Challenges of Network Traffic Classification Using Deep Learning in Virtual Networks

Daniel Spiekermann[1], Jörg Keller[2]

Abstract: The increasing number of network-based attacks like denial-of-service and ransomware have become a serious threat in nowadays digital infrastructures. Therefore, the monitoring of network communications and the classification of network packets is a critical process when protecting the environment. Modern techniques like deep learning aim to help the providers when detecting anomalies or attacks by learning details extracted from a network packet or a flow of packets. Most of these models are trained in networks without any kind of virtualisation, especially network virtualisation overlay environments are not investigated in detail. In this paper, we analyse the classification rate of a Convolutional Neural Network (CNN) faced with encapsulated packets. We evaluate this approach with a proof-of-concept based on a binary classification of a self-curated data-set.

Keywords: virtual networks, network virtualisation overlay, deep learning, neural networks, network traffic classification

1 Introduction

The use of network virtualisation overlay (NVO) is a common technique in modern network infrastructures. A static network design does not provide the required flexibility for modern environments, which demand for dynamic and ubiquitous network access, high speed packet transfers, east-west traffic, customizability and secure tenant isolation [BAM10]. With the evolution of dynamic overlay protocols, limitations of traditional networks were eradicated and the flexibility inside the network as well as the automation of the environment increases. Overlay protocols like VXLAN, GENEVE or NVGRE encapsulate the original network packet by preceding one or more additional header information. Overlay protocols increase the possibilities inside the network, but add complexity for involved devices like switches, routers and firewalls when analyzing network packets because of the additional layers. In addition to this, security staff, digital forensic investigators or support teams are faced with a higher complexity when capturing and analysing the transferred network protocols, i. e. for troubleshooting, malware detection or law enforcement.

Typically, a provider uses only one of these protocols in the environment, but the use of different protocols simultaneously or the combination of protocols at the same time is possible. Using one or more of these encapsulating protocols results in changed network

[1] Polizeiakademie Niedersachsen, Germany daniel.spiekermann@polizei.niedersachsen.de
[2] FernUniversität in Hagen, Germany joerg.keller@fernuni-hagen.de

doi:10.18420/inf2022_08

packets inside the environment. Whereas network packets inside specific network areas (e. g. a tenant network) remain without any overlay protocol, packets transferred inside the backbone network might be changed depending on the path they are traversing. Figure 1 shows an exemplary network with only two different tenants and three network segments.

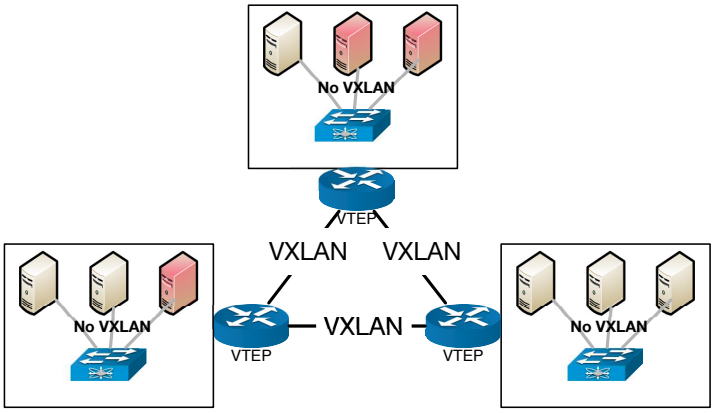


Fig. 1: NVO usage in networks, machines of different tenants are marked in different colours. Each rectancle represents a single subnet (e. g. a rack or compute node), the VTEPs manage the communication of nodes of the same color.

Inside a tenant network, e. g. the upper square with two red VMs, no VXLAN encapsulation is needed, but if the red VM in the left square is connected, each network packet is encapsulated if sent to this VM. Depending on the position of capturing, this protocol change effects all types of network classification, intrusion detection or security devices like firewalls and application layer gateways.

Providers of modern networks use traffic classification techniques to get insights about communication inside the environment. Static approaches like deep-packet inspection as well as dynamic techniques like machine learning or deep learning are used to increase the security of the infrastructure. Anomalies, outliers, adversarial attacks and unwanted traffic can be classified for further processing. So, the knowledge of network traffic inside an environment is crucial for every modern infrastructure.

