. VXLAN uses UDP to tunnel the network packets with VXLAN. Figure 3 shows the encapsulation with VXLAN.

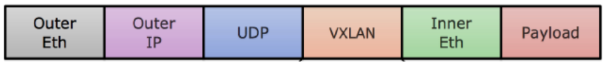


Fig. 3. VXLAN encapsulation

The use of the generic routing encapsulation (GRE) protocol bases on a own header without any additional layer information. Figure 4 shows the encapsulation with GRE.

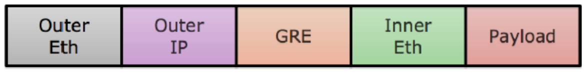


Fig. 4. GRE encapsulation.

GENEVE is similar to VXLAN, because it uses an additional UDP header, too. Figure 5 shows the GENEVE header. In contrast to VXLAN, the header of GENEVE is more generic and allows a flexible modification regarding to the environment it is used in.

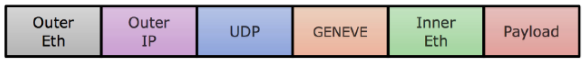


Fig. 5. GENEVE encapsulation.

We configured the two OVS-instances with the internal *ovs-vsctl* command as follows.

- **VXLAN**

```
# Host 1
ovs-vsctl add-br br0
ovs-vsctl add-port br0 vxlan0 \
-- set interface vxlan0 type vxlan \
 options:remote_ip=IP_HOST2
# Host 2
ovs-vsctl add-br br0
ovs-vsctl add-port br0 vxlan0 \
-- set interface vxlan0 type vxlan \
options:remote_ip=IP_HOST1
```

  The other configuration of GRE and GENEVE were similar, it is only necessary to exchange the type of the connection from *vxlan* to *gre* or *geneve*.

- **VLAN**

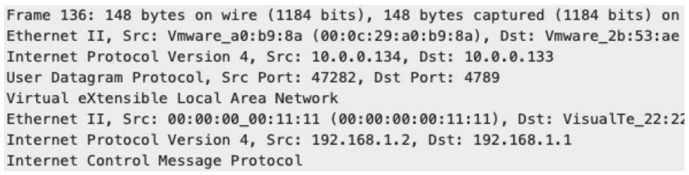
```
# Host 1
ovs-vsctl add-br br0
ovs-vsctl add-port br0 tap0 tag=100
# Host 2
```

```
ovs-vsctl add-br br0
ovs-vsctl add-port br0 tap0 tag=100
```

The capture process was implemented between the connection of host1 and host2. All network packets were captured and stored in a file with the pcap-ng-format.

IV. Analysis

The captured packets were analyzed to verify the correct capture. As shown in Figure 6 the captured data contains all information of the different layers and the encapsulation.



```
Frame 136: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits) on
Ethernet II, Src: Vmware_a0:b9:8a (00:0c:29:a0:b9:8a), Dst: Vmware_2b:53:ae
Internet Protocol Version 4, Src: 10.0.0.134, Dst: 10.0.0.133
User Datagram Protocol, Src Port: 47282, Dst Port: 4789
Virtual eXtensible Local Area Network
Ethernet II, Src: 00:00:00_00:11:11 (00:00:00:00:11:11), Dst: VisualTe_22:22
Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.1
Internet Control Message Protocol
```

Fig. 6. Wireshark output of VXLAN-capture.

A. Definition of the feature set

The definition of the correct features to train the algorithms heavily depends on the goal of the analysis. So the selection of significant parameters which differ benign from malicious traffic is still a difficult process, which depends on various parameters and aspects of the network. In [32] a first feature set of 23 parameters is defined to detect DDoS attacks, after measuring the importance a set of eight parameters were defined as necessary. In contrast to this research, [33] uses only four features consisting only of the parameters source and destination ip-address and combinations of them. Advanced attacks like covert channels do not only use application protocols like HTTP or DNS, but implement their information in lower level protocols like IP or TCP.

Because of this we analyzed all possible protocols of the involved layers with a focus on the changes that arise by using virtual networks with a deeper view on the protocols VLAN, VXLAN, GRE and GENEVE. So we exclude all parameters which to not change during the use of virtual protocols. As shown in Figures 3 and 4, virtual protocols do not manipulate higher protocols like the application layer protocols, but add additional information of layers 2, 3 and 4. Table II lists the additional information for each protocol.

TABLE II
Modification of the OSI-layer by virtual protocols.

Protocol	**Layer**		
	2	3	4
VLAN	x	-	-
VXLAN	x	x	x
GRE	x	x	-
GENEVE	x	x	-

So all aspects of *Ethernet*, *IP* and *TCP/UDP* as well as generic values like the frame.length or the time of the packets are relevant. These changes cover modifications on the underlay layer, changes inside the virtual networks typically concern to changes of the ip-addresses used. The use of length parameters of the protocols like *frame.len* and *ip.len* is useful for the detection of various covert channels like discussed in [34]. So there is no defined list of relevant protocol fields, which should be used for anomaly detection in network forensic investigation.

