Impact of "TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time"

ACSAC Cybersecurity Artifact Competition

Shae McFadden[†§‡], Zeliang Kan[†‡◇], Daniel Arp[‖], Feargus Pendlebury[‡], Roberto Jordaney[*], Johannes Kinder[¶], Fabio Pierazzi[†‡], Lorenzo Cavallaro[‡]

[†]King's College London, [‡]University College London, [§]The Alan Turing Institute, [‖]TU Wien, [¶]LMU Munich, [◇]HiddenLayer, [*]Royal Holloway, University of London

Abstract

Tesseract is an open-source framework which enables an unbiased realistic, time-aware evaluation of machine learning-based malware classification. The Tesseract framework was originally released in conjunction with a paper published at USENIX Security Symposium 2019, which demonstrated how to remove experimental bias. Since the artifact's original release, it has been presented in many keynotes and seminars, and has been used by academics and practitioners worldwide, influencing the design of further research questions and experiments in the field of ML-based malware detection, and garnering 415 Google Scholar citations (as of September 2024).

Citation Note

This paper highlights the impact that Tesseract has had since its original release. If you use Tesseract as part of a project or publication, then please cite the original work https://www.usenix.org/conference/usenixsecurity19/presentation/pendlebury and the extended work https://arxiv.org/abs/2402.01359.

CCS Concepts

• **Security and privacy → Software and application security**; • **Computing methodologies → Machine learning**.

Keywords

Concept Drift, Experimental Bias, Malware Detection, Performance Decay

1 Introduction

The trend of near-perfect F_1 scores in malware classification papers five years ago led to the question of whether Android malware classification was a solved problem. Malware classification was in fact not a solved problem, and the near-perfect performance was a result of spatio-temporal biases. Tesseract was developed to allow a realistic evaluation of a malware classifier over time free from spatial and temporal bias. After Tesseract's release, it became the benchmark for how to perform fair malware classification evaluation, influencing the experimental design of the subsequent papers as evidenced by its *415* citations to date.

Tesseract was implemented as a Python library, designed to easily integrate with common ML workflows. The design of Tesseract was heavily inspired by and is fully compatible with the popular machine learning libraries Scikit-learn [131], Keras [37], and PyTorch [129]. Tesseract provides the following core capabilities:

- *Temporal bias removal* through maintaining the temporal training consistency (C1) and temporal goodware/malware time-window consistency (C2).
- *Spatial bias removal* through enforcing a realistic malware-to-goodware ratio in testing (C3).
- *Time-aware evaluation* of a malware classifier with extensible integration of sampling and rejection mechanisms.
- *Time-aware metric* (AUT) to capture a classifier's robustness to time decay and allows for the fair comparison of different algorithms with optional observation time window.
- *Tuning algorithm* to empirically optimize the performance of a classifier when malware is the minority class.

The Tesseract framework was originally released in 2019, in a private repository that could be accessed with a request form. Since 2024, it has been re-released fully open source at:

https://github.com/s2labres/tesseract-ml-release

The re-release of Tesseract is part of the conference paper's journal extension [78], which included updates and refactoring of the framework. Before being released on GitHub, Tesseract was accessed by more than *102* universities, *10* companies, and *6* research centers. Additional information regarding Tesseract can be found on its project page at: https://s2lab.cs.ucl.ac.uk/projects/tesseract/.

2 The Tesseract Framework

The goal of Tesseract is to ensure an unbiased and time-aware evaluation of ML classifiers (e.g. malware detection). To achieve this, Tesseract enforces temporal and spatial constraints to prevent performance inflation as a result of experimental bias. Tesseract aims to reduce the burden on the algorithm designer by keeping track of these properties at each stage of the experiment pipeline. Furthermore, Tesseract is constructed in a modular fashion corresponding to the different stages of the evaluation cycle to improve interoperability. Therefore, any component of the framework can be appropriately selected and used in conjunction with other libraries or methodologies. The following subsections highlight the core contributions of Tesseract and discuss their connection to the different stages of the experiment pipeline (see Figure 1).

2.1 Temporal Bias

Although a sample is typically represented by a feature vector X and a ground truth label y, Tesseract additionally expects a timestamp t. This allows Tesseract to enforce temporal consistency when

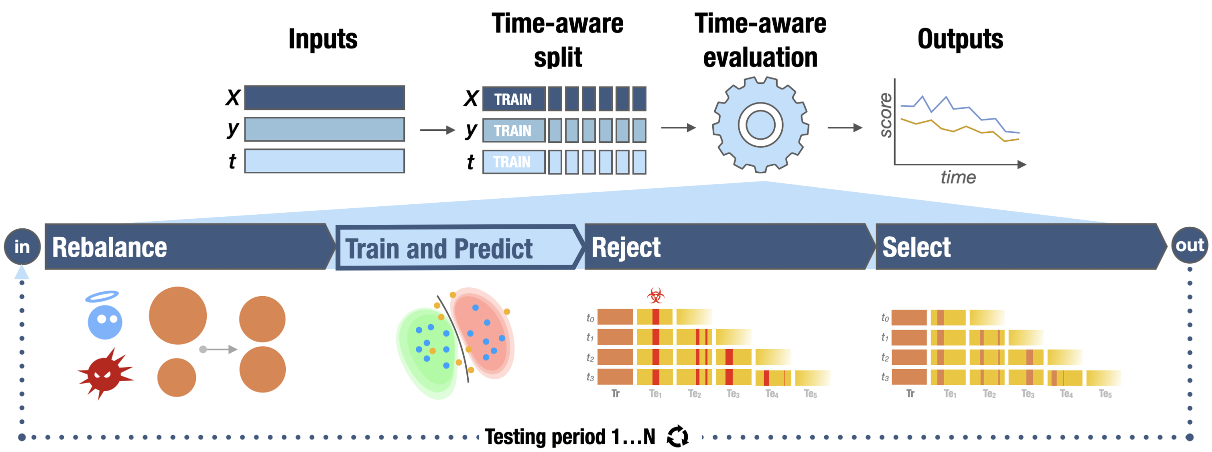

Figure 1: The pipeline of the TESSERACT artifact, using malware classification as an example.

partitioning the dataset for training, validation, and testing sets. The test set is further split into separate testing periods, each of which contains only test samples from that particular time window (time-aware split in Figure 1). The temporal partitioning enforces the following two constraints to eliminate temporal bias.

- **(C1) Temporal training consistency.** All samples in the training set must be *strictly* temporally precedent to the samples in the testing set.
- **(C2) Temporal goodware/malware time-window consistency.** In every testing slot, all test samples must be from the same time window.

The violation of these temporal constraints causes inflated classifier performance as a result of the incorporation of future knowledge (C1) and distinguishing by artifactual differences (C2).

2.2 Spatial Bias

After the dataset has been temporally split into the training and testing sets, the removal of spatial bias can be performed. Each testing period is then checked against the following constraint and enforced via rebalancing.

- **(C3) Realistic mw-to-gw ratio in testing.** The distribution of malware within the testing periods must be as close as possible to the estimated distribution of malware in the wild.

The violation of this spatial constraint can cause an inflated performance as a result of changing the dynamics of the underlying classification problem.

2.3 Tuning Algorithm

The tuning algorithm has the objective of estimating an optimal training set class-ratio (e.g., percentage of malware) to improve a target performance metric on a time-aware validation set. The purpose is to tune the training set so that a classifier achieves a higher performance rate over time.

The tuning algorithm enforces the constraints C1, C2, C3 and relies on AUT (see subsection 2.5) to achieve three possible targets for the malware class: a higher F1 score, higher Precision, or higher Recall. The algorithm performs progressive subsampling of the goodware class to optimize the training class distribution subject to a maximum error rate. This process is performed on the training set available in the rebalancing stage of the evaluation (rebalance in Figure 1).

2.4 Time-aware Evaluation

After all constraints are enforced and optional tuning has occurred, the classifier is trained on the training set available at the current iteration of the evaluation. The classifier then attempts to predict the correct classes for the test samples in the current period (train and predict in Figure 1). Before repeating the process on the next test period, the classifier can perform rejection and selection as described below.