Prior implementations mostly focus on a static ruleset to classify benign or malicious network traffic. As shown in [SK20], a shift from a static to a dynamic network infrastructure might result in more complex techniques, so static approaches are now insufficient.

With the evolution of machine learning approaches, the possibilities of network traffic classification, anomaly detection and the overall network-security increases. Different algorithms provide improved techniques to eradicate common issues like the heterogeneous traffic or user-customized network infrastructures. But these techniques still have issues

when faced with virtualised network traffic, which demands for new or adapted training sets to avoid inappropriate detection and protection results. As an example, [SK21b] details, that different algorithms like IsolationForest have fail in the detection of anomalies when faced with virtualised traffic. Using deep learning algorithms further enlarge the possibilities of artificial intelligence. But similar to machine learning algorithms, most of these applications do not cover the aforementioned virtual environments.

Deep learning techniques use multiple layers in an artificial network to provide various applications like image classification, speech recognition, recommendation systems or fraud detection, which gain an advantage over human or static implementations [GBC16]. In network environments, techniques like Convolutional Neural Networks (CNN) or Generative Adversarial Network (GAN) are used to improve the security [SSE19] of intrusion detection systems [Ta16], detect malware [Vi19] or spam [He17] and classify the network traffic [XLJ21].

An improved detection rate in network classification is reported in the literature [Lo20], but most of the research is focused on plain or encrypted network traffic. In this paper, we analyze the application of a CNN when classifying NVO traffic. As done in [Li19] for networks without virtualization, we convert network packets to pictures and train our CNN with these information. The detection rate of $\geq 99\%$ is similar to approaches like [Li19, Wa18, MMS19]. However, the analysis of the same network traffic encapsulated with different NVO protocols shows a reduced detection rate depending on the protocol.

The main contribution of our research are:

- We identify the role of different network virtualization protocols on anomaly detection algorithms.
- We explore the difficulties that deep learning algorithms encounter when being applied on virtual instead of physical networks.
- We provide a proof-of-concept to support our hypotheses with quantitative experiments. We provide a self-curated data-set for these experiments which we plan to make public.

The remainder of this article is structured as follows. Section 2 discusses related work regarding network traffic classification and virtual networks. In Section 3 we describe the initial classification process with a CNN. The trained model is used in Section 4 to determine the detection rate with encapsulated network packets. Section 6 concludes this paper and gives an outlook to our future research.

2 Related Work

Network traffic classification is a well-known and deeply researched area in all fields of information security like anomaly detection [Ma03] and attack prevention [Fe03] as well as network management and monitoring [Ts18]. [AKO15] gives an overview of different traffic classification techniques. Static techniques like port-based, payload-based algorithms or deep-packet inspection are discussed in [Ac10].

The use of deep learning as discussed in [Li19]. [Wa18] shows the possibilities of a CNN and residual network (ResNet) to classify network traffic, even when the traffic is encrypted.

NVO changes the internal structure of network infrastructures, which might result in critical effects for the detection algorithms [SK20]. The impact of an unsupervised packet based approach is investigated in [SK21c].

3 Deep learning classification

The use of deep learning is a common technique to classify network traffic. Most of the research is done without any dynamic encapsulation protocol [SSJ21, Ac19] but on plain network protocols. In addition to this, deep learning is able to classify network traffic even when it is encrypted or non-plaintext [Ac18]. By this, anomalous network traffic and malware might be detected and discarded or the knowledge of transferred network packets helps to improve the performance in a network.

