1

Cybersecurity in the Era of Artificial Intelligence

The rapid and successful advances of artificial intelligence (AI) and machine learning (ML) offer security researchers and practitioners new approaches and platforms to explore and investigate challenging issues emerging in many safety-critical systems. Those AI/ML-enabled solutions have boosted the efficiency and effectiveness of multiple important security applications. For example, recent advances in AI and ML have been widely applied in intrusion detection system (IDS) (Xu et al., 2017, 2019a,b, 2020), malware detection system (Bradley and Xu, 2021; Bradley, 2022; Ahmed and Xu, 2022), and penetration testing (Chaudhary et al., 2020).

However, the rise of AI and ML is often considered as a "double-edged sword." While AI and ML can be adopted to identify threats more accurately and prevent cyberattacks more efficiently, cybersecurity professionals must respond to the increasingly sophisticated motivations from adversaries. Modern intelligent networking systems have been maliciously manipulated, evaded, and misled, causing significant security incidents in financial systems, cyber-physical systems, and many other critical domains. Threat actors and adversarial attackers have been applying techniques to carry out adversarial attacks targeting various AI/ML-enabled networking systems (Burr and Xu, 2021; Burr, 2022). For instance, an adversary can inject well-designed audio signals to confuse the voice recognition systems in smart speakers to deliver random noises, or compromising the self-driving vehicles by creating visual alterations of the stop sign, leaving the ML model erroneously identify a stop sign as a speed limit sign with *70* miles per hour (mph) (Yuan et al., 2019). Those adversarial attacks could lead to unauthorized disclosure of sensitive information, affect the safety and wellness of users, and thwart Internet freedom. Therefore, cybersecurity professionals must evolve rapidly as technology advances and new cyber threats emerge.

Cybersecurity in Intelligent Networking Systems, First Edition.
Shengjie Xu, Yi Qian, and Rose Qingyang Hu.

1.1 Artificial Intelligence for Cybersecurity

The concepts of AI and ML are firstly introduced, followed by the data-driven workflow for cybersecurity tasks.

1.1.1 Artificial Intelligence

The phrase *AI* is popularly discussed worldwide. Nowadays, AI generally refers to the simulation of human intelligent behavior by computational models to make decisions, and it is a rapidly evolving field of study, research, and application that is being used to improve economic development, modern human lifestyle, and national security. Along with recent technological advances, AI is used for innovation in various critical domains, such as robotics, manufacturing, business, finance, and many others.

AI applications are primarily enabled by *ML*, which is considered as the pillar of AI's success. Many organizations treat ML as the main approach to implement AI applications. It is an exciting field involving multiple subjects, including statistics, computer science, business management, linguistics, and more. Traditionally speaking, ML refers to the process of learning and understanding from historical data, mining and extracting the valuable information by recognizing the pattern and relationship, making decisions, and forecasting outcomes, trends, and behaviors. It involves a vast set of statistical models and tools, including generalized linear models, tree-based methods, neural networks, support vector machines, and nearest neighbors. Nowadays, ML is boosted by Big Data, massive computing power, and advanced learning models. In a technical article (Copeland, 2018), the author uses a Venn diagram to describe AI, ML, deep learning (DL), and their relationship. In Figure 1.1, the broad concept of AI including ML and DL is displayed. Currently, DL is leading the field of AI and ML, and it has made a significant number of progresses in a variety of ML domains, such as image classification, speech recognition, and object recognition.

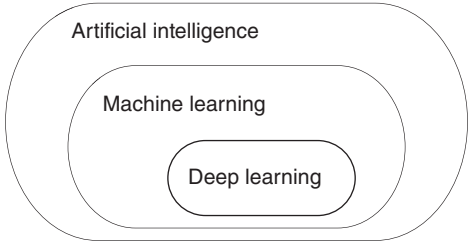

Figure 1.1 Artificial intelligence, machine learning, and deep learning.

Table 1.1 Example of a house price dataset.