Rejection Mechanism. A classifier can choose not to classify a particular observation (abstaining classification; classification with a rejection option, e.g., [15, 77]); rejected objects are quarantined for manual inspection (reject in Figure 1). Their predictions are not included in the performance results, however, TESSERACT reports the quantity of quarantined samples per period. Rising quarantined samples signal the onset of concept drift, the aging of underlying ML models, and the opportunity to explore test-time adaptation and continual learning settings [34].

Selection Mechanism. Following the rejection stage, an active learning sample selection strategy can be deployed to select the most informative testing samples to relabel manually (select in Figure 1). These samples are then integrated into the training set prior to the next cycle [34]. As in rejection, TESSERACT reports the number of selected samples per period.

2.5 Time-Aware Metrics

TESSERACT maintains a set of standard metrics calculated during each iteration of the evaluation cycle. Furthermore, TESSERACT provides the AUT (Area Under Time) metric, which allows the evaluation of malware classifier performance against time decay in realistic experimental settings obtained by enforcing C1, C2, and C3. AUT is defined as follows:

$$AUT(\mathbb{P}, N) = \frac{1}{N-1} \sum_{k=1}^{N-1} \frac{[\mathbb{P}(X_{k+1}) + \mathbb{P}(X_k)]}{2} \tag{1}$$

where $\mathbb{P}(X_k)$ is the value of the point estimate of the performance metric $\mathbb{P}$ (e.g., F_1) evaluated at point $X_k := (W + k\Delta)$, N is the number of test slots, and $1/(N-1)$ is a normalization factor so that AUT $\in [0, 1]$. The perfect classifier with robustness to time decay in the time window S has AUT = 1.

3 Impact

Since its initial release in August 2019, TESSERACT has generated significant impact across academia, education, and industry. The statistics on impact in this section refer primarily to the timeframe of August 2019–January 2024, where accessing TESSERACT required filling out a form. TESSERACT is now freely accessible on GitHub.

3.1 Academic Impact

In the context of academic impact, *108* academic or research institutions from *24* countries around the world requested the artifact prior to its re-release. The complete alphabetized list can be found in Appendix A. Moreover, the TESSERACT paper [133], which introduced the artifact, has received *415* citations to date. The impact of TESSERACT on subsequent research is evident through the consideration and elimination of temporal and spatial biases in the evaluation of classifiers across a wide range of machine learning applications in security domains. We delineate the areas of research in which the TESSERACT artifact has had a notable impact as follows.

- **Spatial**: *17* papers removed spatial bias from their evaluation citing TESSERACT [17–21, 23, 46, 54, 75, 92, 98, 117, 146, 152, 176, 179, 182].
- **Temporal**: *34* papers removed temporal bias from their evaluation citing TESSERACT [5, 6, 28, 42, 44, 47, 49, 50, 55, 64, 68, 72, 73, 80–82, 89, 96, 125, 143, 144, 153, 159, 163, 164, 171, 173, 183–186, 190, 191, 197].
- **Spatio-Temporal**: *27* papers removed both temporal and spatial bias from their evaluation citing TESSERACT [9, 25, 26, 35, 39, 48, 51, 59, 74, 88, 93, 97, 100, 101, 107, 109, 115, 121, 124, 157, 161, 166, 172, 177, 178, 193, 194].
- **AUT**: *10* papers used the AUT Metric to perform their evaluation [26, 39, 59, 65, 93, 186, 190, 193, 194, 199].

Beyond TESSERACT's concrete impact on the evaluations of academic papers, it has also influenced PhD and Masters theses, as well as surveys and SoKs on the topic of classifications tasks for security.

- **PhD**: *41* PhD theses cite TESSERACT [2–4, 24, 32, 33, 40, 41, 43, 53, 58, 63, 67, 70, 71, 76, 84–86, 94, 99, 106, 114, 128, 132, 140, 142, 145, 148, 149, 151, 155, 156, 165, 169, 175, 181, 187, 195, 196, 198].
- **Masters**: *12* Masters theses cite TESSERACT [27, 69, 120, 122, 123, 127, 130, 141, 154, 162, 167, 189].
- **Surveys & SoKs**: *38* surveys and SoKs cite TESSERACT as part of their review [1, 7, 10–12, 16, 22, 31, 45, 56, 57, 60–62, 66, 83, 87, 91, 102, 103, 108, 110, 113, 116, 118, 119, 126, 139, 147, 150, 158, 160, 168, 170, 174, 180, 188, 192].

In addition to the broader implications of TESSERACT, this work has underpinned subsequent publications in leading security conferences and workshops by the authors of the artifact [8, 13–15, 29, 30, 36, 38, 79, 90, 111, 112, 136–138]. A notable example is the paper on "*Dos and Don'ts of Machine Learning in Computer Security*" [13], which won a Distinguished Paper Award at the USENIX Security Symposium 2022, and originated as a follow-up collaboration from the TESSERACT conference paper [133]. The extended journal version of TESSERACT [78] is currently under review and brings with it an updated version of the artifact to ensure its continued relevance.

3.2 Educational Impact

Beyond the impact TESSERACT has had on academic research, its influence extends to machine learning for cybersecurity education at multiple institutions. In the context of university education, TESSERACT is taught as a part of classes, as well as presented in talks and seminars at the following institutions:

- *University of Bologna & University of Cagliari*: taught in labs associated with machine learning security [134, 135].
- *University College London*: taught as a part of the malware course in the MSc in Information Security [105].
- *University of Modena*: presented as part of a series of seminars.
- *Imperial College London*: presented in a keynote at the Machine Learning and Cyber Security Symposium 2024 [104].
- *Karslruhe Institute of Technology (KIT)*: taught as part of guest lectures on drift in malware classification in 2021–2022.
- *TU Berlin*: presented in the Software Engineering Ph.D. & PostDoc Winter School.
- *KU Leuven*: presented in several independent talks and keynotes at the Security and Privacy in the Age of AI Summer School in 2022–2024 [95].

Furthermore, TESSERACT has been presented in several invited talks and keynotes, including the Deep Learning and Security workshop 2023 (co-located with IEEE S&P 2023), Tsinghua University, Zhejiang University, BIFOLD TU Berlin, University of Luxembourg, AI Security SIG Meeting, and the University of British Columbia, to name a few. Finally, TESSERACT is used on track three of the ELSA EU benchmark competition for cybersecurity [52], showing the continued value of the evaluation this artifact provides.

3.3 Industrial Impact

The influence TESSERACT has had on industry is evident through the different companies in security and artificial intelligence, as well as industrial research laboratories that have requested access to the artifact.

- **Security companies**: *CKIN*, a company specializing in European telecommunication security; *IOvation*, a provider of zero-trust enterprise security solutions; and *ESTsecurity*, focusing on malware analysis and threat intelligence.

- **AI company**: *Vicomtech* provides artificial intelligence and visual computing knowledge transfer to industry.
- **Industrial research laboratories**: Toshiba Research Innovation Laboratory, Visa Research, Samsung Research, Capital One, and MITRE.

In addition to the requests for access to the Tesseract artifact, it has also been presented at the following industry-focused events:

- IBM AI Masterclass in Dublin in 2024
- USENIX ENIGMA 2019
- Avast CyberSec&AI Connected in Prague in 2019.

The diverse array of companies expressing interest in the Tesseract artifact underscores its impact, which extends beyond its initial application.

4 Conclusion

Tesseract re-established standards for the evaluation of ML-based classifiers in various cybersecurity domains. The artifact showed that the tantalizing performances of up to 99% present in prior papers were often inflated. Therefore, Tesseract re-orientated the research field toward realistic settings and drift mitigation strategies.

The academic impact of this artifact is demonstrated by its influence on subsequent research carried out by both the original authors and the broader academic community. Furthermore, the artifact functions as an educational resource, ensuring that the biases present in prior research are not perpetuated in the future.

Our journal extension of the original paper, along with updates to the artifact, attests to our dedication to the continuation of this work and to the enduring significance of the artifact.

Acknowledgments

This research has been partially sponsored by: UK EPSRC Research Grants EP/L022710/1, EP/P009301/1 and EP/X015971/1; the Vienna Science and Technology Fund (WWTF) [Grant ID: 10.47379/VRG23011]; and Google ASPIRE awards. Zeliang's work was completed during his studies at KCL and UCL. We would also like to thank Louise Bowler (from the Research Software Engineering team at KCL) for supporting refactoring and testing of the Tesseract prototype on Github.