To evaluate the impact of a network change to a CNN, [Ze19] proposes the following steps: *Packet Generation, Traffic Purification, Traffic Refiner, Length Unification* and *Packet conversion*. To generate network packets, we created a virtual network environment, which facilitates network packet captures at various positions. Inside this network, we focus on the transmission of ICMP (Internet Control Message Protocol) packets. These packets differ from a huge number of common network packets like TCP or UDP traffic which simplifies the initial process of network packet classification. So, we were able to improve the analysis of the classification process. The traffic purification and refinement was done by sanitizing the network traffic to remove unwanted network traffic like ARP requests or other broadcast packets. After this, our capture files contain only wanted network packets. Our first capture file contains 14,072 ICMP messages with different types like ICMP request (type 0) and response (type 8) or destination unreachable (type 3) [Po81]. The next step was the transformation of every single packet to a picture representing this packet. We use an adaptation of the Python module file2image[3], which ensures the correct transformation of the packet to a picture in png-format. Every network packet was scaled to 128×128 pixels, a result is shown in Figure 2.

We use Tensorflow and Keras[4] to create a CNN for image classification. We use the following

[3] Details can be found at `https://pypi.org/project/file2image/`
[4] Details can be found at `https://keras.io/about/`

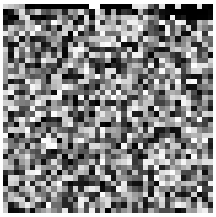


Fig. 2: Image of a network packet generated with file2image and scaled to 128×128 pixels

layers derived from [KJ18, SHA17] to create our model, which is trained with 50 epochs, but without any special optimisations: *Conv2D, MaxPooling2D, Activation, Dense, Add, Dropout* and *BatchNormalization*.

Our binary classification is used to detect ICMP packets in a number of different network packets and has a validation accuracy of 99%.

After this we analyse the model with a second data-set containing various network packets and protocols like HTTP, RTP, SSH and ICMP, but no NVO traffic. This data-set M comprises 2,761,441 network packets, the subset of ICMP packets is denoted by $I \subset M$. By enumerating each picture we are able to verify the prediction for each network protocol more easily.

Our model analyses each network protocol and predicts the classification rate, i.e. the probability p_i that packet i an ICMP packet. The following excerpt exemplifies the output of the analysis.

```
684.png is 99.97 percent NO ICMP and 0.03 percent ICMP.
685.png is 0.03 percent NO ICMP and 99.97 percent ICMP.
686.png is 99.99 percent NO ICMP and 0.01 percent ICMP.
687.png is 0.02 percent NO ICMP and 99.98 percent ICMP.
```

We use *tshark* to validate this results. Packets 685 and 688 are correctly classified as ICMP packets by the model, the other network packets are correctly classified as non-ICMP, because they are RTP, UDP and TCP packets. The following snipped illustrates how the predictions are validated.

```
tshark -r capture.pcap   -T fields -e frame.number -e_ws.col.Protocol
684 RTP
685 ICMP
686 UDP
687 ICMP
```

This demonstrates the correctness of our model, which has an average detection rate $\sum_{i \in I} p_i / |I|$ of 99.98%.

4 Packet transformation

This section describes the process of creating virtual network packets and the classification of these adapted packets with the already trained model.

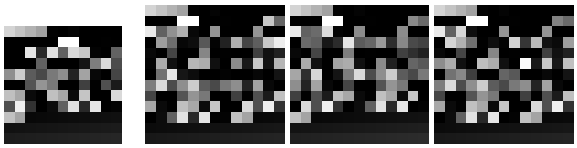
The shift of network packets to virtual packets is done with *Encapcap* [SK21a]. This tool helps to create encapsulated network packets from plain network packets by adding necessary header information and is able to fill randomized, but still plausible values into the new headers of these network packets, which simulates virtual behaviour like protocol swapping or changes of the IP addressing scheme. We transformed different network captures containing network packets with protocols (HTTP, HTTPS, ICMP, RTP, SSH and QUIC) with *Encapcap* to data-sets with encapsulating network protocols. We use VXLAN, NVGRE and GENEVE as NVO protocols.

