VizNN: Visual Data Augmentation with Convolutional Neural Networks for Cybersecurity Investigation

Amélie Raymond[1], Baptiste Brument[1], and Pierre Parrend[2,3]

[1] Télécom-Physique – University of Strasbourg, France
amelie.raymond@etu.unistra.fr, baptiste.brument@etu.unistra.fr
[2] EPITA
pierre.parrend@epita.fr
[3] ICube laboratory
University of Strasbourg

Abstract. One of the key challenges of Security Operating Centers (SOCs) is to provide rich information to the security analyst to ease the investigation phase in front of a cyberattack. This requires the combination of supervision with detection capabilities. Supervision enables the security analysts to gain an overview on the security state of the information system under protection. Detection uses advanced algorithms to extract suspicious events from the huge amount of traces produced by the system. To enable coupling an efficient supervision with performance detection, the use of visualisation-based analysis is a appealing approach, which into the bargain provides an elegant solution for data augmentation and thus improved detection performance. We propose VizNN, a Convolutional Neural Networks for analysing trace features through their graphical representation. VizNN enables to gain a visual overview of the traces of interests, and Convolutional Neural Networks leverage a scalability capability. An evaluation of the proposed scheme is performed against reference classifiers for detecting attacks, XGBoost and Random Forests.

Keywords: Data augmentation, Visualisation, Neural Network, Benchmark, Cybersecurity, Investigation

1 Introduction

Security Operating Centre (SOCs) continuously experience increasing amounts of supervision data, that require scalable processing capability coupled with fine-grained detection of known and unknown (zero-days) attacks. New solutions are thus required to back the investigation efforts of security analyst teams. The core requirements of these solutions are: explicability and traceability of alert to the original trace; investigation support through visualisation; scalability of detection.

The availability of scalable detection algorithms is thus key for building efficient SOCs capable of handling the data deluge. To this aim, neural networks such as Convolutional Neural Networks [1] are very competitive candidates, but still lack of maturity for rapid operational deployment. One of the key challenges for using neural networks in operation environment is the capability of performing suitable data augmentation in the analysis flow [2, 3]. Data augmentation can serve several goals: completing the data representation is areas where available information do not support satisfactory learning (scarce zones of valid data, or imbalanced dataset); altering the information to avoid overfitting and anticipate random modifications of the observed behaviours; changing the data format, for instance to apply algorithms with specific data input such as images to other data types such as text, sequences or qualitative features.

In this paper, we propose VizNN, a visual data augmentation model with Convolutional Neural Networks (CNN) for cybersecurity investigation. Its goal is to leverage scalability and detection capability of CNNs to the analysis of quantitative logs for detection of known attacks. Alternative detection solutions such as Random Forests [4] and XGBoost [5] also provide excellent results for medium size dataset. They are used as reference for benchmarking.

To evaluate the relevance and efficiency of the proposed approach, we apply it to the analysis of security properties of the DoH protocol [6]. DoH (DNS-Over-HTTPS) aims at encrypting DNS requests by encapsulating them in a classic HTTPS stream. Its goal is to solve the vulnerabilities of DNS (Domain Name Systems) which is used for each and every request on the Internet to bind human-readable domain names with machine-readable IP addresses. DNS experiences several majors weaknesses in its default configuration [7, 8], which urge the community to propose adequate countermeasures.

The evaluation of VizNN scheme is performed on CIRA-CIC-DoHBrw-2020[4] dataset [9], created by the Canadian Institute of Cybersecurity (CIC) and funded by the Canadian Internet Registration Authority (CIRA). Data is collected from the top 10,000 visited websites according to Alexa rankings. The raw data, in PCAP format, was processed and converted into CSV files using the DOHMeter tool.

The paper is organised as follows. Section 2 introduces the related works. Section 3 defines the data processing methodology, section 4 details the data preparation and exploration phase, whereas section 5 defines the propose data anaysis scheme. Section 6 evaluates the results. Section 7 concludes this work.