A Tesseract Access Requests

The complete alphabetized list of all institutions that requested access to Tesseract before public release on GitHub is as follows: ANSSI, Beijing Institute of Technology, Beijing University of Posts and Telecommunications, Birla Institute of Technology and Science, Boise State University, Cairo University, California State University Long Beach, Carnegie Mellon University, Central Queensland University, China University of Geosciences, Columbia University, Czech Technical University, Deakin University, Donghua University, Eindhoven University of Technology, Federal University of Paraná (UFPR), Florida International University, French Institute for Research in Computer Science and Automation, Georgia Institute of Technology, Guangdong University, Guilin University of Electronic Science and Technology, Hamad Bin Khalifa University, HeiFei University of Technology, Heidelberg University, Helmholtz Center for Information Security (CISPA), Huazhong University of Science and Technology (HUST), IIT Hyderabad, IIT Indore, IIT Kanpur, IIT Madras, ITWM Fraunhofer, Indraprastha Institute of Information and Technology Delhi (IIITD), Institute for Infocomm Research, Institute for Information Industry, Jinan University, King's College London, La Trobe University, Leiden University, Maulana Abul Kalam Azad University of Technology, Monash University, Nanjing University, National Institute of Technology Rourkela, National Institute of Technology Tiruchirappalli, National Security Research Institute Korea, National Taiwan University, National University Of Sciences and Technology (NUST), National University of Defence Technology China, National University of Singapore, New York University, Nirma University, Northeastern University, Northwestern University, Osaka University, Peking University, Princeton University, Queen's University Belfast, Rice University, Rochester Institute of Technology, Royal Holloway, Shanghai Jiaotong University, Singapore Management University, Swinburne University of Technology, TU Berlin, TU Braunschweig, TU Darmstadt, TU Dublin, TU Munich, Tezpur University, The Alan Turing Institute, The Hong Kong Polytechnic University, The Interdisciplinary Center Herzliya (IDC), Tsinghua University, Ulsan National Institute of Science & Technology Korea (UNIST), UniBw, Universidad Carlos III de Madrid, University College London, University of Adelaide, University of Bari, University of Birmingham, University of Bristol, University of British Columbia, University of Cagliari, University of Chinese Academy of Sciences, University of Florida, University of Hertfordshire, University of Kerala, University of Luxembourg, University of Maryland College Park, University of Milan, University of Neuchâtel, University of New South Wales, University of Notre Dame, University of Quebec, University of Rennes, University of Salerno, University of Science and Technology of China, University of Southampton, University of Toronto, University of Trento, University of West England Bristol, University of York, University of the Basque Country, VIT Bhopal, Washington State University, Wrocław University of Science and Technology, Xidian University, Yonsei University, Zhejiang University.

References

[1] A Abusitta, MQ Li, and BCM Fung. 2021. Malware classification and composition analysis: A survey of recent developments. *Journal of Information Security and . . .* (2021). https://www.sciencedirect.com/science/article/pii/S2214212621000648

[2] A Abusnaina. 2022. Studying the Robustness of Machine Learning-based Malware Detection Models: Analysis, Design, and Implementation. (2022). https://stars.library.ucf.edu/etd2020/1458/

[3] H Aghakhani. 2023. Machine Learning and Security in Adversarial Settings. (2023). https://search.proquest.com/openview/62b8a0558446d181fddd57eef1f9f6a6/1?pq-origsite=gscholar&cbl=18750&diss=y

[4] BA Alahmadi. 2019. Malware detection in security operation centres. (2019). https://ora.ox.ac.uk/objects/uuid:56ff69ad-24cd-4877-819f-38cc87055efb

[5] BA Alahmadi, E Mariconti, R Spolaor, and ... 2020. BOTection: Bot detection by building Markov Chain models of bots network behavior. *Proceedings of the 15th . . .* (2020). https://doi.org/10.1145/3320269.3372202

[6] F Alotaibi and S Maffeis. 2024. Mateen: Adaptive Ensemble Learning for Network Anomaly Detection. (2024). https://www.doc.ic.ac.uk/~maffeis/papers/raid24.pdf

[7] A Alzubaidi. 2021. Recent advances in android mobile malware detection: A systematic literature review. *IEEE Access* (2021). https://ieeexplore.ieee.org/abstract/document/9585476/

[8] Giuseppina Andresini, Feargus Pendlebury, Fabio Pierazzi, Corrado Loglisci, Annalisa Appice, and Lorenzo Cavallaro. 2021. Insomnia: Towards concept-drift robustness in network intrusion detection. In *Proceedings of the 14th ACM workshop on artificial intelligence and security*. 111–122.

[9] D Angioni, L Demetrio, M Pintor, and B Biggio. 2022. Robust machine learning for malware detection over time. *arXiv preprint arXiv …* (2022). https://arxiv.org/abs/2208.04838

[10] G Apruzzese, P Laskov, and ... 2022. SoK: The impact of unlabelled data in cyberthreat detection. *2022 IEEE 7th European …* (2022). https://ieeexplore.ieee.org/abstract/document/9797356/

[11] G Apruzzese, P Laskov, and ... 2023. SoK: Pragmatic assessment of machine learning for network intrusion detection. *2023 IEEE 8th European …* (2023). https://ieeexplore.ieee.org/abstract/document/10190520/

[12] G Apruzzese, P Laskov, E Montes de Oca, and ... 2023. The role of machine learning in cybersecurity. *… Threats: Research and …* (2023). https://doi.org/10.1145/3545574

[13] Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. 2022. Dos and don'ts of machine learning in computer security. In *31st USENIX Security Symposium (USENIX Security 22)*. 3971–3988.

[14] Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. 2023. Lessons Learned on Machine Learning for Computer Security. *IEEE Security & Privacy* 21, 5 (2023), 72–77.

[15] Federico Barbero, Feargus Pendlebury, Fabio Pierazzi, and Lorenzo Cavallaro. 2022. Transcending transcend: Revisiting malware classification in the presence of concept drift. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 805–823.

[16] F Barr-Smith, X Ugarte-Pedrero, and ... 2021. Survivalism: Systematic analysis of windows malware living-off-the-land. *… IEEE Symposium on …* (2021). https://ieeexplore.ieee.org/abstract/document/9519480/

[17] H Berger, A Dvir, C Hajaj, and R Ronen. 2022. Do you think you can hold me? the real challenge of problem-space evasion attacks. *arXiv preprint arXiv:2205.04293* (2022). https://arxiv.org/abs/2205.04293

[18] H Berger, C Hajaj, and A Dvir. 2020. Evasion is not enough: A case study of android malware. *… on Cyber Security Cryptography and Machine …* (2020). https://doi.org/10.1007/978-3-030-49785-9_11

[19] H Berger, C Hajaj, and A Dvir. 2020. When the guard failed the droid: A case study of android malware. *arXiv preprint arXiv:2003.14123* (2020). https://arxiv.org/abs/2003.14123

[20] H Berger, C Hajaj, E Mariconti, and A Dvir. 2021. Crystal ball: From innovative attacks to attack effectiveness classifier. *IEEE Access* (2021). https://ieeexplore.ieee.org/abstract/document/9663162/

[21] H Berger, C Hajaj, E Mariconti, and A Dvir. 2022. MaMaDroid2.0–The Holes of Control Flow Graphs. *arXiv preprint arXiv:2202.13922* (2022). https://arxiv.org/abs/2202.13922

[22] T Bilot, N El Madhoun, K Al Agha, and A Zouaoui. 2024. A survey on malware detection with graph representation learning. *Comput. Surveys* (2024). https://doi.org/10.1145/3664649

[23] H Bostani and V Moonsamy. 2024. Evadedroid: A practical evasion attack on machine learning for black-box android malware detection. *Computers &Security* (2024). https://www.sciencedirect.com/science/article/pii/S0167404823005862

[24] P Bountakas. 2023. Implementing AI-driven methodologies for cyberattack detection. (2023). https://dione.lib.unipi.gr/xmlui/handle/unipi/15647

[25] P Bountakas, C Ntantogian, and C Xenakis. 2022. EKnad: Exploit Kits' network activity detection. *Future Generation Computer …* (2022). https://www.sciencedirect.com/science/article/pii/S0167739X22001248

[26] H Bullock and M Edwards. 2023. Temporal Constraints in Online Dating Fraud Classification. *ICISSP* (2023). https://www.researchgate.net/profile/Matthew-Edwards-12/publication/369019079_Temporal_Constraints_in_Online_Dating_Fraud_Classification/links/64320b634e83cd0e2f9d3dd8/Temporal-Constraints-in-Online-Dating-Fraud-Classification.pdf

[27] M Cao. 2022. Understanding the characteristics of invasive malware from the Google Play Store. (2022). https://open.library.ubc.ca/soa/cIRcle/collections/ubctheses/24/items/1.0406233

[28] M Cao, S Badihi, K Ahmed, P Xiong, and ... 2020. On benign features in malware detection. *Proceedings of the 35th …* (2020). https://doi.org/10.1145/3324884.3418926

[29] Lorenzo Cavallaro and Emiliano De Cristofaro. 2023. Security and Privacy of AI Knowledge Guide Issue 1.0. 0. (2023).