To simplify our approach, we reduce the impact of the application data and its related header fields, we choose a pre-defined communication based on the ICMP-protocol. Our test environment based on an internal LAN without any effects from external networks like packet loss, changing transmission times or different routes between the involved hosts, so all timing-parameter as well as the Time-to-live of the packets were discarded. Table III lists our feature set, the name of the feature derives from the display-filter name used by *Wireshark*.

TABLE III
Used feature set

Feature	**Layer**	**Description**
frame.len	-	Length of the entire frame
eth.src	2	MAC-address of the source
eth.dst	2	MAC-address of the destination
eth.type	2	Protocol ID of the upper layer
vlan.id	2	ID of the used VLAN
ip.len	3	Length of the IP packet
ip.flags	3	Flags of the IP protocol header
ip.fragment	3	Fragmentation of the packet
ip.dst	3	ip-address of the destination
ip.src	3	ip-address of the source
ip.proto	3	Protocol ID of the upper layer
ip.tos	3	Type of service
udp.srcport	4	Destination port of the UDP datagram
udp.dstport	4	Source port of the UDP datagram

B. Machine learning algorithm

The first step of our analysis was a unsupervised clustering of the data to examine possible differences in the network packets. We used *kMeans* as the clustering algorithm [35] with an unevenly number of packets. As shown in Figure 7, the feature set facilitates a clustering, which requires a deeper look into the captured packets.

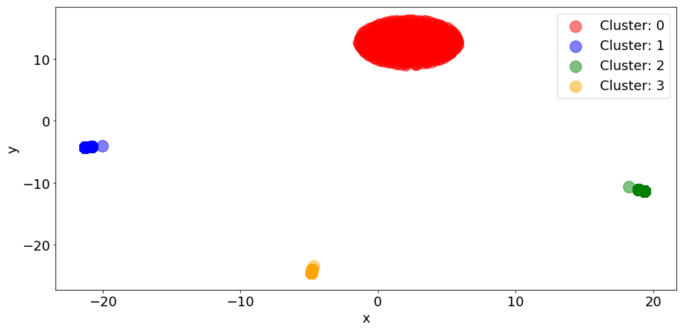


Fig. 7. Clustering with kMeans.

The clusters are created by the different protocols (VXLAN, GENEVE, GRE) as well as internal changes of the network addressing scheme as listed in Table II. The different size of the areas depends on the number of packets assigned to this cluster. Because of the testbed structure the number of packets assigned to the different protocols varies, so the sizes of the coloured areas vary, too.

To create a data-set for training, we separated the packets according to their differences (e. g. the virtual network protocol) as shown in Table I. Anomaly detection with ML is part of supervised learning and can be done by a classification of the different values [36]. The algorithms use different methods to calculate the anomaly score of the different changes. Typical algorithm are *Decision Trees*, *Support Vector Machine*, *k-nearest Neighbors* and *probabilistic methods* [10], [37]. We use the *IsolationForest*-algorithm (IF) [38], which defines anomalies as

> *. . . 'few and different', which make them more susceptible to isolation than normal points.*

IF algorithms as a branch of *Random Forests* detect outliers by randomly selecting features and isolating them by a value between the minimum and the maximum values of this feature.

V. Discussion

Table IV shows the percentage of deviation from a given virtual environment and the different changes. The number of the testrun refers to Table I. The testruns were performed five times and the calculated value is the mean of all values.

The anomaly score is between 57% and 71%. This expresses an anomaly is detected with a prediction of these values. The results cover the real world. To verify our result, we further calculate the anomaly score of a test set containing values of the training set. We assume a score of $\approx 50\%$, because no anomaly should be detectable in this data.

```
>>>pdf = pd.read_csv('./ntraffic.csv')
>>>df=pdf.to_numpy()
>>>clf = IsolationForest(n_estimators=20)
>>>train1=df[1:83]
>>>test1=df[2:20]
>>>clf.fit(train1)
>>>clf.score_samples(test1)
50.96264907259983
```

As shown in this code snipplet, the anomaly score of testdata is $\approx 50\%$, which verifies our results.

The change of the network environment, virtual network protocols as well as the internal network structure creates detectable impact in classification. So each anomaly detection, which uses a similar or parts of our feature set as shown in Table III might be faced with the effect.

VI. Conclusion

Modern networks provide various virtualisation techniques to create a highly flexible and dynamic environment. Whereas this customizability creates benefit for the administrator or the user, IT security processes are hampered when using ML. These processes require a valid basement of data used in the algorithms, but this is not guaranteed in modern networks. The appearance of various changes in the environment might lead to different effects referred to the used algorithms like false positives or false negatives. So to avoid a notable increase in the number of false positive alarms may require additional measures, such as consideration of concept drift [39].