- **VXLAN** The most notable protocol to implement NVO is virtual eXtensible LAN (VXLAN) [Ma14], which is similar to the well-known VLAN protocol, but expands its features and increases the number of possible virtual separated subnets to 16,777,216 networks. VXLAN uses a 24-bit header field named virtual network identifier (VNI) to isolate the different networks, the encapsulation is done with the User Datagram Protocol (UDP).
- **NVGRE** Network Virtualization using Generic Routing Encapsulation (NVGRE) is defined in RFC 7637 [GW15]. It bases on GRE as the encapsulating protocol and uses parts of the packet header to manage and control the network separation. NVGRE uses 24 bit of the optional fields to add a Virtual Subnet ID (VSID).
- **GENEVE** Generic Network Virtualization Encapsulation (GENEVE) is defined in [GGS20] and uses a 24-bit header field named Virtual Network Identifier (VNI). In addition to this, GENEVE provides huge flexibility for the encapsulation of different network packets. GENEVE uses a small fixed header followed by variable-length option fields used for the transmission of specific information like identifiers. Similar to VXLAN, GENEVE uses UDP as the transport protocol.

All implementations use a 24-bit header field for network isolation, but in combination with various header fields a resulting network packet differs after the encapsulation process depending on the used virtualization protocol.

We created one capture file for each aforementioned virtualization protocol. Each capture file contains the correct number of network packets. As described in Section 3, each network packet of each capture file is transformed to an image. Due to different protocols and their

type of encapsulation, different pictures result from this process. Figure 3 shows the different pictures.



(a) Plain (b) VXLAN (c) NVGRE (d) GENEVE

Fig. 3: Pictures of network packet before and after transformation

The transformed packets have a bigger size because the creation process takes care of the packet length and therefore creates pictures with a size depending on the packet size, and thus depending on the virtualization protocol. To ensure compatibility, we scale each image to 128×128 pixels.

5 Discussion

We use our trained model to classify each network packet i in the capture file M_P (the subset of ICMP packets is denoted as I_P) for virtualization protocol P either as ICMP or as non-ICMP, i.e. for each packet i we get probability p'_i of being an ICMP packet. Table 1 shows the average $\sum_{i \in I_P} p'_i / |I_P|$, best $\max_{i \in I_P} p'_i$ and worst $\min_{i \in I_P} p'_i$ detection rates for each protocol P. For comparison, we repeat that the average detection rate of plain network packets was 99,98%, as listed in Section 3.

Tab. 1: Detection rate

Protocol	**Average**	**Best**	**Worst**
VXLAN	73.14%	99,99%	14,48%
GENEVE	82.21%	99,99%	29,99%
NVGRE	99.94%	99,99%	99,97%

Figure 4 shows the worst, average and best prediction rate for each protocol. The colored rectangle marks the upper and lower quartile of the prediction.

Whereas the predicted detection rate of our trained model for VXLAN and GENEVE spread widely, the prediction rate of NVGRE remains nearly to the prediction rate of plain traffic. This can be explained by investigating the encapsulation structure in each protocol. VXLAN, as well as GENEVE, adds UDP and as a second addition, a protocol dependent header in front of the original network packet. Each UDP packet contains a checksum, which varies for every packet. This additional information results in a more different network protocol, which confuses the trained model, and leads to the reduced prediction rate.

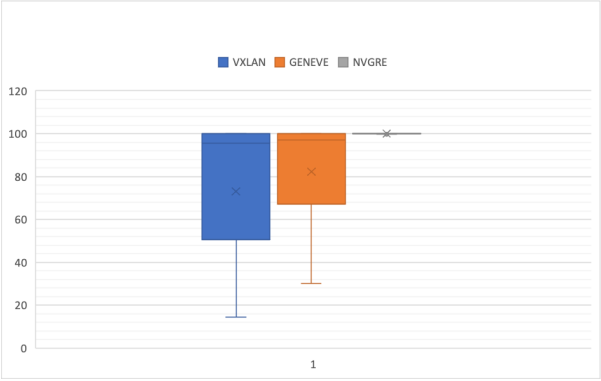


Fig. 4: Prediction distribution per protocol.

6 Conclusions

The use of deep learning algorithms provides an improved classification of network packets and detection of malicious or unwanted network traffic. But most of the research focus on plain network traffic without any encapsulation, and are therefore limited in their significance for modern and virtual environments. In this paper, we analyzed a model with a high detection rate of 99,98% when classifying non-encapsulated network traffic. Using this model for traffic classification of encapsulated network packets, the detection rate decreases to 70 to 80%. As a result, security implementations in modern networks using NVO have to consider the high dynamic and possible protocol changes inside the environment to reach or steady the needed detection rate.