[30] Lorenzo Cavallaro, Johannes Kinder, Feargus Pendlebury, and Fabio Pierazzi. 2023. Are machine learning models for malware detection ready for prime time? *IEEE Security & Privacy* 21, 2 (2023), 53–56.

[31] F Ceschin, M Botacin, A Bifet, B Pfahringer, and ... 2024. Machine learning (in) security: A stream of problems. *… Threats: Research and …* (2024). https://doi.org/10.1145/3617897

[32] X Chen. 2020. Studying the Security Centric Intelligence on Android Malware Detection. (2020). https://researchbank.swinburne.edu.au/items/592de582-765e-440b-90ca-c8fbab6b4e8a/1/xiao_chen_thesis.pdf

[33] X Chen. 2021. Intrusion Response via Graph-Based Low-Level System Event Analysis. (2021). https://search.proquest.com/openview/a08a3b5d2880223f765e33f76f5620d3/1?pq-origsite=gscholar&cbl=18750&diss=y

[34] Yizheng Chen, Zhoujie Ding, and David Wagner. 2023. Continuous Learning for Android Malware Detection. In *USENIX Security Symposium*.

[35] YH Chen, SC Lin, SC Huang, CL Lei, and ... 2023. Guided malware sample analysis based on graph neural networks. *IEEE Transactions on …* (2023). https://ieeexplore.ieee.org/abstract/document/10145856/

[36] Zhi Chen, Zhenning Zhang, Zeliang Kan, Limin Yang, Jacopo Cortellazzi, Feargus Pendlebury, Fabio Pierazzi, Lorenzo Cavallaro, and Gang Wang. 2023. Is It Overkill? Analyzing Feature-Space Concept Drift in Malware Detectors. In *2023 IEEE Deep Learning Security and Privacy Workshop (DLSP). IEEE.*

[37] François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

[38] Theo Chow, Zeliang Kan, Lorenz Linhardt, Daniel Arp, Lorenzo Cavallaro, and Fabio Pierazzi. 2023. Drift Forensics of Malware Classifiers. In *Proc. of the ACM Workshop on Artificial Intelligence and Security (AISec)*. ACM.

[39] M Chowdhury, B Ray, S Chowdhury, and ... 2021. A novel insider attack and machine learning based detection for the internet of things. *ACM Transactions on …* (2021). https://doi.org/10.1145/3466721

[40] AE Cina. 2023. Vulnerability of Machine Learning: A Study on Poisoning Attacks. (2023). http://dspace.unive.it/handle/10579/23716

[41] ML Colangelo. 2023. Malware family classification with semi-supervised learning. (2023). https://webthesis.biblio.polito.it/29460/

[42] A Corsini, SJ Yang, and G Apruzzese. 2021. On the evaluation of sequential machine learning for network intrusion detection. *Proceedings of the 16th International …* (2021). https://doi.org/10.1145/3465481.3470065

[43] R Coulter. 2021. Towards Developing Data-Driven Detection Methods for Advanced Persistent Threats In Multi-Domains. (2021). https://researchbank.swinburne.edu.au/items/70b8d545-7e79-42ae-bd33-a593d7224314/1/Rory_Coulter_Thesis.pdf

[44] L Cui, J Cui, Y Ji, Z Hao, L Li, and Z Ding. 2023. Api2vec: Learning representations of api sequences for malware detection. *Proceedings of the 32nd ACM …* (2023). https://doi.org/10.1145/3597926.3598054

[45] N Daoudi, K Allix, TF Bissyandé, and J Klein. 2021. Lessons learnt on reproducibility in machine learning based android malware detection. *Empirical Software Engineering* (2021). https://doi.org/10.1007/s10664-021-09955-7

[46] M D'Onghia, F Di Cesare, L Gallo, M Carminati, and ... 2023. Lookin'Out My Backdoor! Investigating Backdooring Attacks Against DL-driven Malware Detectors. *Proceedings of the 16th …* (2023). https://doi.org/10.1145/3605764.3623919

[47] A Drichel, M Meyer, and U Meyer. 2024. Towards Robust Domain Generation Algorithm Classification. *Proceedings of the 19th ACM Asia …* (2024). https://doi.org/10.1145/3634737.3656287

[48] A Duby, T Taylor, G Bloom, and ... 2022. Detecting and classifying self-deleting windows malware using prefetch files. *2022 IEEE 12th Annual …* (2022). https://ieeexplore.ieee.org/abstract/document/9720874/

[49] A Duby, T Taylor, and Y Zhuang. 2022. Malware family classification via residual prefetch artifacts. *2022 IEEE 19th Annual …* (2022). https://ieeexplore.ieee.org/abstract/document/9700530/

[50] MR Ebrahimi, W Li, Y Chai, J Pacheco, and ... 2022. An Adversarial Reinforcement Learning Framework for Robust Machine Learning-based Malware Detection. *… Conference on Data …* (2022). https://ieeexplore.ieee.org/abstract/document/10031124/

[51] T Van Ede, H Aghakhani, N Spahn, and ... 2022. Deepcase: Semi-supervised contextual analysis of security events. *… IEEE Symposium on …* (2022). https://ieeexplore.ieee.org/abstract/document/9833671/

[52] ELSA EU. 2024. ELSA Benchmarks Platform - Cybersecurity. https://benchmarks.elsa-ai.eu/?ch=6&com=introduction. [Accessed 11-09-2024].

[53] DMA FAHIM. 2021. A Critique of Android Malware Classification Systems. (2021). https://e-archivo.uc3m.es/bitstream/handle/10016/35054/tesis_mohammed_ahmed_fahim_rashed_2022.pdf?sequence=1

[54] Y Fan, T Shibahara, Y Ohsita, D Chiba, and ... 2021. Understanding update of machine-learning-based malware detection by clustering changes in feature attributions. *Advances in Information …* (2021). https://doi.org/10.1007/978-3-030-85987-9_6

[55] M Fleming and O Olukoya. 2024. A temporal analysis and evaluation of fuzzy hashing algorithms for Android malware analysis. *Forensic Science International: Digital Investigation* (2024). https://www.sciencedirect.com/science/article/pii/S2666281724000891

[56] C Gao, G Huang, H Li, B Wu, Y Wu, and ... 2024. A Comprehensive Study of Learning-based Android Malware Detectors under Challenging Environments. *Proceedings of the 46th …* (2024). https://doi.org/10.1145/3597503.3623320

[57] X Ge, Y Huang, Z Hui, X Wang, and ... 2021. Impact of datasets on machine learning based methods in android malware detection: an empirical study. *2021 IEEE 21st …* (2021). https://ieeexplore.ieee.org/abstract/document/9724975/

[58] L Gitzinger. 2020. Surviving the massive proliferation of mobile malware. (2020). https://theses.hal.science/tel-03194472/

[59] Y Gu and L Li. 2021. Graphevolvedroid: Mitigate model degradation in the scenario of android ecosystem evolution. *Proceedings of the 30th ACM International Conference …* (2021). https://doi.org/10.1145/3459637.3482118

[60] A Guerra-Manzanares. 2023. Android malware detection: mission accomplished? A review of open challenges and future perspectives. *Computers &Security* (2023). https://www.sciencedirect.com/science/article/pii/S0167404823005631

[61] A Guerra-Manzanares, H Bahsi, and M Luckner. 2023. Leveraging the first line of defense: A study on the evolution and usage of android security permissions for enhanced android malware detection. *Journal of Computer Virology ...* (2023). https://doi.org/10.1007/s11416-022-00432-3