(x_0)	(x_1)	(x_2)	(x_3)	…	(y)
Index	**Number of bedrooms**	**Square footage (sqft)**	**Number of bathrooms**	**…**	**Price ($)**
1	2	1600	2	…	≈ 250 000
2	4	2200	5	…	≈ 550 000
3	3	1800	3	…	≈ 400 000
…	…	…	…	…	…
100	4	2100	4	…	≈ 450 000

1.1.2 Machine Learning

ML offers computers to learn by mining massive datasets. Here, four broad categories of ML are described. They are supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.

1.1.2.1 Supervised Learning

Most of the ML problems fall into supervised or unsupervised. For instance, there is a house pricing dataset (Table 1.1), in which each row (observation) represents a house and each column (feature) represents an attribute (e.g. number of bedrooms). For each observation, an associated target value is shown. Here, the objective is to build a model that captures the relationship between the target value y (price) and the attributes $(x_0, \ldots, x_3)$ so that accurate predictions for future observations can be achieved.

Supervised learning addresses this type of problem by training the model with features and labeled data (y). A supervised learning model takes a set of known input data (features) and known output data (response/target) and trains a model to make reasonable predictions for the response to new data. Regression and classification are the main categories for supervised learning problems. In regression problems, there are many classical models available for training, including linear regression, ordinal regression, and neural network regression. In classification problems, there are also many classical models available for training, including logistic regression, tree-based methods, support vector machine, random forest, and boosting methods.

1.1.2.2 Unsupervised Learning

Unsupervised learning trains the model with unlabeled data. Its goal is to unveil the patterns in the data. Unsupervised learning serves as a good

approach to simplify the data by reducing the dimensionality, finding similar groups, and perceiving intrinsic structures. Clustering and dimensionality reduction are the main categories for unsupervised learning problems. In clustering problems, there are many classical models available for training, including *K*-means, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and hierarchical clustering. In dimensionality reduction problems, there are also many classical models available for training, including principal component analysis (PCA) and linear discriminant analysis (LDA).

1.1.2.3 Semi-supervised Learning

Semi-supervised learning deals with partially labeled data, which typically consist of a small amount of labeled and a large amount of unlabeled data. It falls between supervised learning, where completely labeled data are needed, and unsupervised learning, where no labeled data are needed. The trained model from semi-supervised learning can be highly accurate. Semi-supervised learning is also widely applied in the field of cybersecurity, especially in anomaly detection.

1.1.2.4 Reinforcement Learning

Reinforcement learning is a unique ML paradigm. This model learns a series of actions by maximizing a Reward Function f. The function f can be maximized by penalizing "bad action" and/or rewarding "good action." In the reinforcement learning setting, an agent takes actions in an environment that is treated as a reward and a representation of the state that is fed back to the agent. There are many popular examples that are enabled by reinforcement learning, such as self-driving vehicles (Gyawali et al., 2020) and AlphaGo (Silver et al., 2017).

1.1.3 Data-Driven Workflow for Cybersecurity

In the field of cybersecurity, data-driven methods are playing a crucial role in cybersecurity tasks, such as threat intelligence, risk analysis, vulnerability testing, and defense against adversarial behaviors. Figure 1.2 presents the general data-driven workflow to solve cybersecurity problems. The first step starts from formulating a concrete security problem and justifying the need to apply data-driven methods. For example, security practitioners can start defining a problem about intrusion detection, analyze the possible outputs given the inputs, and then argue whether data-driven methods are appropriate to automate the task. In the second step, data collection and preprocessing are conducted. Data acquisition is essential, and it is important to assure