2 Related Work

2.1 Neural networks for cybersecurity

The advent of Deep Learning (DL) for the analysis of cybersecurity events takes its roots in the limitation of pre-existing Machine Learning (ML) algorithms. Machine Learning is used mainly for classification or regression, and keeps relying on feature engineering [10]. The expected advantages of Deep Learning is its capacity to perform well when data amount increases, and to provide very efficient test operation even though training phase is usually significantly longer. One of the key factors for the performance of Deep Learning is the numerous matrix operations it relies on, which let it be preferably executed over GPU.

Deep Learning comes in numerous flavours when applied to cybersecurity: Deep Neural Networks, which extracts hidden patterns from in the internal layers of the network, Recurrent Neural Networks such as LSTM (Long Short Term Memory) [11] which retain the memory of previous states, Convolutional Neural Networks [1, 12] to process data having a high degree of similarity, Restricted Boltzmann Machines for data generation or classification, Deep Belief Networks as brick of larger neural nets, Deep auto-encoders for trace de-noising or classification [13, 14].

[4] https://www.unb.ca/cic/datasets/dohbrw-2020.html

2.2 Data augmentation for neural network learning

Data augmentation is the process of artificially enriching real data with synthetic mock-ups to improve learning, in particular to remove learning bias [2]. One typical application of data augmentation is the densification of data zones where events are legit but scarce to avoid the generation of false negatives. The principle of data augmentation originally referred to statistical data enhancement like SMOTE [15], which is meant for handling imbalanced data. It increasingly refers now to the generation and enrichment of images to be analysed, in particular in the context of Convolutional Neural Networks. The objective is to leverage the capability of data volume scaling of deep learning approaches. It proves to be efficient for the detection of object landmarks such as invariance in shape, pose and illumination [16].

Image augmentation can be performed through basic image manipulations such as geometric transformations, flipping, colouring, cropping, noise injection, through geometric and photometric transformations such as kernel filters, mixing images, random erasing, or through deep-learning such as feature space augmentation, adversarial training, Generative Adversarial Networks or neural style transfer [2]. The combination of these operations, especially the simple ones, can be performed automatically to improve the validation accuracy like with AutoAugment tool [3]. The first image augmentation tool can be considered to be the auto-encoder of Hinton, which is applied for dimension reduction, noise reduction, data and image generation bases on multi-layer architecture with internal small-dimension layers [17]. More recent solutions focus on Generative Adversarial Networks (GAN) for data and image generation, which base on a pair of Deconvolutional/Convolutional Neural Networks [18]. This approach has made radical qualitative progress since its inception in 2014 [19].

2.3 Data augmentation for cybersecurity

Data augmentation for cybersecurity is used both for supervised and for unsupervised detection. It is used for supervised detection of malwares to address the variability of malicious code by adding noise in the training mode under Gaussian, Poisson or Laplace model. Then, a Convolutional Neural Networks performs the learning operation [20].

For unsupervised anomaly detection, the strategy is to oversample normal, rare data which usually causes most false positives. This approach is only applicable to well-defined data distribution if one does not want to inject excessive bias in the dataset. Lim proposes to augment not the input data, but a representative latent vector at the core of the neural network, through multivariate Gaussian sample generation [21]. It integrates an adversarial auto-encoder (AAE), which is an extension of GAN [22].

3 Data processing

Performed data analysis is performed in two steps: first, a generic DAP (Data Analysis Process), which is meant for reuse independently of the analysis methodology; then, VizNN, the visual data augmentation model we propose for Convolutional Neural Networks.

3.1 DAP - the Data Analysis Process

The Data Analysis Process entails following steps:

- Capture: the system behaviour is collected and gathered in a dedicated datalake for immediate analysis or later reference. The behaviour can be represented as actions (logs), system measures (scalar probes) or action measures (scalar measures derived from logs).
- Cleanup: incomplete or inconsistent data is removed, such as action or probe occurrences (data lines) with missing values or features (data columns) with non discriminating values.
- Standardisation: data is normalised or enhanced with metadata, in particular through FAIRification[5] (*i.e* to make it Findable; Accessible; Interoperable; Reusable).
- DE – Data Exploration: a manual scrutiny of the data is performed to highlight its particularities.
- DA – Data Analysis: automated statistical or Machine Learning algorithms are applied to the data in order to perform anomaly detection (in unknown rare events such as an abnormal access to a resource) or classification (discriminate known events such as a known attack).
- Visualisation: the data or extractions thereof is presented to the user.
- Investigation: the expert, here the security analyst, performs computer-assisted complementary inspection of the data to understand the output of the automated analysis and to identify behaviours not characterised by automation.