The use of ICMP messages does not completely cover the real world, but it shows the risks of trusting deep learning models in highly dynamic networks. We use a static approach of network packet transformation, but virtual networks in operation include more possible shifts and different complex scenarios like user-customized networks. By this, customers are able to create their own networks in a provider environment, thus the provider does not have knowledge of these changes. As a result, the provider is unable to react timely. Our future research will focus on more real-world scenarios, supported by the analysis of well-known data-sets like CAIDA Anonymized Internet Traces 2016 Dataset[5] or KDDCUP99[6], either with or without encapsulated protocols.

Bibliography

[Ac10] Aceto, Giuseppe; Dainotti, Alberto; De Donato, Walter; Pescapé, Antonio: PortLoad: taking the best of two worlds in traffic classification. In: 2010 INFOCOM IEEE Conference on Computer Communications Workshops. IEEE, New York, NY, pp. 1–5, 2010.

[5] Online available at: `https://www.caida.org/data/passive/passive2016dataset.xml`
[6] Online available at: `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`

[Ac18] Aceto, Giuseppe; Ciuonzo, Domenico; Montieri, Antonio; Pescapé, Antonio: Mobile encrypted traffic classification using deep learning. In: 2018 Network traffic measurement and analysis conference (TMA). IEEE, New York, NY, pp. 1–8, 2018.

[Ac19] Aceto, Giuseppe; Ciuonzo, Domenico; Montieri, Antonio; Pescapè, Antonio: MIMETIC: Mobile encrypted traffic classification using multimodal deep learning. Computer Networks, 165:106944, 2019.

[AKO15] Al Khater, Noora; Overill, Richard E: Network traffic classification techniques and challenges. In: 2015 Tenth international conference on digital information management (ICDIM). IEEE, New York, NY, pp. 43–48, 2015.

[BAM10] Benson, Theophilus; Akella, Aditya; Maltz, David A: Network traffic characteristics of data centers in the wild. In: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. pp. 267–280, 2010.

[Fe03] Feinstein, Laura; Schnackenberg, Dan; Balupari, Ravindra; Kindred, Darrell: Statistical approaches to DDoS attack detection and response. In: Proceedings DARPA information survivability conference and exposition. volume 1, IEEE, New York, NY, pp. 303–314, 2003.

[GBC16] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron: Deep learning. MIT Press, Cambridge, MA, 2016.

[GGS20] Gross, Jesse; Ganga, Ilango; Sridhar, T.: , Geneve: Generic Network Virtualization Encapsulation. RFC 8926, November 2020.

[GW15] Garg, Pankaj; Wang, Yu-Shun: , NVGRE: Network Virtualization Using Generic Routing Encapsulation. RFC 7637, September 2015.

[He17] He, Hongmei; Watson, Tim; Maple, Carsten; Mehnen, Jörn; Tiwari, Ashutosh: A new semantic attribute deep learning with a linguistic attribute hierarchy for spam detection. In: 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, New York, NY, pp. 3862–3869, 2017.

[KJ18] Kannojia, Suresh Prasad; Jaiswal, Gaurav: Effects of varying resolution on performance of CNN based image classification: An experimental study. Int. J. Comput. Sci. Eng, 6(9):451–456, 2018.

[Li19] Lim, Hyun-Kyo; Kim, Ju-Bong; Heo, Joo-Seong; Kim, Kwihoon; Hong, Yong-Geun; Han, Youn-Hee: Packet-based network traffic classification using deep learning. In: 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC). IEEE, New York, NY, pp. 046–051, 2019.

[Lo20] Lotfollahi, Mohammad; Jafari Siavoshani, Mahdi; Shirali Hossein Zade, Ramin; Saberian, Mohammdsadegh: Deep packet: A novel approach for encrypted traffic classification using deep learning. Soft Computing, 24(3):1999–2012, 2020.