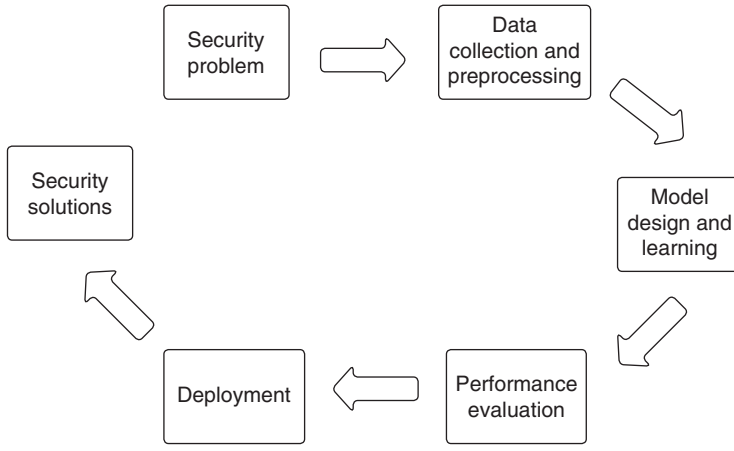

Figure 1.2 Data-driven workflow for cybersecurity.

that not only sufficient data are collected but also labeling and data sampling are correct and unbiased. In the third step, ML and other statistical models are designed and trained. It is important to assure that the trained model can be generalized well to future data and even unseen data. In the fourth step, performance evaluation is conducted to assess the quality of the trained model. Model assessment should be carried out by using suitable baseline data and appropriate metrics. In the fifth step, model deployment is performed. It is crucial to note that the deployment should perform well outside of a lab environment, and it should also work well under different settings in various threat models. Lastly, a mature and robust security solution for learning-based IDS is released.

1.2 Key Areas and Challenges

The research communities have been actively exploring both the offense and defense sides of data-driven cybersecurity. As more achievements are accomplished, several research challenges that emerge rapidly are pending to be solved. Here, three aspects are described. They are anomaly detection, trustworthy AI, and privacy preservation.

1.2.1 Anomaly Detection

In the context of cybersecurity, anomalies refer to those abnormal behaviors that harm the information systems. To defend against them, anomaly

detection focuses on identifying anomalous behaviors during a period of operations. This can be extended to a few use cases in security, such as intrusion detection, malware detection, phishing detection, spam detection, and defense against zero-day attack.

The goal of intrusion detection is to examine traffic data and classify normal activities and attack behaviors (Xu et al., 2019). More topics about cyber intrusions and intrusion detection will be covered in detail in Chapters 2, 4, 5, and 6.

Malware has been near the forefront of modern cybersecurity issues (Ahmed and Xu, 2022). The detection and prevention of malware has become a major challenge. In many cases, antivirus tools such as Windows Defender utilize ML to scan files and detect malicious patterns. With the recent trend of ransomware cases around the world, it has become more important than ever to have effective anti-malware to keep users and organizations safe. Modern malware can take many forms. It can be embedded in document macros, run as shellcode, and much more. Malware is also versatile in the taskings it can complete. It can implant a command-and-control server (C2) beacon, install a keylogger, or ransom a computer by encrypting all of the files. In addition to completing malicious tasks, malware has a secondary objective of avoiding detection by disguising itself as a valid process on a computer and obfuscate the detector.

Phishing is a tactic that is used by threat actors to achieve their goals, such as obtaining credentials of employees or delivering malware (Khonji et al., 2013). With the prevalence of these types of attacks, it is important to detect and stop the attacks at any point in their lifecycle. Being able to detect that a website is likely a phishing site could be a useful tool in mitigating the success of the attacks.

In recent years, the poisoning attack has become a new form of attack to compromise networking systems and the learning models they integrate. Data poisoning attack is the intentional act of polluting data that the algorithms need to train (Huang et al., 2021). It can impact organizations as well as individuals in a negative manner. Some real-life examples of this type of attack include an attacker changing what an email spam filter might mark as spam. This would allow an attacker to send any kind of email they want without being flagged. Similarly, there are firewall tools that use ML to monitor network traffic and look for malware entering the network. An attacker could use a similar method to evade detection.