3.2 VizNN - Visual data augmentation for Convolutional Neural Network

The VizNN pipeline, shown in Figure 1, consists of five steps: data import and clean-up; selection of the features you want to keep; creation of images; preparation of images; training of the CNN model.

Data import and cleanup The import and cleaning steps are similar to those of a Machine Learning project.
After importing the data, missing values are completed or deleted. Categorical features are then converted to numerical ones. A correlation analysis may be relevant to remove unnecessary features. Finally, a ”label” column is added to each row for classification.

Selection of the features you want to keep Rather than converting the entire dataset into images, a feature selection step is performed to only keep the columns that maximize performance.

Therefore, a basic XGBoost model is trained in order to retrieve the list of most important features on the model's gain.
Our proposal is to create images with the N most important features. To create images, this number cannot be a prime number since the width and the height have to be integers. The data is then filtered so that only the N most important features are kept.

Creation of images The filtered DataFrame is converted to a list of grayscale Image objects using PIL library[6].
Images then undergo data augmentation through resizing with bicubic interpolation to generate new pixels and therefore enlarge them.

[5] https://www.go-fair.org/fair-principles/fairification-process/
[6] https://pillow.readthedocs.io/en/stable/reference/Image.html

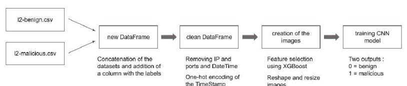
l2-benign.csv
l2-malicious.csv
new DataFrame
clean DataFrame
creation of the images
training CNN model
Concatenation of the datasets and addition of a column with the labels
Removing IP and ports and DateTime
One-hot encoding of the TimeStamp
Feature selection using XGBoost
Reshape and resize images
Two outputs :
0 = benign
1 = malicious

Fig. 1. Full pipeline process

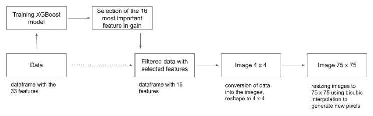
Training XGBoost model
Selection of the 16 most important feature in gain
Data
Filtered data with selected features
Image 4 x 4
Image 75 x 75
dataframe with the 33 features
dataframe with 16 features
conversion of data into the images, reshape to 4 x 4
resizing images to 75 x 75 using bicubic interpolation to generate new pixels

Fig. 2. Image creation process

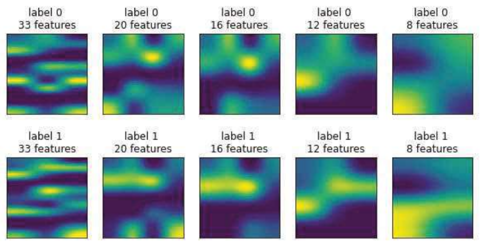

Fig. 3. Examples of augmentated feature images for VizNN

Preparation of images Several operations are performed on the images to transform them into CNN inputs.
First, the list of Image objects is converted to a NumPy array. Second, the data is split into a training and a testing dataset with the ”StratifiedShuffleSplit” method. It shuffles and separates the data while respecting the proportion of each class. Thus, it ensures that even with unbalanced data, models are trained and tested on all classes with representative proportions.
Third, a dimension is added to the image so that it has three dimensions: width, height and number of channels(1 for gray-scale images and 3 for RGB images). Finally, images are normalized to have, for each pixel, values between 0 and 1.

Training of the CNN model The Convolutional Neural Networks is trained using the images generated by the reference dataset.

