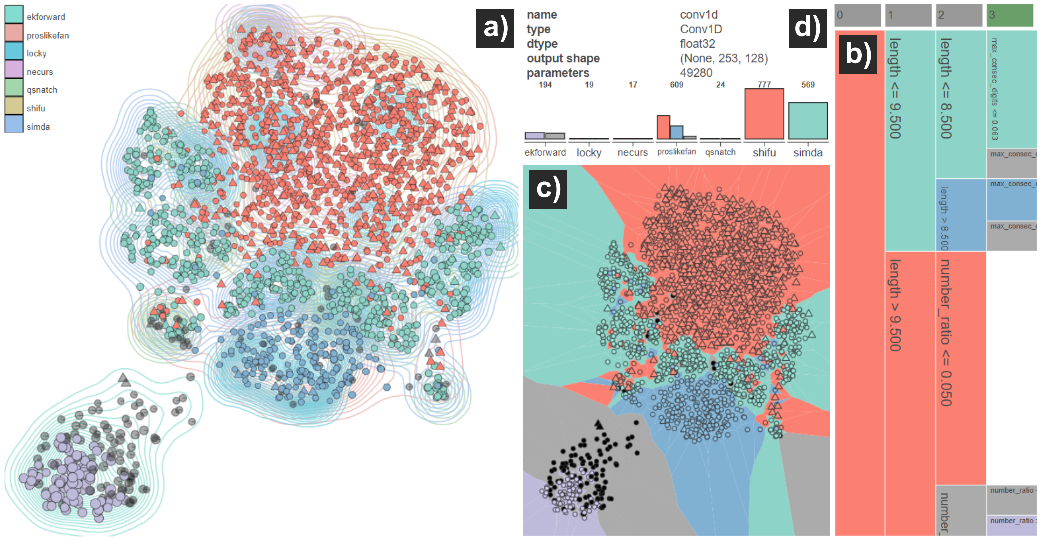
Interpretable Visualizations of Deep Neural Networks for Domain Generation Algorithm Detection

Franziska Becker[*]
University of Stuttgart

Arthur Drichel[†]
RWTH Aachen University

Christoph Müller[‡]
University of Stuttgart

Thomas Ertl[§]
University of Stuttgart



name conv1d
type Conv1D
dtype float32
output shape (None, 253, 128)
parameters 49280

Figure 1: The analysis view for the first convolutional layer of the CNN (for domains with length in $[8, 10]$ and top-level domain eu): **a)** 2D projection of the activations, displaying cluster and class membership. **b)** Calculated decision tree for the cluster labels, which uses length as the primary separator. **c)** Voronoi diagram of the 2D projection, showing the separation in the last decision tree level. **d)** Layer information and per-class cluster distribution.

Abstract

Due to their success in many application areas, deep learning models have found wide adoption for many problems. However, their black-box nature makes it hard to trust their decisions and to evaluate their line of reasoning. In the field of cybersecurity, this lack of trust and understanding poses a significant challenge for the utilization of deep learning models. Thus, we present a visual analytics system that provides designer of deep learning models for the classification of domain generation algorithms with understandable interpretations of their model. We cluster the activations of the models' nodes and leverage decision trees to explain these clusters. In combination with a 2D projection, the user can explore how the model views the data at different layers. In a preliminary evaluation of our system, we show how it can be employed to better understand misclassifications, identify potential biases and reason about the role different layers in a model may play.

Index Terms: explainable artificial intelligence (XAI)—visual analytics—model visualization—DGA detection; cybersecurity;

1 Introduction

Deep learning models have proven successful in many different domains from facial recognition and natural language processing to complex real-time strategy games [34], making it desirable to employ them in the context of cybersecurity as well. However, the black-box nature of such models means that their results cannot be trusted or evaluated easily. This makes their application to problems in cybersecurity difficult, where decisions often involve human operators and may have severe consequences . The rapidly growing field of explainable artificial intelligence (XAI) addresses such problems by providing insight into machine learning models in order to better understand their inner workings, improve the trust of model users or comply with legislation [1, 10].

In this work, want to present our first steps towards a visual analytics system for designers of deep learning models that helps them to explore decisions their models make. Designers of deep learning models are faced with many challenges when developing new models, which consist of thousands to millions of trainable parameters and many possibilities to assemble the architecture. In addition, data may contain artifacts that can lead to biases in the model, which can be hard to find using only performance metrics or simple data exploration. The goal of our system is to provide model developers with the opportunity to investigate their model's performance, how and when it separates data and derive possible measures by which to improve their model.