[62] A Guerra-Manzanares, M Luckner, and H Bahsi. 2022. Concept drift and cross-device behavior: Challenges and implications for effective android malware detection. *Computers &Security* (2022). https://www.sciencedirect.com/science/article/pii/S0167404822001523

[63] W Guo. 2022. Explainable AI Techniques for Security. (2022). https://etda.libraries.psu.edu/catalog/22962wzg13

[64] P He, Y Xia, X Zhang, and S Ji. 2023. Efficient query-based attack against ML-based Android malware detection under zero knowledge setting. *Proceedings of the 2023 ACM SIGSAC ...* (2023). https://doi.org/10.1145/3576915.3623117

[65] Y He, Y Liu, L Wu, Z Yang, K Ren, and ... 2022. MsDroid: Identifying Malicious Snippets for Android Malware Detection. *IEEE Transactions on ...* (2022). https://ieeexplore.ieee.org/abstract/document/9762803/

[66] H Hindy, D Brosset, E Bayne, AK Seeam, and ... 2020. A taxonomy of network threats and the effect of current datasets on intrusion detection systems. *IEEE ...* (2020). https://ieeexplore.ieee.org/abstract/document/9108270/

[67] G Ho. 2020. Thwarting Sophisticated Enterprise Attacks: Data-Driven Methods and Insights. (2020). https://search.proquest.com/openview/fbd088a79a2f5ddca82699ed58f34a33/1?pq-origsite=gscholar&cbl=18750&diss=y

[68] H Hu, Y Liu, Y Zhao, Y Liu, X Sun, and ... 2023. Detecting Temporal Inconsistency in Biased Datasets for Android Malware Detection. *2023 38th IEEE/ACM ...* (2023). https://ieeexplore.ieee.org/abstract/document/10298755/

[69] T Hu. 2020. Detecting Bots using Stream-based System with Data Synthesis. (2020). https://vtechworks.lib.vt.edu/items/a2d54e17-2c80-4233-b5bf-2546577fa344

[70] JE Charlton III. 2021. Inferring Malware Detector Metrics in the Absence of Ground-Truth. (2021). https://search.proquest.com/openview/db28c3e7c29ecfbbbacb63af2fd03900/1?pq-origsite=gscholar&cbl=18750&diss=y

[71] STK Jan. 2020. Robustifying Machine Learning based Security Applications. (2020). https://vtechworks.lib.vt.edu/items/05c52c7f-7d18-43c0-889a-00b2a441a2d3

[72] STK Jan, Q Hao, T Hu, J Pu, S Oswal, and ... 2020. Throwing darts in the dark? detecting bots with limited data using neural data augmentation. *... IEEE symposium on ...* (2020). https://ieeexplore.ieee.org/abstract/document/9152805/

[73] C Jiang, C Xia, M Liu, C Chen, H Li, and ... 2024. FedDRC: A Robust Federated Learning-based Android Malware Classifier under Heterogeneous Distribution. *... Cooperative Work in ...* (2024). https://ieeexplore.ieee.org/abstract/document/10580300/

[74] C Jiang, K Yin, C Xia, and W Huang. 2022. Fedhgcdroid: An adaptive multi-dimensional federated learning for privacy-preserving android malware classification. *Entropy* (2022). https://www.mdpi.com/1099-4300/24/7/919

[75] TS John, T Thomas, and S Emmanuel. 2020. Graph convolutional networks for android malware detection with system call graphs. *2020 Third ISEA ...* (2020). https://ieeexplore.ieee.org/abstract/document/9079308/

[76] R Jordaney and L Cavallaro. 2019. Machine Learning Techniques for Evolving Threats. (2019). https://pure.royalholloway.ac.uk/ws/portalfiles/portal/33898189/2019jordaneyrphd.pdf

[77] Roberto Jordaney, Kumar Sharad, Santanu Kumar Dash, Zhi Wang, Davide Papini, Ilia Nouretdinov, and Lorenzo Cavallaro. 2017. Transcend: Detecting Concept Drift in Malware Classification Models. In *USENIX Security*.

[78] Zeliang Kan, Shae McFadden, Daniel Arp, Feargus Pendlebury, Roberto Jordaney, Johannes Kinder, Fabio Pierazzi, and Lorenzo Cavallaro. 2024. TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time (Extended Version). *arXiv preprint arXiv:2402.01359* (2024).

[79] Zeliang Kan, Feargus Pendlebury, Fabio Pierazzi, and Lorenzo Cavallaro. 2021. Investigating labelless drift adaptation for malware detection. In *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*. 123–134.

[80] S Karapoola, N Singh, C Rebeiro, and ... 2024. : A Case-Sensitive Detection Engine to Counter Malware Diversity. *IEEE Transactions on ...* (2024). https://ieeexplore.ieee.org/abstract/document/10541049/

[81] S Karapoola, N Singh, and C Rebeiro. 2022. SUNDEW: An Ensemble of Predictors for Case-Sensitive Detection of Malware. *arXiv preprint arXiv:2211.06153* (2022). https://arxiv.org/abs/2211.06153

[82] M Kim, W Jang, JN Hur, and MK Yoon. 2024. Relative Frequency-Rank Encoding for Unsupervised Network Anomaly Detection. *IEEE/ACM Transactions on ...* (2024). https://ieeexplore.ieee.org/abstract/document/10517994/

[83] D Kleidermacher, E Arriaga, E Wang, S Porst, and ... 2024. Security and Privacy Product Inclusion. *arXiv preprint arXiv ...* (2024). https://arxiv.org/abs/2404.13220

[84] T Komárek. 2023. Učení Modelů na Datech se Složitou Strukturou s Aplikacemi pro Kybernetickou Bezpečnost. (2023). https://search.proquest.com/openview/542016b540bea426b1b3a03813c94dcc/1?pq-origsite=gscholar&cbl=2026366&diss=y

[85] B Kondracki. 2023. Leveraging Side-Channels to Fingerprint Software Systems. (2023). https://search.proquest.com/openview/10747e66849a1927d16c98314879f4aa/1?pq-origsite=gscholar&cbl=18750&diss=y

[86] DF Koranek. 2022. Evaluating Deep Learning Explanations on RISC-V Assembly as a Reverse Engineering Aid. (2022). https://scholar.afit.edu/etd/6900/

[87] P Krishnan, C Cifuentes, L Li, and ... 2023. Why Is Static Application Security Testing Hard to Learn? *IEEE Security & ...* (2023). https://ieeexplore.ieee.org/abstract/document/10242233/

[88] S Kumar, D Mishra, B Panda, and ... 2021. Deepdetect: A practical on-device android malware detector. *2021 IEEE 21st ...* (2021). https://ieeexplore.ieee.org/abstract/document/9724811/

[89] Y Kurogome, Y Otsuki, Y Kawakoya, and ... 2019. EIGER: automated IOC generation for accurate and interpretable endpoint malware detection. *Proceedings of the 35th ...* (2019). https://doi.org/10.1145/3359789.3359808

[90] Raphael Labaca-Castro, Luis Muñoz-González, Feargus Pendlebury, Gabi Dreo Rodosek, Fabio Pierazzi, and Lorenzo Cavallaro. 2021. Realizable universal adversarial perturbations for malware. *arXiv preprint arXiv:2102.06747* (2021).

[91] S Layton, T Tucker, D Olszewski, K Warren, and ... 2024. {SoK}: The Good, The Bad, and The Unbalanced: Measuring Structural Limitations of Deepfake Media Datasets. *33rd USENIX Security ...* (2024). https://www.usenix.org/conference/usenixsecurity24/presentation/layton

[92] C Lee and S Son. 2023. AdCPG: Classifying JavaScript Code Property Graphs with Explanations for Ad and Tracker Blocking. *Proceedings of the 2023 ACM SIGSAC Conference on ...* (2023). https://doi.org/10.1145/3576915.3623084

[93] H Lee, S Cho, H Han, W Cho, and ... 2022. Enhancing Sustainability in Machine Learning-based Android Malware Detection using API calls. *2022 IEEE Fifth ...* (2022). https://ieeexplore.ieee.org/abstract/document/9939276/

[94] H Lesfari. 2022. Foundations of networks towards AI. (2022). https://theses.hal.science/tel-04060601/

[95] KU Leuven. [n. d.]. Summer School on Security & Privacy in the Age of AI 2024. https://cybersecurity-research.be/summer-school-on-security-privacy-in-the-age-of-ai-2024. [Accessed 11-09-2024].