4 Data Preparation and Exploration

Data exploration is performed by data cleaning and subsequent scrutiny through highly efficient, through very widespread, learning algorithms: Random Forests and XGBoost.

4.1 The dataset

The CIRA-CIC-DoHBrw dataset entails two kind of traffic: HTTPS traffic without DOH and DOH traffic. In case of DOH traffic, attacks and normal traffic were distinguished. Thus, 4 subdatasets are considered: “non DOH”, “DOH”, “benign DOH” and “malicious DOH”. These datasets include 33 features that define the traffic.

This study focuses on the classification of DOH traffic (benign or malicious). The two corresponding CSV files are imported and combined into one DataFrame with an

added column entitled ”label”. Label 0 corresponds to benign DOH traffic and label 1 to malicious DOH traffic.
The dataset is unbalanced: there are 67% of attacks and 23% of benign traffic. Several columns such as ”sourceIP”, ”DestinationPort” were removed since they are artifically crafted for analysis stakes. Afterwards, the ”TimeStamps” column is transformed into categorical data. In this way, four categories are used to describe the period of the day: ”morning”, ”day”, ”evening” and ”night”.

4.2 Random Forests

Random Forests is a classification algorithm based on ensemble learning, which consists of using so-called “weak” models to make predictions and then to combine them into a larger model. It operates by creating a given number of decision trees. More specifically, an improved version of bagging is used in order to reduce the correlation between each tree. The samples of the training data are provided as an input for each tree. The latter then make their prediction. The results are aggregated using a majority rule which means that the final predicted class is the class that was predicted the most by the individual decision tree.

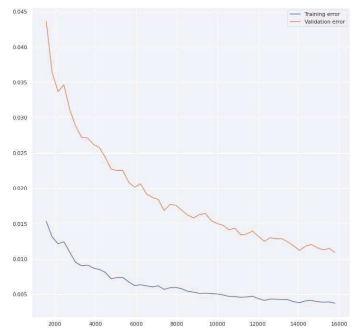

Fig. 4. The learning curves for Random Forests

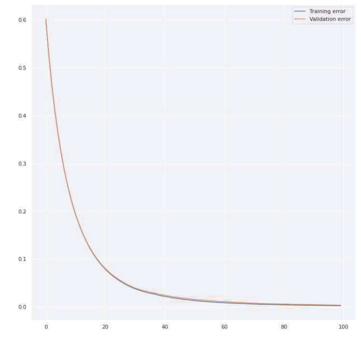

Fig. 5. The learning curves for XGBoost

Figure 4 shows the loss curve for Random Forests learning. Note that the ordinate spans from 0 to 0.045, which makes it a very close zoom. Training curve do not stabilise before 14,000 iterations.

The extraction of feature importance by gain, for the Random Forests algorithm, shows that the main features are statistics derived from the packet length: mode, mean, median and standard deviation. Note that the original dataset entails such aggregated data, and not directly raw communication traces.

4.3 XGBoost

XGBoost stands for Extreme Gradient Boosting [5]. This algorithm is also based on ensemble learning strategy. Gradient boosting is a subclass of boosting algorithms. In boosting, each sub-model is weighted. This allows greater flexibility as it gives the ability to give more importance to certain models in certain learning cases. Thus, this type of models is particularly effective when the training data is not balanced. The specificity of gradient boosting is that the contributions of the decision tree models to the final model are calculated from the optimization of the gradient descent. This technique gives very accurate results. XGBoost is considered one of the best algorithms in the current state of the art classification algorithms.

Figure 5 shows the loss curve for XGBoost learning.

The extraction of feature importance by gain, for the XGBoost algorithm, shows that the main features there are packet length mode, number of flow bytes received, duration, as well as packet length statistics.

5 DA – Data Analysis with VizNN

This section presents the implementation of VizNN and the search for optimal model parameters for the data preparation flow. The Convolutional Neural Networks takes a series of images representing the features to be analysed as input. It is built by a convolutional layer, maxpooling layer, renewed convolutional and maxpooling layers, two flatten and dense neuron layers, and a two-value output layer. RELU activation is used. Each experiment is performed with 50-folds cross-validation, a validation set consisting of 20% of the training data and over 20 epochs. Figure 6 shows this architecture.