Changes in virtual networks are relevant and might occur as often as needed. As discussed in section IV, the model validation is the step in which the model is evaluated. If an error is detected, all prior steps have to be reconfigured. A change in the network topology might raise such errors, so the complete process has to be re-established. Because of the flexibility inside the environment, such reconfigurations might occur again and again. So an implementation of different machine learning algorithms as part of the datacenter security strategy has to consider this flexibility.

The testbed contains only two systems, which communicate in a clean and isolated environment. This scenario does not match the real world, so in our future research we will analyse the impact of the virtual networks with a higher number of clients running inside these networks. In addition to this our future research will be focused on the analysis of larger network traffic data-sets like the CAIDA Anonymized Internet Traces 2016 Dataset[3] or the KDDCUP99[4] when used inside virtual networks. In addition to this we will analyse different algorithm to verify our results or to develop possible countermeasures to prevent the impact of virtual networks.

Acknowledgements

This work is supported by project SIMARGL (Secure intelligent methods for advanced recognition of malware, stegomalware & information hiding methods, *https://simargl.eu*), which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 833042.

References

[1] Institute of Electrical and Electronics Engineers, "IEEE standard for local and metropolitan area network–bridges and bridged networks," IEEE Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014), 2018.

[2] J. Gross, "GENEVE: Generic network virtualization encapsulation," Internet Engineering Task Force, September 2019. [Online]. Available: http://tools.ietf.org/html/draft-davie-stt-06

[3] P. Garg and Y. Wang, "NVGRE: Network virtualization using generic routing encapsulation," Internet Requests for Comments, RFC 7637, September 2015. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7637.txt

[4] B. Davie and J. Gross, "A stateless transport tunneling protocol for network virtualization (STT)," Internet Engineering Task Force, April 2014. [Online]. Available: http://tools.ietf.org/html/draft-davie-stt-06

[5] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright, "Virtual extensible local area network (VXLAN): A framework for overlaying virtualized layer 2 networks over layer 3 networks," Internet Requests for Comments, RFC 7348, August 2014. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7348.txt

[6] Symantec, "Internet security threat report 24," Symantec, Tech. Rep., 2019. [Online]. Available: https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-2019-en.pdf

[3]Online available at:
https://www.caida.org/data/passive/passive_2016_dataset.xml

[4]Online available at:
http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

TABLE IV
RESULTS OF ANOMALY DETECTION WITH ISOLATIONFOREST IN PERCENTAGE

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	x	63,51	65,81	66,11	66,34	61,17	65,79	66,13	61,23	62,15	62,7	63,33
2	59,64	x	64,91	66,75	63,87	62,28	60,35	60,35	60,77	67,23	67,6	66,98
3	65,34	67,04	x	58,96	57,69	66,38	71,68	69,32	63,71	61,98	59,35	62,84
4	61,22	61,2	59,15	x	61,22	62,71	63,93	60,15	62,07	63,88	64,68	63,22
5	62,8	63,01	62,13	61,01	x	62,53	64,66	67	61,83	62,5	62,13	60,14
6	63,12	67,04	64,13	64,12	62,49	x	63,1	61,28	64,05	63,67	64,21	64,34
7	64,52	64,51	64,63	66,06	64,53	63,67	x	63,49	64,16	64,55	64,56	64,48
8	64,64	64,9	64,55	68,2	64,62	63,13	61,77	x	62,49	64,03	64,53	64,05
9	60,5	60,88	60,79	60,33	60,24	63,13	62,92	61,49	x	60,81	61,6	58,98
10	64,26	65	61,37	63,44	65,07	64,21	61,54	62,18	62,16	x	62,81	64,33
11	60,39	60,77	65,05	62,83	60,4	60,64	60,36	60,71	60,54	61,95	x	61,92
12	60,5	61,44	60,98	63,11	60,68	60,95	60,05	61,32	59,7	57,12	61,65	x

[7] C. Cimpanu, "Ransomware attack hits major us data center provider," Online, Dec. 2019. [Online]. Available: https://www.zdnet.com/article/ransomware-attack-hits-major-us-data-center-provider/

[8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[9] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, 2013.

[10] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," in *Emerging Artificial Intelligence Applications in Computer Engineering*. IOS Press, 2007, pp. 3–24.

[11] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10 206–10 222, 2009.

[12] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for internet of things," *Future Generation Computer Systems*, vol. 82, pp. 761–768, 2018.

[13] T. Sohn, J. Seo, and J. Moon, "A study on the covert channel detection of tcp/ip header using support vector machine," in *International Conference on Information and Communications Security*. Springer, 2003, pp. 313–324.