[96] C Li, Q Lv, N Li, Y Wang, D Sun, and Y Qiao. 2022. A novel deep framework for dynamic malware detection based on API sequence intrinsic features. *Computers &Security* (2022). https://www.sciencedirect.com/science/article/pii/S0167404822000840

[97] D Li, T Qiu, S Chen, Q Li, and S Xu. 2021. Can we leverage predictive uncertainty to detect dataset shift and adversarial examples in android malware detection? *... of the 37th Annual Computer Security ...* (2021). https://doi.org/10.1145/3485832.3485916

[98] H Li, Z Cheng, B Wu, L Yuan, C Gao, W Yuan, and ... 2023. Black-box Adversarial Example Attack towards {FCG} Based Android Malware Detection under Incomplete Feature Information. *32nd USENIX Security ...* (2023). https://www.usenix.org/conference/usenixsecurity23/presentation/li-heng

[99] T Lihong. 2022. Evolutionary Study of Android Apps. (2022). https://researchbank.swinburne.edu.au/file/9bd83420-2192-4532-b630-15e5c4679ba2/1/lihong_tang_thesis.pdf

[100] C Liu, B Li, J Zhao, W Feng, X Liu, and ... 2023. A2-CLM: Few-Shot Malware Detection Based on Adversarial Heterogeneous Graph Augmentation. *IEEE Transactions on ...* (2023). https://ieeexplore.ieee.org/abstract/document/10368085/

[101] J Liu, J Zeng, F Pierazzi, L Cavallaro, and ... 2024. Unraveling the Key of Machine Learning Solutions for Android Malware Detection. *arXiv preprint arXiv ...* (2024). https://arxiv.org/abs/2402.02953

[102] K Liu, S Xu, G Xu, M Zhang, D Sun, and H Liu. 2020. A review of android malware detection approaches based on machine learning. *IEEE access* (2020). https://ieeexplore.ieee.org/abstract/document/9130686/

[103] Y Liu, C Tantithamthavorn, L Li, and Y Liu. 2022. Deep learning for android malware defenses: a systematic literature review. *Comput. Surveys* (2022). https://doi.org/10.1145/3544968

[104] Imperial College London. 2024. Machine Learning and Cyber Security Symposium @ Imperial 2024. https://ml-css.cybersec.fun. [Accessed 11-09-2024].

[105] University College London. 2024. Malware (COMP0060). https://www.ucl.ac.uk/module-catalogue/modules/malware-COMP0060. [Accessed 11-09-2024].

[106] S Lounici. 2022. Watermarking machine learning models. (2022). https://theses.hal.science/tel-04091291/

[107] P Maniriho, AN Mahmood, and MJM Chowdhury. 2023. API-MalDetect: Automated malware detection framework for windows based on API calls and deep learning techniques. *Journal of Network and ...* (2023). https://www.sciencedirect.com/science/article/pii/S1084804523001236

[108] P Maniriho, AN Mahmood, and MJM Chowdhury. 2023. A systematic literature review on Windows malware detection: Techniques, research issues, and future directions. *Journal of Systems and ...* (2023). https://www.sciencedirect.com/