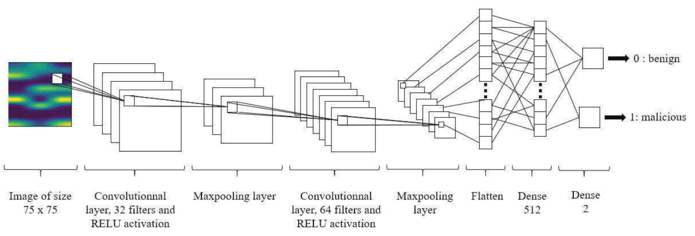
0 : benign
1: malicious
Image of size 75 x 75
Convolutionnal layer, 32 filters and RELU activation
Maxpooling layer
Convolutionnal layer, 64 filters and RELU activation
Maxpooling layer
Flatten
Dense 512
Dense 2

Fig. 6. VizNN architecture

Influence of image size To evaluate the impact of image size on the quality of the detection, we first evaluate the detection performance for square images with side ranging from 50 to 95 pixels. Images are created using all dataset features. Several models with the same architecture are then trained on these images, each model testing a different image size. The results indicate that image size has a significant influence on performance, as shown in Figure 7. According to the scores, 75 × 75 is the size that maximises the performance. Hence, the size is set as 75 × 75 for the rest of the study.

Influence of the number of features Another image-specific parameter that can influence the results is the number of features used to create images. Following the same procedure as for the image size search, several models are trained with images created from different number of features. For the dataset used for the experiment, keeping the 16 (out of 33) most important features from XGBoost's gain provides the best scores, as shown in Figure 8. Also, the model is under-trained when there are not enough features because the training images are not representative enough. On the contrary, if too many features

are kept, the model is over-trained.
For the rest of the study, the number of features is thus set to 16.

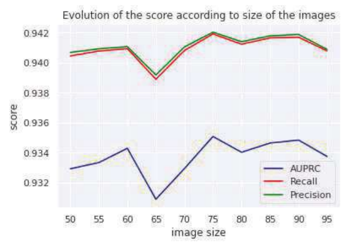

Fig. 7. Performance of VizNN according to the size of the images

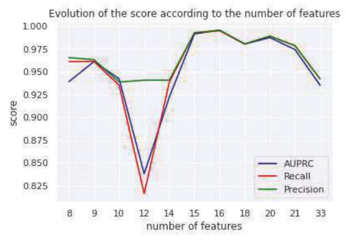

Fig. 8. Performance of VizNN according to the number of features

Influence of the architecture Once the image-related parameters are set, several CNN architectures are tested by changing the number of layers and filters. As shown on Figure 9, an architecture with 32 and 48 for the first and the second convolutional layer respectively provides the best detection capability for the dataset under consideration.

Figure 10 shows the loss for resulting architecture. Learning stabilised after 12 epochs for both training and validation.

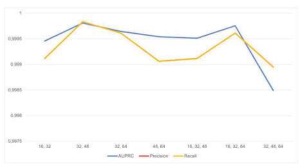

Fig. 9. Impact of CNN architecture on performance

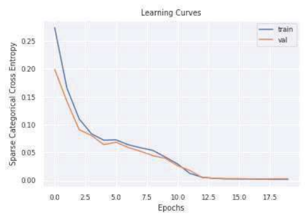

Fig. 10. The learning curves for VizNN Model

These evaluations for parameter settings show that each parameter has a specific optimum for optimising learning performance on the current dataset. It is highly likely that these values are dependant on the data, and can not be considered as general recommendations. Rather, the described process should be perform over again when the security context changes.

6 Evaluation

This section presents the overall results of the experiments on CIRA-CIC-DoHBrw-2020 dataset. Learning performance is evaluated using the AUPRC (Area under the Precision Recall Curve, suitable for imbalanced data), precision and recall metrics. Training and prediction time is evaluated for the VizNN and the two reference algorithms, Random Forests and XGBoost. Each score corresponds to an average over a 50-folds cross-validation process.