Using our system, we evaluate and visualize two different types of deep learning-based classifiers for multi-class domain generation algorithm (DGA) detection, for which high-performance deep learning-based classifiers have recently been developed [35,37]. One model is based on convolutional neural networks (CNNs) [37] and

[*] e-mail: franziska.becker@vis.uni-stuttgart.de
[†] e-mail: drichel@itsec.rwth-aachen.de
[‡] e-mail: christoph.mueller@visus.uni-stuttgart.de
[§] e-mail: thomas.ertl@vis.uni-stuttgart.de

the other is based on recurrent neural networks (RNNs) [35]. Both classifiers are trained using real-world benign data, captured at the central DNS resolver of RWTH Aachen University, and malicious data obtained from DGArchive [22]. The data set utilized comprises an approximate total of $540,000$ unique data points consisting of 92 different classes.

2 Related work

In this section, we discuss related work for our application domain of deep learning models for DGA detection, as well as research efforts in visual analytics for cybersecurity and explainable artificial intelligence.

2.1 DGA Detection Models

Botnets rely on domain generation algorithms to establish a connection to their command and control (C2) server. These DGAs periodically generate a large number of algorithmically-generated domains (AGDs), which serve as rendezvous points for the botnet. The bots query all of the AGDs, but only the ones registered by the botnet herder in advance resolve to valid IP addresses. This way, blocking connection attempts of a bot to its C2 server is more difficult compared to the use of fixed IP addresses or fixed domain names.

In the past, several approaches, which differ in the amount of information needed for classification, have been proposed to separate benign domains from malicious AGDs. The classification of contextless approaches (e.g. [8,25,28,35,37]) is based solely on the domain name that is to be classified. In contrast, context-aware approaches (e.g. [2,4,9,26,29,36]) leverage additional contextual information trying to enhance the classification. Prior work (e.g. [8,28,35,37]) shows that the contextless approaches are able to solve this binary classification task with state-of-the-art performance and are, in contrast to context-aware approaches, less resource-intensive and less intrusive regarding privacy.

Within the group of contextless approaches, two types of machine learning classifiers have been proposed: feature-based classifiers such as support vector machines (SVMs) or random forests (RFs) (e.g. [28]), and featureless models such as convolutional or recurrent neural networks (e.g. [8,25,35,37]). While the deep learning classifiers achieve superior classification performance [8,20,33,35], they fall short in the explainability of their predictions.

While the binary classification task has been intensively studied in the past, the more challenging task of attributing domain names to a specific DGA, which generated a malicious domain name, is less well studied. The advantage of this multi-class classification is that it ultimately enables the identification of the malware family that generated a specific malicious domain and thus enables targeted remediation measures. To the best of our knowledge, there is currently no contextless feature-based approach for DGA multi-class classification.

2.2 Visual Analytics for XAI

Recent years have seen an increased interest in the connection of visual analytics and explainable machine learning models [1,10], allowing for insight into the inner workings of models previously treated as black boxes.

Attribution methods (e.g. [3,30,38]) show which regions of the input data, such as pixels or characters, contribute most to a particular classification. In contrast, *feature visualization* methods (e.g. [6,14,16,18]) try to generate the features that maximally activate a particular node or class, using optimization with different strategies for regularization and diversity. More recently, Olah et al. [19] managed to combine feature visualization and attribution to allow the user to gain a more complete picture of the model's behavior, and Carter et al. [6] compute large-scale 2D projections of generated features to provide the user with a global map of what the model has learned to detect.

On the side of visual analytics, many new systems have been developed to evaluate, understand or improve machine learning models. With 'Squares' [23], the authors developed a system to compare and evaluate the performance of classification models. Other work (e.g. [21,32]) has produced visual analytics systems to interactively assess a model during training in order to improve it. Smilkov et al. [31] use different 2D and 3D projections to visualize word embeddings or image data sets. This allows the user to find global clusters, explore local neighborhoods and potentially discover meaningful directions in the projected space. However, such exploration is increasingly complicated for data with no intuitive interpretations. Exploring global structures is also difficult if single data points cannot be fully displayed (due to space limitations) or if there is much overdraw.

Prior work in this field has often focused on applications that work with image data, especially for feature visualization, which to our knowledge has not been successfully transferred to other types of data. In addition, even when methods can be applied to other data types, they are rarely tested for *hard* data, i. e. data without intuitive semantics. While humans excel at understanding natural images, language or music, their performance deteriorates for artificial data like sequences of pseudo-randomly generated characters, as is the case for many AGDs. Our method provides a supplementary explanation in the form of a decision tree that aims to alleviate this problem, making it possible to explore model decisions for *hard* data.