[14] W. Mazurczyk and L. Caviglione, "Information hiding as a challenge for malware detection," *IEEE Secur. Priv.*, vol. 13, no. 2, pp. 89–93, 2015.

[15] E. S. Pilli, R. C. Joshi, and R. Niyogi, "Network forensic frameworks: Survey and research challenges," *Digital Investigation*, vol. 7, no. 1-2, pp. 14–27, 2010.

[16] S. Garfinkel, "Network forensics: Tapping the internet," *IEEE Internet Computing*, vol. 6, pp. 60–66, 2002.

[17] D. Spiekermann and T. Eggendorfer, "Challenges of network forensic investigation in virtual networks," *Journal of Cyber Security and Mobility*, vol. 5, no. 2, pp. 15–46, 2016.

[18] A. H. R. Al Awadi, "Dual-layer sdn model for deploying and securing network forensic in distributed data center," *Current Journal of Applied Science and Technology*, pp. 1–11, 2017.

[19] M. T. Nashnosh, A. E. Abuhamra, M. M. Alkabiir, T. A. Shaladi, M. M. Abdunnabi, and H. M. Hamedan, "Design and implementation of a forensic logger for software defined networks," in *International Conference on Technical Sciences (ICST2019)*, vol. 6, 2019, p. 04.

[20] D. Spiekermann, J. Keller, and T. Eggendorfer, "Improving lawful interception in virtual datacenters," in *Central European Cybersecurity Conference 2018*. ACM, 2018, p. 8.

[21] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, *Network traffic anomaly detection and prevention: concepts, techniques, and tools*. Springer, 2017.

[22] L. D. Merkle, "Automated network forensics," in *10th Annual Conference Companion on Genetic and Evolutionary Computation*. ACM, 2008, pp. 1929–1932.

[23] J. Zhang, H. Li, Q. Gao, H. Wang, and Y. Luo, "Detecting anomalies from big network traffic data using an adaptive detection approach," *Information Sciences*, vol. 318, pp. 91–110, 2015.

[24] M. Thottan and C. Ji, "Anomaly detection in IP networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2191–2204, 2003.

[25] M. A. Lopez, D. M. Ferrazani Mattos, and O. C. M. B. Duarte, "An elastic intrusion detection system for software networks," *Annals of Telecommunications*, vol. 71, no. 11, pp. 595–605, Dec 2016. [Online]. Available: https://doi.org/10.1007/s12243-016-0506-y

[26] T. Shon, Y. Kim, C. Lee, and J. Moon, "A machine learning framework for network anomaly detection using SVM and GA," in *6th annual IEEE SMC Information Assurance Workshop*. IEEE, 2005, pp. 176–183.

[27] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection," in *Applications of data mining in computer security*. Springer, 2002, pp. 77–101.

[28] D. Spiekermann, J. Keller, and T. Eggendorfer, "Network forensic investigation in openflow networks with forcon," *Digital Investigation*, vol. 20, pp. S66–S74, 2017.

[29] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, "Machine learning for networking: Workflow, advances and opportunities," *IEEE Network*, vol. 32, no. 2, pp. 92–99, March 2018.

[30] D. Spiekermann and T. Eggendorfer, "Towards digital investigation in virtual networks: A study of challenges and open problems," in *11th International Conference on Availability, Reliability and Security (ARES)*, Aug 2016, pp. 406–413.

[31] Y. Rekhter, R. G. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "Address allocation for private internets," Internet Requests for Comments, BCP 5, February 1996. [Online]. Available: https://tools.ietf.org/html/rfc1918

[32] M. Suresh and R. Anitha, "Evaluating machine learning algorithms for detecting ddos attacks," in *International Conference on Network Security and Applications*. Springer, 2011, pp. 441–452.

[33] S. Zhao, M. Chandrashekar, Y. Lee, and D. Medhi, "Real-time network anomaly detection system using machine learning," in *11th International Conference on the Design of Reliable Communication Networks (DRCN)*. IEEE, 2015, pp. 267–270.

[34] M. A. Elsadig and Y. A. Fadlalla, "Packet length covert channel: A detection scheme," in *1st International Conference on Computer Applications Information Security (ICCAIS)*, April 2018, pp. 1–7.

[35] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in *18th International Conference on Machine Learning (ICML 2001)*. Morgan Kaufmann, 2001, pp. 577–584.

[36] T. Lane and C. E. Brodley, "An application of machine learning to anomaly detection," in *20th National Information Systems Security Conference*, vol. 377. Baltimore, USA, 1997, pp. 366–380.

[37] C. C. Aggarwal, *Data classification: algorithms and applications*. CRC Press, 2014.

[38] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *8th IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[39] M. Choras and M. Wozniak, "Concept drift analysis for improving anomaly detection systems in cybersecurity," in *Central European Cybersecurity Conference, CECC*, 2017, pp. 35–42.