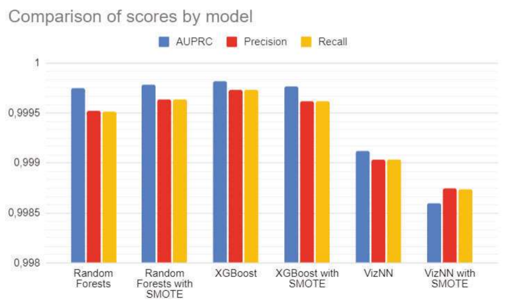

Fig. 11. Comparison of scores by model with and without SMOTE

Figure 11 shows the comparison of the different models: Random Forests, XGBoost and VizNN, without and with SMOTE (Synthetic Minority Over-Sampling TEchnique) oversampling strategy.

The first observation is that XGBoost outperforms Random Forests and VizNN for the three metrics AUPRC, precision and recall. Indeed, for the tests without SMOTE XGBoost reaches 0.9998 in AUPRC and 0.9997 for the precision and the recall. Random Forests comes in second place with an AUPRC of 0.9997, a precision of 0.9995 and a recall of 0.9995 VizNN gets an AUPRC of 0.9991, a recall of 0.9990 and a precision of 0.9990. Concerning the experiments on the dataset increased by SMOTE, the results are slightly different but the order remains unchanged. XGBoost provides an AUPRC of 0.9998, a precision of 0.9996 and a recall of 0.9996. Random Forests gets better results and reaches the same scores than XGBoost, namely 0.9998 for its AUPRC, 0.9996 of precision and 0.9996 for the recall. Finally, VizNN yields to 0.9986 in AUPRC, 0.9987 in precision and 0.9987 in recall.

For all these evaluation rounds, VizNN proves to be very competitive though slightly behind the reference algorithms – less that 0,08%.

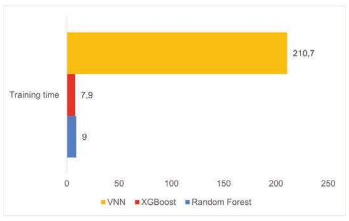

Fig. 12. Comparison of training time

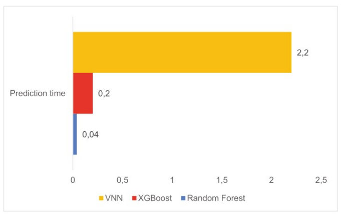

Fig. 13. Comparison of prediction time

Figure 12 shows the respective training times of Random Forests, XGBoost and VizNN. Training requires 9 seconds for Random Forests, 7.9 seconds for XGBoost, and 210.7 seconds for VizNN.

Figure 13 shows the respective prediction times of Random Forests, XGBoost and VizNN. Prediction requires 0.04 seconds for Random Forests, 0.2 seconds for XGBoost, and 2.2 seconds for VizNN.

In VizNN, the detection process is performed through a CNN which architecture is the following one: a convolutional layer with 32 filters and RELU activation; a maxpooling layer; a convolutional layer with 64 filters and RELU activation; a second maxpooling layer; and three final filtering layers. The configuration of our experimental setup has optimal results for 16 features, image size of 75x75 pixels and an architecture of 32 and 48 layers.

The evaluation of the proposed scheme shows that it is highly competitive with the reference algorithms XGBoost and Random Forest. It is performed by detecting attacks present in CIRA-CIC-DoHBrw-2020 dataset, which entails 60178 rows and 35 columns. The AUPRC metric achieves 0,9991 for VizNN, against 0,99982 for XGBoost and 0,99975 for Random Forest. The use of SMOTE has a slightly positive impact on detection capability in the Random Forest case only ($AUPRC = 0,99970$), and slightly negative impact for VizNN (0,9985) and for XGBoost (0,99976).

7 Conclusions

In this work, we propose VizNN, a new scheme for applying Convolutional Neural Networks to the detection of cybersecurity attacks through data augmentation. VizNN works by generating images from the system logs. It deliberately ignore non-representative information such as IP addresses and ports, and uses one-hot encoding of categorical information and timestamps. VizNN selects the most importance features as ranked by the gain they bring to learning, converts this features to image and resize them for optimizing the detection.