2.3 Visual Analytics for Cybersecurity

In the domain of cybersecurity, vast amounts of data are generated on a daily basis. Attack patterns constantly evolve and adapt, requiring the same of cybersecurity experts and their tools. To tackle these challenges, using the power of visual analytics for cybersecurity applications has been gaining more traction in the research community.

Many visual analytics systems in cybersecurity have a specific application case and are intended to be used by respective domain experts. Related topics range from privacy, anomaly detection and network traffic analysis to malware detection. [12]

Our approach is among a few that try to bring deep learning models and the advantages they provide into the domain of cybersecurity by helping users evaluate and interpret their models. Related work in a similar direction, but with a different goal, includes the exploration of adversarial examples by Norton [17] and Kahng [11]. Both works provide a web-based interactive playground to explore the generation of adversarial examples. Their goal is to educate the user on how adversarial examples are constructed and how they affect the classification. However, both works are limited in what data they use for their interface. Norton considers a subset of the MNIST [13] data set, i. e. small black and white images of hand-written digits, while Kanhg uses synthetic 2D data with a user-specified distribution. This makes it hard to use their approaches for data with less intuitive semantics and even then, constructing interpretations from the examples is left to the user.

3 Our Visual Analytics Approach

In the following, we describe our design goals, related tasks and the structure of our resulting visual analytics system.

3.1 Overview and Design Goals

The visualizations we developed aim to assist model developers who are intimately familiar with the working mechanisms of deep neural networks as well as the structure of their model. To formalize our main design goals, we surveyed previous work in the area of

XAI [7, 10, 24] and conducted an informal interview with a model developer who works on our application case:

DG1: Visualize different types of deep learning models. Since we have different types of deep learning models at our disposal, trained on the same data to solve the same classification task, a system that works for any kind of deep learning model is an important goal. In general, a deep-learning model has nodes (or neurons) which are packed into different layers and connected by weighted links. During training, these weights are modified in a way to increase the model's performance, making them a good candidate to try to understand how deep learning models work. However, the trained weights alone are only one part of the what determines the result, leaving out the bias term as well as the non-linear activation function used inside the model. Therefore, to provide a more complete picture in a (deep learning) model-agnostic fashion, we focus on how the different layers of the model are activated by specific data.

DG2: Explain patterns model layers find. To better understand how the model arrives at its results, we want to explore which explanations can be extracted from patterns inside the different model layers. As we deal with data that does not necessarily exhibit intuitive semantics to a human, the system must provide a mechanism to describe instances, groups or patterns of the data in a way that is interpretable by the user.

DG3: Provide an overview of the data and model performance. Deep learning models commonly require large amounts of data to train on, which makes it hard for the model developer to have detailed knowledge about peculiarities it may contain. In addition, finding interesting subsets to consider for in-depth investigation is difficult without some form of overview of the complete data set.

3.1.1 Tasks

Using our previously formulated design goals together with our user's comments, we identified several tasks that our visual analytics system should address.

T1: Investigate causes for misclassification Especially when there is a large number of classes, causes for class confusion are not readily apparent from consulting examples or a confusion matrix. However, in order to improve a model's performance, understanding of how such confusion arises is vital. This task requires the user to be able to find and select misclassified data (**DG3**) and explore patterns like a common feature value (**DG2**).

T2: Find possible sources of bias A trust-worthy model is one with little to no bias. This is hard to achieve, particularly when large amounts of data are used for training and it is difficult for the user to manually remove bias from the data or find it in the first place. Similarly to the first task, the user needs information about the data to formulate a hypothesis (**DG3**) and then verify its correctness (**DG2**).

T3: Explore the role of different layers In relation to **DG1** and **DG2**, the user may gain a better understanding of the roles of different types of layers in a deep learning model by exploring the data layer by layer and comparing layers of the same type across models. This may facilitate transfer learning by giving the user an idea which layers could be useful to speed up the development of new models for different problems.

3.2 Visual Analytics System

The visual analytics system consists of several separate workspaces, including one for **data exploration & selection** and one for **analysis** of the progression that instances make across the different layers of a model.