science/article/pii/S0164121223003163
[109] P Maniriho, AN Mahmood, and MJM Chowdhury. 2024. MeMalDet: A memory analysis-based malware detection framework using deep autoencoders and stacked ensemble under temporal evaluations. *Computers &Security* (2024). https://www.sciencedirect.com/science/article/pii/S0167404824001652
[110] P Maniriho, AN Mahmood, and MJM Chowdhury. 2024. A survey of recent advances in deep learning models for detecting malware in desktop and mobile platforms. *Comput. Surveys* (2024). https://doi.org/10.1145/3638240
[111] Shae McFadden, Mark Kan, Lorenzo Cavallaro, and Fabio Pierazzi. 2024. The Impact of Active Learning on Availability Data Poisoning for Android Malware Classifiers. In *Proceedings of the Annual Computer Security Applications Conference Workshops (ACSAC Workshops)*. IEEE.
[112] Shae McFadden, Zeliang Kan, Lorenzo Cavallaro, and Fabio Pierazzi. 2023. Poster: RPAL-Recovering Malware Classifiers from Data Poisoning using Active Learning. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 3561–3563.
[113] A Mesbah, I Baddari, and MA Riahla. 2023. LONGCGDROID: ANDROID MALWARE DETECTION THROUGH LONGITUDINAL STUDY FOR MACHINE LEARNING AND DEEP LEARNING. *Jordanian Journal of ...* (2023). https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=24139351&AN=174483473&h=RYNyibZov8nAWij%2BLdG1D9Ys3ZIdBaPhElgfxxhKDbDiYnamBPoECwjVW4x9gnXTlXYJbwhY%2FSr%2FILHsT1Bl4Q%3D%3D&crl=c
[114] TJC Miranda. 2022. Profiling and Visualizing Android Malware Datasets. (2022). https://theses.hal.science/tel-04003806/
[115] TC Miranda, PF Gimenez, JF Lalande, and ... 2022. Debiasing android malware datasets: How can i trust your results if your dataset is biased? *IEEE Transactions ...* (2022). https://ieeexplore.ieee.org/abstract/document/9787514/
[116] I Moisejevs. [n. d.]. Adversarial Attacks and Defenses in Malware Classification: A Survey. ([n. d.]).
[117] B Molina-Coronado, U Mori, A Mendiburu, and ... 2023. Efficient concept drift handling for batch android malware detection models. *Pervasive and Mobile ...* (2023). https://www.sciencedirect.com/science/article/pii/S1574119223001074
[118] A Nadeem, V Rimmer, W Joosen, and S Verwer. [n. d.]. Intelligent Malware Defenses: A Survey. *azqanadeem.github.io* ([n. d.]). https://azqanadeem.github.io/assets/pdf/papers/Nadeem2022_Chapter_IntelligentMalwareDefenses.pdf
[119] A Nadeem, V Rimmer, W Joosen, and S Verwer. 2022. Intelligent malware defenses. *Security and artificial ...* (2022). https://doi.org/10.1007/978-3-030-98795-4_10
[120] C Nader. 2022. Identifying IoT Devices Behind a NAT by Using Empirical Data and Learning Methods. (2022). https://search.proquest.com/openview/ccf38d517b22b3a230a366e1ecb4ca57/1?pq-origsite=gscholar&cbl=18750&diss=y
[121] TG Nguyen, T Le-Cong, HJ Kang, and ... 2023. Multi-granularity detector for vulnerability fixes. *IEEE Transactions ...* (2023). https://ieeexplore.ieee.org/abstract/document/10138621/
[122] JJ Nielen. 2023. Dynamic Detection and Classification of Persistence Techniques in Windows Malware. (2023). http://essay.utwente.nl/94945/
[123] U Nisslmueller. 2022. LOLBin detection through unsupervised learning: An approach based on explicit featurization of the command line and parent-child relationships. (2022). http://essay.utwente.nl/93265/
[124] M Noppel, L Peter, and ... 2023. Disguising attacks with explanation-aware backdoors. *2023 IEEE Symposium on ...* (2023). https://ieeexplore.ieee.org/abstract/document/10179308/
[125] M Noppel, L Peter, and C Wressnegger. 2022. Backdooring explainable machine learning. *arXiv preprint arXiv:2204.09498* (2022). https://arxiv.org/abs/2204.09498
[126] D Olszewski, A Lu, C Stillman, K Warren, and ... 2023. " Get in Researchers; We're Measuring Reproducibility": A Reproducibility Study of Machine Learning Papers in Tier 1 Security Conferences. *Proceedings of the ...* (2023). https://doi.org/10.1145/3576915.3623130
[127] IC Olteanu. 2022. Evaluating the response effectiveness of XDR technology in a scaled down environment. (2022). https://research.tue.nl/files/305661196/Olteanu_I.C..pdf
[128] L Onwuzurike. 2019. Measuring and Mitigating Security and Privacy Issues on Android Applications. (2019). https://discovery.ucl.ac.uk/id/eprint/10066602/
[129] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. https://pytorch.org/.
[130] P Pathak. 2021. Leveraging attention-based deep neural networks for security vetting of Android applications. (2021). https://rave.ohiolink.edu/etdc/view?acc_num=bgsu1617215079338328
[131] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-Learn: Machine Learning in Python. *JMLR* (2011).
[132] F Pendlebury. 2021. Machine Learning for Security in Hostile Environments. (2021). https://core.ac.uk/download/pdf/492540975.pdf
[133] Feargus Pendlebury, Fabio Pierazzi, Roberto Jordaney, Johannes Kinder, and Lorenzo Cavallaro. 2019. {TESSERACT}: Eliminating experimental bias in malware classification across space and time. In *28th USENIX Security Symposium (USENIX Security 19)*. 729–746.
[134] Fabio Pierazzi. 2024. Exercises for practicing MLSec for Systems Security. https://github.com/isneslab/mlsec-labs. [Accessed 11-09-2024].
[135] Fabio Pierazzi. 2024. Ph.D. Course at UniBo (2024). https://fabio.pierazzi.com/courses/mlsec-unibo-2024. [Accessed 11-09-2024].
[136] Fabio Pierazzi, Stefano Cristalli, Danilo Bruschi, Michele Colajanni, Mirco Marchetti, and Andrea Lanzi. 2020. Glyph: Efficient ML-based detection of heap spraying attacks. *IEEE Transactions on Information Forensics and Security* 16 (2020), 740–755.
[137] Fabio Pierazzi, Ghita Mezzour, Qian Han, Michele Colajanni, and VS Subrahmanian. 2020. A data-driven characterization of modern Android spyware. *ACM Transactions on Management Information Systems (TMIS)* 11, 1 (2020), 1–38.
[138] Fabio Pierazzi, Feargus Pendlebury, Jacopo Cortellazzi, and Lorenzo Cavallaro. 2020. Intriguing properties of adversarial ml attacks in the problem space. In *2020 IEEE symposium on security and privacy (SP)*. IEEE, 1332–1349.
[139] TSR Pimenta, F Ceschin, and A Gregio. 2024. Androidgyny: Reviewing clustering techniques for Android malware family classification. *Digital Threats: Research and ...* (2024). https://doi.org/10.1145/3587471
[140] DJ Plohmann. 2022. Classification, Characterization, and Contextualization of Windows Malware using Static Behavior and Similarity Analysis. (2022). https://bonndoc.ulb.uni-bonn.de/xmlui/handle/20.500.11811/9992
[141] P Poudel. 2021. Security Vetting Of Android Applications Using Graph Based Deep Learning Approaches. (2021). https://rave.ohiolink.edu/etdc/view?acc_num=bgsu1617199500076786
[142] R Purwanto. 2022. Adaptive Phishing Detection System using Machine Learning. (2022). https://unsworks.unsw.edu.au/server/api/core/bitstreams/5fab4050-695b-4a58-a5ad-4d9edd582f1f/content
[143] RW Purwanto, A Pal, A Blair, and ... 2022. PhishSim: aiding phishing website detection with a feature-free tool. *IEEE Transactions on ...* (2022). https://ieeexplore.ieee.org/abstract/document/9745933/
[144] R Purwanto, A Pal, A Blair, and S Jha. 2020. PhishZip: A new compression-based algorithm for detecting phishing websites. *2020 IEEE Conference on ...* (2020). https://ieeexplore.ieee.org/abstract/document/9162211/
[145] G Puts. [n. d.]. Improving Anomaly-Based Intrusion Detection for IT Networks. *research.tue.nl* ([n. d.]). https://research.tue.nl/files/333615728/Puts_G.pdf
[146] Y Qing, Q Yin, X Deng, Y Chen, Z Liu, K Sun, and ... 2023. Low-quality training data only? A robust framework for detecting encrypted malicious network traffic. *arXiv preprint arXiv ...* (2023). https://arxiv.org/abs/2309.04798
[147] J Qiu, J Zhang, W Luo, L Pan, S Nepal, and ... 2020. A survey of android malware detection with deep neural models. *ACM Computing Surveys ...* (2020). https://doi.org/10.1145/3417978
[148] PHN Rajput. 2023. Hardware-Assisted Non-Intrusive Security Controls for Modern Industrial Control Systems. (2023). https://search.proquest.com/openview/7cc6179975efd6097d5106212c1e0d58/1?pq-origsite=gscholar&cbl=18750&diss=y
[149] A Rashid. 2023. Exploring Defenses Against Adversarial Attacks in Machine Learning-Based Malware Detection. (2023). https://kclpure.kcl.ac.uk/portal/files/229374552/2023_Rashid_Aqib_1515232_ethesis.pdf
[150] K Rendall, A Mylonas, and S Vidalis. 2022. Toward situational awareness in threat detection. A survey. *... Reviews: Forensic Science* (2022). https://doi.org/10.1002/wfs2.1448
[151] M Rhode. 2021. Racing demons: Malware detection in early execution. (2021). https://orca.cardiff.ac.uk/id/eprint/151083/
[152] M Rhode, P Burnap, and A Wedgbury. 2021. Real-Time Malware Process Detection and Automated Process Killing. *Security and ...* (2021). https://doi.org/10.1155/2021/8933681
[153] TM Roelofs, E Barbaro, S Pekarskikh, and ... 2024. Finding Harmony in the Noise: Blending Security Alerts for Attack Detection. *Proceedings of the 39th ...* (2024). https://doi.org/10.1145/3605098.3635981
[154] O Roques, S Maffeis, and M Cova. 2019. Detecting malware in TLS traffic. (2019). https://www.imperial.ac.uk/media/imperial-college/faculty-of-engineering/computing/public/1819-pg-projects/Detecting-Malware-in-TLS-Traf%EF%AC%81c.pdf
[155] A Salem. 2021. Accurate and Reliable Labeling for Effective Detection of Android Malware. (2021). https://mediatum.ub.tum.de/1574488
[156] M Scalas. 2021. Malware analysis and detection with explainable machine learning. (2021). https://iris.unica.it/handle/11584/310630
[157] LK Shar, BF Demissie, M Ceccato, YN Tun, and ... 2023. Experimental comparison of features, analyses, and classifiers for Android malware detection. *Empirical Software ...* (2023). https://doi.org/10.1007/s10664-023-10375-y
[158] X She, Y Liu, Y Zhao, Y He, L Li, and ... 2023. Pitfalls in language models for code intelligence: A taxonomy and survey. *arXiv preprint arXiv ...* (2023). https://arxiv.org/abs/2310.17903

[159] Y Shen and G Stringhini. 2021. ANDRUSPEX: leveraging graph representation learning to predict harmful app installations on mobile devices. *2021 IEEE European Symposium on …* (2021). https://ieeexplore.ieee.org/abstract/document/9581262/

[160] L Shu, S Dong, H Su, and J Huang. 2023. Android malware detection methods based on convolutional neural network: A survey. *IEEE Transactions on …* (2023). https://ieeexplore.ieee.org/abstract/document/10153773/

[161] A Singh, M Tanha, Y Girdhar, and A Hunter. 2024. Interpretable Android Malware Detection Based on Dynamic Analysis. *ICISSP* (2024). https://www.scitepress.org/Papers/2024/124158/124158.pdf

[162] P Singh. [n. d.]. Detection of Malicious OOXML Documents Using Domain Specific Features. *researchgate.net* ([n. d.]). https://www.researchgate.net/profile/Priyansh-Singh-5/publication/362620918_Detection_of_Malicious_OOXML_Documents_Using_Domain_Specific_Features/links/62f4cc0952130a3cd71338cf/Detection-of-Malicious-OOXML-Documents-Using-Domain-Specific-Features.pdf

[163] D Soi, A Sanna, D Maiorca, and G Giacinto. 2024. Enhancing android malware detection explainability through function call graph APIs. *Journal of Information Security and …* (2024). https://www.sciencedirect.com/science/article/pii/S2214212623002752

[164] B Soman, A Torkamani, MJ Morais, J Bickford, and ... 2022. Firenze: Model evaluation using weak signals. *arXiv preprint arXiv …* (2022). https://arxiv.org/abs/2207.00827

[165] OP Suciu. 2021. Data-Driven Techniques for Vulnerability Assessments. (2021). https://search.proquest.com/openview/2143c3964f9515176b7d85605e82a36b/1?pq-origsite=gscholar&cbl=18750&diss=y