[Ma03] Mahoney, Matthew V: Network traffic anomaly detection based on packet bytes. In: Proceedings of the 2003 ACM symposium on Applied computing. ACM, New York, NY, pp. 346–350, 2003.

[Ma14] Mahalingam, Mallik; Dutt, Dinesh; Duda, Kenneth; Agarwal, Puneet; Kreeger, Larry; Sridhar, T.; Bursell, Mike; Wright, Chris: , Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. RFC 7348, August 2014.

[MMS19] Mohammed, AysŞe Rumeysa; Mohammed, Shady A; Shirmohammadi, Shervin: Machine learning and deep learning based traffic classification and prediction in software defined networking. In: 2019 IEEE International Symposium on Measurements & Networking (M&N). IEEE, New York, NY, pp. 1–6, 2019.

[Po81] Postel, J.: Internet Control Message Protocol. STD 5, RFC Editor, September 1981. http://www.rfc-editor.org/rfc/rfc792.txt.

[SHA17] Shiddieqy, Hasbi Ash; Hariadi, Farkhad Ihsan; Adiono, Trio: Implementation of deep-learning based image classification on single board computer. In: 2017 International Symposium on Electronics and Smart Devices (ISESD). IEEE, pp. 133–137, 2017.

[SK20] Spiekermann, Daniel; Keller, Jörg: Impact of virtual networks on anomaly detection with machine learning. In: 2020 6th IEEE Conference on Network Softwarization (NetSoft). IEEE, New York, NY, pp. 430–436, 2020.

[SK21a] Spiekermann, Daniel; Keller, Jörg: Encapcap: Transforming Network Traces to Virtual Networks. In: 2021 IEEE 7th International Conference on Network Softwarization (NetSoft). IEEE, New York, NY, pp. 437–442, 2021.

[SK21b] Spiekermann, Daniel; Keller, Jörg: Unsupervised packet-based anomaly detection in virtual networks. Computer Networks, 192:108017, 2021.

[SK21c] Spiekermann, Daniel; Keller, Jörg: Unsupervised packet-based anomaly detection in virtual networks. Comput. Networks, 192:108017, 2021.

[SSE19] Sagduyu, Yalin E; Shi, Yi; Erpek, Tugba: IoT network security from the perspective of adversarial deep learning. In: 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). IEEE, New York, NY, pp. 1–9, 2019.

[SSJ21] Sadeghzadeh, Amir Mahdi; Shiravi, Saeed; Jalili, Rasool: Adversarial Network Traffic: Towards Evaluating the Robustness of Deep-Learning-Based Network Traffic Classification. IEEE Transactions on Network and Service Management, 18(2):1962–1976, 2021.

[Ta16] Tang, Tuan A; Mhamdi, Lotfi; McLernon, Des; Zaidi, Syed Ali Raza; Ghogho, Mounir: Deep learning approach for network intrusion detection in software defined networking. In: 2016 international conference on wireless networks and mobile communications (WINCOM). IEEE, 2016, pp. 258–263, 2016.

[Ts18] Tsai, Pang-Wei; Tsai, Chun-Wei; Hsu, Chia-Wei; Yang, Chu-Sing: Network monitoring in software-defined networking: A review. IEEE Systems Journal, 12(4):3958–3969, 2018.

[Vi19] Vinayakumar, R; Alazab, Mamoun; Soman, KP; Poornachandran, Prabaharan; Venkatraman, Sitalakshmi: Robust intelligent malware detection using deep learning. IEEE Access, 7:46717–46738, 2019.

[Wa18] Wang, Pan; Ye, Feng; Chen, Xuejiao; Qian, Yi: Datanet: Deep learning based encrypted network traffic classification in sdn home gateway. IEEE Access, 6:55380–55391, 2018.

[XLJ21] Xie, Guorui; Li, Qing; Jiang, Yong: Self-attentive deep learning method for online traffic classification and its interpretability. Computer Networks, 196:108267, 2021.

[Ze19] Zeng, Yi; Gu, Huaxi; Wei, Wenting; Guo, Yantao: $Deep-Full-Range$: a deep learning based network encrypted traffic classification and intrusion detection framework. IEEE Access, 7:45182–45190, 2019.