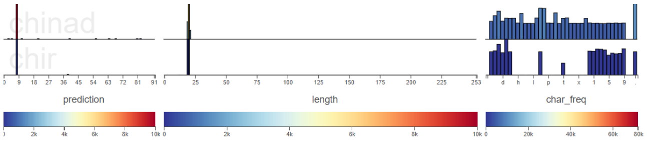


Figure 2: Histogram matrix segment showing the two rows of the *chinad* (top) and *chir* classes (bottom) for the CNN. The latter row shows that the *chir* class contains only a small number of instances, all of which are predicted to be members of the *chinad* class.

3.2.1 Data Exploration & Selection

To address design goal **DG3**, this workspace provides an overview of the complete data set and functions as a means of specifying a customized selection.

Because of the large number of classes in the data set, we chose a compact visualization where each class is represented by a row in a histogram matrix. The three columns of the matrix contain values for the three features we found to give a good general overview of the data: *prediction*, *length* and *character frequency* (cf. Fig. 2). The *prediction* column displays how many instances are predicted for each class, using that class' unique id for the x-axis placement. This enables the user to see how the model performed for the various classes and where class confusion may exist. The other columns allow the user to identify potentially interesting subsets since DGAs often use only a special set of lengths and characters.

The bars' color scale encodes the relative relationship to the largest value for that column in the whole data set, preventing perception problems that may occur when trying to use a global height scale for all bars and showing any differences in class representation. The user can interact with the histograms in order to select a subset of the data, which is then used inside the analysis work space. In addition, we provide an interactive user interface (UI) where the user can define more complex subsets, using additional attributes such as class membership, the correctness of the classification, or regular expressions the domains should match.

3.2.2 Analysis

The analysis workspace is the centerpiece of our system, addressing design goal **DG2**. For each layer l_i of the model, the following transformation pipeline is executed in order to compute the necessary data needed for the different visualizations in this workspace:

1. Calculate the activations a_i of the model nodes for the current layer l_i.
2. Reduce the dimensionality of a_i using principal component analysis (PCA) to get act_i'.
3. Perform unsupervised clustering on a_i' employing the HDBSCAN [5] algorithm.
4. Reduce the dimensionality a_i' to 2 dimensions using the UMAP [15] algorithm.
5. Train a decision tree to predict the cluster labels.

The first visualization computed from this data is a scatter plot with density contours (see Fig. 1**a**) which displays a 2D projection of the activation data for the current layer. Each glyph represents a data point, colored according to its cluster membership and scaled depending on the magnitude of its activation vector. In order to visually identify misclassified instances, they are plotted as triangles instead of circles. In the background, density contours are drawn for each class in the selection. These visual encodings allow the user to combine the information of class membership, cluster membership and classification correctness at a glance.

In Fig. 1**b**, we draw the trained decision tree as an icicle plot, where the color of a leaf node is chosen according to its cluster. Intermediate nodes have the same color as their child with the most assigned instances. To visually connect the 2D projection with the decision tree, we provide a complementary visualization (see Fig. 1**c**) that displays the Voronoi diagram of the projected data points. The user can interactively switch between the branching levels of the decision tree by clicking on the level label, which automatically updates the Voronoi diagram. This allows for exploration of how the separation of the clusters is constructed by the decision tree. Finally, in Fig. 1**d** we show some information about the currently selected layer as well as an overview of the per-class cluster distribution.

The transformation pipeline is sequentially executed for each layer and the user may switch between the resulting visualizations as he sees fit. In order to support the user's ability to find correspondences between the scatter plots of the different layers, we employ an orthogonal procrustes method [27] to find the best rotation to align any layer with its predecessor and smoothly transition between the two using animation.

4 Results

This section discusses the preliminary results we observed for tasks **T1** to **T3** when using our visual analytics system with a CNN and a RNN trained for multi-class DGA detection. The CNN [37] consists of five trainable layers, including two 1-dimensional convolutional layers, while the RNN [35] is based on a long short-term memory layer and combines three trainable layers in total. We choose specific subsets of the data set in order to demonstrate the advantages our tool provides and prove that our system is able to address the previously defined tasks.