[166] R Surendran, T Thomas, and ... 2020. On Existence of Common Malicious System Call Codes in Android Malware Families. *IEEE Transactions on …* (2020). https://ieeexplore.ieee.org/abstract/document/9069305/

[167] T Teesselink. 2019. Identifying Application Phases in Mobile Encrypted Network Traffic. (2019). https://essay.utwente.nl/79732/

[168] W Thompson, T Elammas, M Kalappattil, and ... 2024. Malware Dataset Availability &Inherent Bias Study. *2024 IEEE …* (2024). https://ieeexplore.ieee.org/abstract/document/10609911/

[169] G Di Tizio. 2023. Leveraging Security Data for a Quantitative Evaluation of Security Mitigation Strategies. (2023). https://iris.unitn.it/handle/11572/374972

[170] H Turtiainen, A Costin, and T Hämäläinen. 2022. Defensive machine learning methods and the cyber defence chain. *Artificial Intelligence and …* (2022). https://doi.org/10.1007/978-3-031-15030-2_7

[171] R Vaarandi and A Guerra-Manzanares. 2024. Network IDS alert classification with active learning techniques. *Journal of Information Security and …* (2024). https://www.sciencedirect.com/science/article/pii/S2214212623002715

[172] T van Ede, N Khasuntsev, B Steen, and ... 2022. Detecting Anomalous Misconfigurations in AWS Identity and Access Management Policies. *Proceedings of the 2022 …* (2022). https://doi.org/10.1145/3560810.3564264

[173] X Wei, C Li, Q Lv, N Li, D Sun, and Y Wang. 2024. Mitigating the Impact of Malware Evolution on API Sequence-based Windows Malware Detector. *arXiv preprint arXiv:2408.01661* (2024). https://arxiv.org/abs/2408.01661

[174] Q Wu, X Zhu, and B Liu. 2021. A survey of android malware static detection technology based on machine learning. *Mobile Information Systems* (2021). https://doi.org/10.1155/2021/8896013

[175] J XU. 2021. Machine learning based approaches towards robust Android malware detection. (2021). https://ink.library.smu.edu.sg/etd_coll/320/

[176] J Xu, Y Li, RH Deng, and K Xu. 2020. Sdac: A slow-aging solution for android malware detection using semantic distance based api clustering. *IEEE Transactions on Dependable …* (2020). https://ieeexplore.ieee.org/abstract/document/9127784/

[177] P Xu. 2021. Android-coco: Android malware detection with graph neural network for byte-and native-code. *arXiv preprint arXiv:2112.10038* (2021). https://arxiv.org/abs/2112.10038

[178] P Xu, C Eckert, and A Zarras. 2021. Hybrid-falcon: Hybrid pattern malware detection and categorization with network traffic and program code. *arXiv preprint arXiv:2112.10035* (2021). https://arxiv.org/abs/2112.10035

[179] Y Xu, D Li, Q Li, and S Xu. 2023. Malware Evasion Attacks Against IoT and Other Devices: An Empirical Study. *Tsinghua Science and Technology* (2023). https://ieeexplore.ieee.org/abstract/document/10225279/

[180] S Yan, J Ren, W Wang, L Sun, and ... 2022. A survey of adversarial attack and defense methods for malware classification in cyber security. *… Surveys &Tutorials* (2022). https://ieeexplore.ieee.org/abstract/document/9964330/

[181] L Yang. 2023. Machine learning for security applications under dynamic and adversarial environments. (2023). https://www.ideals.illinois.edu/items/129213

[182] L Yang, Z Chen, J Cortellazzi, and ... 2023. Jigsaw puzzle: Selective backdoor attack to subvert malware classifiers. *… IEEE Symposium on …* (2023). https://ieeexplore.ieee.org/abstract/document/10179347/

[183] L Yang, W Guo, Q Hao, A Ciptadi, and ... 2021. {CADE}: Detecting and explaining concept drift samples for security applications. *30th USENIX Security …* (2021). https://www.usenix.org/conference/usenixsecurity21/presentation/yang-limin

[184] R Yang, X Chen, H Xu, Y Cheng, and ... 2020. RATScope: Recording and Reconstructing Missing RAT Semantic Behaviors for Forensic Analysis on Windows. *… on Dependable and …* (2020). https://ieeexplore.ieee.org/abstract/document/9234076/

[185] S Yang, J Cao, H Zeng, B Shen, and ... 2021. Locating faulty methods with a mixed RNN and attention model. *2021 IEEE/ACM 29th …* (2021). https://ieeexplore.ieee.org/abstract/document/9462960/

[186] Y Yang, B Yuan, J Lou, and Z Qin. 2024. SCRR: Stable Malware Detection under Unknown Deployment Environment Shift by Decoupled Spurious Correlations Filtering. *IEEE Transactions on …* (2024). https://ieeexplore.ieee.org/abstract/document/10444904/

[187] M Yousefiazar. 2020. Machine learning for automatic malware representation and analysis. (2020). https://figshare.mq.edu.au/ndownloader/files/38222790

[188] L Yuan, H Li, B Xia, C Gao, M Liu, W Yuan, and X You. 2022. Recent Advances in Concept Drift Adaptation Methods for Deep Learning. *IJCAI* (2022). https://www.ijcai.org/proceedings/2022/0788.pdf

[189] F Yun. [n. d.]. Understanding Machine Learning Model Updates in Malware Detection Systems Based on Feature Attribution Changes. *www-mura.ist.osaka-u.ac.jp* ([n. d.]). https://www-mura.ist.osaka-u.ac.jp/achievements/annual_report/web2020/paper/h-un21master_thesis-UnderstandingMLupdates.pdf

[190] G Zhang, H Li, Z Chen, L Peng, Y Zhu, and ... 2021. AndroCreme: Unseen Android Malware Detection Based on Inductive Conformal Learning. *2021 IEEE 20th …* (2021). https://ieeexplore.ieee.org/abstract/document/9724463/

[191] H Zhang, L Zhao, A Yu, L Cai, and ... 2024. Ranker: Early Ransomware Detection through Kernel-level Behavioral Analysis. *IEEE Transactions on …* (2024). https://ieeexplore.ieee.org/abstract/document/10551299/

[192] J Zhang, L Pan, QL Han, C Chen, and ... 2021. Deep learning based attack detection for cyber-physical system cybersecurity: A survey. *IEEE/CAA Journal of …* (2021). https://ieeexplore.ieee.org/abstract/document/9536650/

[193] X Zhang, M Zhang, Y Zhang, M Zhong, and ... 2022. Slowing down the aging of learning-based malware detectors with api knowledge. *… on Dependable and …* (2022). https://ieeexplore.ieee.org/abstract/document/9693181/

[194] X Zhang, Y Zhang, M Zhong, D Ding, Y Cao, and ... 2020. Enhancing state-of-the-art classifiers with api semantics to detect evolved android malware. *Proceedings of the …* (2020). https://doi.org/10.1145/3372297.3417291

[195] Y Zhang. 2022. Machine Learning Detection and Analysis on Obfuscated Android Malware. (2022). https://search.proquest.com/openview/da1a351bc9e02a0623272d56c12142c8/1?pq-origsite=gscholar&cbl=2026366&diss=y

[196] Y Zhao. 2023. Improving Android App Quality Using Big Code Analysis-based Techniques. (2023). https://bridges.monash.edu/articles/thesis/Improving_Android_App_Quality_Using_Big_Code_Analysis-based_Techniques/24064140

[197] Z Zhao, Z Liu, H Chen, F Zhang, and ... 2024. Effective DDoS mitigation via ML-driven in-network traffic shaping. *IEEE Transactions on …* (2024). https://ieeexplore.ieee.org/abstract/document/10380450/

[198] Z Zhu. 2019. Automatic Feature Engineering for Discovering and Explaining Malicious Behaviors. (2019). https://search.proquest.com/openview/a90f2bfc65917ace706109e73a840fee/1?pq-origsite=gscholar&cbl=18750&diss=y

[199] F Zola, JL Bruse, and M Galar. 2023. Temporal analysis of distribution shifts in malware classification for digital forensics. *2023 IEEE European Symposium …* (2023). https://ieeexplore.ieee.org/abstract/document/10190715/