VizNN succeeds at integrating data augmentation in the analysis flow and at applying a highly scalable detection approach through Convolutional Neural Networks. It enhances the visibility of the analysis process by providing intermediate graphical representations of the system states to the security analyst. Nonetheless, VizNN still experience several limitations that need to be solved in future works. First, the graphical representations need important experience to be readable by the security analyst, like medical X-Rays require experienced practitioners. Second, the explicability and traceability of alerts to original traces are still better performed by alternative solutions like Random Forests or XGBoost.

The next step of this study is the evaluate to proposed scheme for massive datasets ($> 10^6$ data entries) to challenge the scalability capability of the various schemes under evaluation. Such an evaluation would be more representative of the amount of data that a Security Operating Centre (SOC) handles daily.

References

1. Kravchik, M., Shabtai, A.: Detecting cyber attacks in industrial control systems using convolutional neural networks. In: Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy. (2018) 72–83

2. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of Big Data **6**(1) (2019) 1–48
3. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 113–123
4. Breiman, L.: Random forests. Machine learning **45**(1) (2001) 5–32
5. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. (2016) 785–794
6. Hounsel, A., Borgolte, K., Schmitt, P., Holland, J., Feamster, N.: Analyzing the costs (and benefits) of dns, dot, and doh for the modern web. In: Proceedings of the Applied Networking Research Workshop. (2019) 20–22
7. Ariyapperuma, S., Mitchell, C.J.: Security vulnerabilities in dns and dnssec. In: The Second International Conference on Availability, Reliability and Security (ARES'07), IEEE (2007) 335–342
8. Afek, Y., Bremler-Barr, A., Shafir, L.: Nxnsattack: Recursive {DNS} inefficiencies and vulnerabilities. In: 29th {USENIX} Security Symposium ({USENIX} Security 20). (2020) 631–648
9. Banadaki, Y.M.: Detecting malicious dns over https traffic in domain name system using machine learning classifiers. Journal of Computer Sciences and Applications **8**(2) (2020) 46–55
10. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., Wang, C.: Machine learning and deep learning methods for cybersecurity. IEEE Access **6** (2018) 35365–35381
11. Gasmi, H., Bouras, A., Laval, J.: Lstm recurrent neural networks for cybersecurity named entity recognition. ICSEA **11** (2018) 2018
12. Kwon, D., Natarajan, K., Suh, S.C., Kim, H., Kim, J.: An empirical study on network anomaly detection using convolutional neural networks. In: 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), IEEE (2018) 1595–1598
13. Berman, D.S., Buczak, A.L., Chavis, J.S., Corbett, C.L.: A survey of deep learning methods for cyber security. Information **10**(4) (2019) 122
14. Ferrag, M.A., Maglaras, L., Moschoyiannis, S., Janicke, H.: Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications **50** (2020) 102419
15. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16** (2002) 321–357
16. Jakab, T., Gupta, A., Bilen, H., Vedaldi, A.: Unsupervised learning of object landmarks through conditional image generation. In: Advances in neural information processing systems. (2018) 4016–4027
17. Hinton, G.E., Krizhevsky, A., Wang, S.D.: Transforming auto-encoders. In: International conference on artificial neural networks, Springer (2011) 44–51
18. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. arXiv preprint arXiv:1406.2661 (2014)
19. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11) (2020) 139–144
20. Catak, F.O., Ahmed, J., Sahinbas, K., Khand, Z.H.: Data augmentation based malware detection using convolutional neural networks. PeerJ Computer Science **7** (2021) e346
21. Lim, S.K., Loo, Y., Tran, N.T., Cheung, N.M., Roig, G., Elovici, Y.: Doping: Generative data augmentation for unsupervised anomaly detection with gan. In: 2018 IEEE International Conference on Data Mining (ICDM), IEEE (2018) 1122–1127
22. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644 (2015)