T1 Using the overview visualization in the selection workspace, we find out that the CNN has trouble with the class *chir*, whose instances are all predicted to be members of class *chinad*. We can also see that the *chir* class consists of a much smaller number of instances, has instances with similar length but uses a smaller subset of characters than the *chinad* class. In order to investigate this misclassification, we select both classes and switch to the analysis view. For the first layer, we can see that more than half of all instances are not assigned to a cluster and that there is much overlap between the two classes in regard to both cluster membership as well as spatial positioning. In subsequent layers, the number of instances assigned to a cluster fluctuates and we observe several occurrences where the misclassified instances form clearly separated clusters. The decision tree also finds explanations for these clusters, although the clusters cannot be perfectly separated and still include some *chinad* instances. While the CNN model seems to be capable of separating both classes in intermediate layers it fails to discriminate these two classes at the final output layer. We suspect that this behavior is caused by the fact that the *chir* class is overshadowed by the large number of *chinad* instances. The output layer is used for the final attribution and biased to the better represented class, while we believe that the intermediate layers learn to extract features and benefit from all samples during training and not just from a single class. For comparison, we inspect the same selection for the RNN. Surprisingly, we see that it has no problems separating the two classes and the corresponding decision tree uses the same features as the one for the CNN for the separation.

T2 To find potential biases, we inspect the overview visualization to find that only a small number of classes include more than one hyphen in their domains. Selecting all domains with at least two hyphens contains the two malicious classes *matsnu* and *tsifiri* as well as the *benign* class. The analysis view then reveals that both models quickly separate these classes. However, the decision trees show that for many layers most of the benign instances can be separated based only on their length and maximum number of consecutive consonants. This finding may present an opportunity to craft and test adversarial examples for this particular model, which requires further work.

T3 To determine the roles of the convolutional layers of the CNN, we investigate how many clusters they find and how these are separated. To achieve this, we investigate several selections: We limit the data either to a particular top-level domain or to a fixed length range. When the top-level domain is fixed, we always find two clusters and some unassigned instances in the fist convolutional layer. One cluster only contains a single class while the other cluster is mixed. All decision trees employ length to separate the bulk of these two clusters.

In case of the fixed length range, we find two to three clusters, with one mixed cluster for all ranges except one, which only includes two different classes that are perfectly separated by the clustering. Here, the decision trees primarily use the top-level domain to separate clusters. For all selections, later layers use more complex features such as the maximum number of consecutive consonants or digits to split the data, although the number of clusters does not necessarily increase.

These observations indicate that the first convolutional layer most strongly reacts to variations in length and top-level domain, which is expected given that many DGAs only use a specific set of top-level domains and length values. Later layers additionally learn to discriminate classes by higher level features such as features based on ratios.

5 Conclusion & Future Work

In this paper, we present the design and preliminary results for a visual analytics system that allows designers of deep learning models for multi-class DGA classification to explore the interpretations of their models' results. We provide a novel approach to tackle current research questions in the field of XAI for algorithmically generated data with hard-to-interpret semantics from the context of cybersecurity. We allow the user to analyze the progression of a customized subset of the data set throughout the different layers of a deep learning model. We employ clustering to find patterns in the activations of the model's nodes and present them to the user with a decision tree to interpret these clusters.

A limitation of our approach concerns the meaning one may attribute to the clustering result. Although our results offer an indication that the clustering works well for this application, it requires more thorough evaluation. This issue may be aggravated by the fact that we only operate on a subset of the data set. Especially when the number of classes in the selection is large, the clustering results can vary, which in turn can lead to misinterpretation of the provided result. In addition, there is no clear connection between the provided explanations of the decision tree and the model's classification, i. e. the decision tree provides a possible explanation for the clusters which does not necessarily have to reflect how the model separates this data, especially when there are many possible explanations with equal merit.

For future research, we would like to perform an in-depth user study to evaluate the usability and utility of our developed system. Furthermore, we want to use those results to extend our system such that other users could benefit from it, e. g. that security analysts may consult it for the investigation of malicious domains. In addition, we want to explore how interactive manipulation of the data or features used to build the decision tree could be incorporated. Such functionalities could be used to test how robust the model is to adversarial examples and where more sophisticated features have been learned by the model.

Finally, we see potential in investigating whether our visualization system can be extended to test features to facilitate feature selection for the development of models with better intrinsic interpretabiliy. Such a strategy could help to bridge the gap between high-performance black-box models and hard-to-develop interpretable models.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833418. Simulations were performed with computing resources granted by RWTH Aachen University under project rwth0438.

References

[1] A. Adadi and M. Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 2018.

[2] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon. From throw-away traffic to bots: Detecting the rise of DGA-based malware. In *USENIX Security Symposium*, 2012.

[3] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 2015.

[4] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel. Exposure: A passive DNS analysis service to detect and report malicious domains. *ACM Transactions on Information and System Security*, 2014.

[5] R. J. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 2013.

[6] S. Carter, Z. Armstrong, L. Schubert, I. Johnson, and C. Olah. Activation atlas. *Distill*, 2019. `https://distill.pub/2019/activation-atlas`. doi: 10.23915/distill.00015

[7] J. Choo and S. Liu. Visual Analytics for Explainable Deep Learning. *IEEE Computer Graphics and Applications*, 2018.

[8] A. Drichel, U. Meyer, S. Schüppen, and D. Teubert. Analyzing the Real-World Applicability of DGA Classifiers. In *International Conference on Availability, Reliability and Security*. ACM, 2020.

[9] M. Grill, I. Nikolaev, V. Valeros, and M. Rehak. Detecting DGA malware using netflow. In *IFIP/IEEE International Symposium on Integrated Network Management*, 2015.

[10] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *IEEE Transactions on Visualization and Computer Graphics*, 2019.

[11] M. Kahng, N. Thorat, D. H. P. Chau, F. B. Viégas, and M. Wattenberg. Gan lab: Understanding complex deep generative models using interactive visual experimentation. *IEEE transactions on visualization and computer graphics*, 2018.

[12] V. Lavigne and D. Gouin. Visual Analytics for cyber security and intelligence. *The Journal of Defense Modeling and Simulation*, 2014. `https://doi.org/10.1177/1548512912464532`. doi: 10.1177/1548512912464532

[13] Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2010.

[14] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[15] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[16] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in neural information processing systems*, 2016.

[17] A. P. Norton and Y. Qi. Adversarial-Playground: A visualization suite showing how adversarial examples fool deep learning. In *2017 IEEE Symposium on Visualization for Cyber Security (VizSec)*. IEEE, 2017.

[18] C. Olah, A. Mordvintsev, and L. Schubert. Feature Visualization. *Distill*, 2017. `https://distill.pub/2017/feature-visualization`. doi: 10.23915/distill.00007

[19] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev. The building blocks of interpretability. *Distill*, 2018. `https://distill.pub/2018/building-blocks`. doi: 10.23915/distill.00010

[20] J. Peck, C. Nie, R. Sivaguru, C. Grumer, F. Olumofin, B. Yu, A. Nascimento, and M. De Cock. CharBot: A Simple and Effective Method for Evading DGA Classifiers. *arXiv:1905.01078*, 2019.

[21] N. Pezzotti, T. Höllt, J. Van Gemert, B. P. Lelieveldt, E. Eisemann, and A. Vilanova. Deepeyes: Progressive visual analytics for designing deep neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[22] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla. A comprehensive measurement study of domain generating malware. In *USENIX Security Symposium*, 2016.

[23] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams. Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE Transactions on Visualization and Computer Graphics*, 2016.

[24] W. Samek and K.-R. Müller. Towards explainable artificial intelligence. In W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, eds., *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer International Publishing, 2019. doi: 10.1007/978-3-030-28954-6_1

[25] J. Saxe and K. Berlin. eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys. *arXiv:1702.08568*, 2017.

[26] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero. Phoenix: DGA-based botnet tracking and intelligence. In *Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2014.

[27] P. H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 1966.

[28] S. Schüppen, D. Teubert, P. Herrmann, and U. Meyer. FANCI: Feature-based automated nxdomain classification and intelligence. In *USENIX Security Symposium*, 2018.

[29] Y. Shi, G. Chen, and J. Li. Malicious domain name detection based on extreme machine learning. *Neural Processing Letters*, 2018.

[30] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[31] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F. B. Viégas, and M. Wattenberg. Embedding projector: Interactive visualization and interpretation of embeddings. *arXiv preprint arXiv:1611.05469*, 2016.

[32] T. Spinner, U. Schlegel, H. Schäfer, and M. El-Assady. explainer: A visual analytics framework for interactive and explainable machine learning. *IEEE Transactions on Visualization and Computer Graphics*, 2019.

[33] J. Spooren, D. Preuveneers, L. Desmet, P. Janssen, and W. Joosen. Detection of algorithmically generated domain names used by botnets: A dual arms race. In *Proceedings of the 34rd ACM/SIGAPP Symposium On Applied Computing*. Association for Computing Machinery, 2019.

[34] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 2019.

[35] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant. Predicting domain generation algorithms with long short-term memory networks. *arXiv:1611.00791*, 2016.

[36] S. Yadav and A. L. N. Reddy. Winning with dns failures: Strategies for faster botnet detection. In *International Conference on Security and Privacy in Communication Systems*. Springer, 2011.

[37] B. Yu, J. Pan, J. Hu, A. Nascimento, and M. De Cock. Character level based detection of DGA domain names. In *International Joint Conference on Neural Networks*. IEEE, 2018.

[38] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.