1.2.2 Trustworthy Artificial Intelligence

Most of the AI/ML models face the issue of being trustworthy because they are vulnerable to various kinds of attacks (e.g. adversarial examples)

(Li et al., 2021a,b). This is because they are not yet explainable owing to the black-box nature of many AI/ML models, especially DL models (Zou et al., 2021), and their uncertainty has not been quantified (Li et al., 2021c). This highlights the importance of additional research activities in the trustworthiness of AI/ML models. Moreover, more studies are needed to systematically understand the trustworthiness of AI/ML models, as well as to advance the state-of-the-art trustworthiness, robustness, and interpretability of AI/ML.

Currently, research challenges of trustworthy AI and adversarial ML include, but not limited to, the following aspects: quantifying trustworthiness of AI/ML methods against sophisticated attacks, such as adversarial examples, poisoning attacks, or evasive attacks; determining the conditions to whether trust AI/ML models or not; explaining and interpreting recommendations made by AI/ML models; detecting and mitigating adversarial examples against AI/ML models; and enhancing trustworthiness of AI/ML models by incorporating countermeasures. Trustworthy AI and adversarial ML will be covered in detail in Chapter 8.

1.2.3 Privacy Preservation

The protection of sensitive data is critical. In the field of health study, the data privacy problem is growing because of the challenges in data privacy regulations, privacy leakage by attackers, and the pervasive data mining operations. In one study (Na et al., 2018), the authors discussed that by utilizing large national physical activity datasets, the children and adults were reidentified by ML when 20 minute data with several pieces of demographic information were used. Unfortunately, more incidents are reported worldwide regarding privacy leakage. With the enforcement of General Data Protection Regulation (GDPR) (Voigt and Von dem Bussche, 2017), companies and organizations will need to comply with specific terms and conditions to protect the data privacy of European Union citizens.

The increasing attention on security and privacy has motivated the rapid design and implementation of multiple privacy-preserving methods. For instance, federated learning (FL) was designed to train ML models across multiple end nodes without sharing the local data to a centralized server. One article shared that Google has implemented a mobile application that offers privacy-preserving word prediction-based FL (Hard et al., 2018).

There are a few modern approaches for privacy preservation, including anonymization, differential privacy (DP), homomorphic encryption (HE), and secure multi-party computation (SMPC). These approaches will be covered in detail in Chapter 7.

1.3 Toolbox to Build Secure and Intelligent Systems

A few scientific computing tools are commonly utilized as the toolbox to build intelligent systems. Here, a few Python libraries are described. They are for ML and DL, privacy-preserving ML, and adversarial ML.

1.3.1 Machine Learning and Deep Learning

Five Python libraries are widely used for ML and scientific computing. They are NumPy,[1] SciPy,[2] scikit-learn,[3] PyTorch,[4] and TensorFlow.[5]

1.3.1.1 NumPy

NumPy is the fundamental library for scientific computing in Python. It is a Python library that provides a multi-dimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and much more (Harris et al., 2020).

1.3.1.2 SciPy

SciPy is a Python library for mathematics, science, and engineering. It includes modules for statistics, optimization, integration, linear algebra, Fourier transforms, signal and image processing, ordinary differential equations (ODE) solvers, and more (Virtanen et al., 2020). SciPy is built to work with NumPy arrays and provides many user-friendly and efficient numerical routines, such as routines for numerical integration and optimization.

1.3.1.3 Scikit-learn

Scikit-learn is a Python library for ML built on top of SciPy (Pedregosa et al., 2011). It offers multiple modules for ML tasks, including classification, regression, clustering, dimensionality reduction, model selection, and preprocessing.

1.3.1.4 PyTorch

PyTorch is an optimized tensor library for DL. It provides deep neural networks built on an automatic differentiation system (autograd), as well as tensor computation with strong graphics processing unit (GPU) acceleration.

It can be used either as a replacement for NumPy to use the power of GPUs or a DL research platform that provides maximum flexibility and speed (Paszke et al., 2019).

1.3.1.5 TensorFlow

TensorFlow is an open-source platform for ML and DL. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML-powered applications (Abadi et al., 2016).

1.3.2 Privacy-Preserving Machine Learning

Three Python libraries are widely used for privacy-preserving ML. They are Syft,[6] TensorFlow Federated,[7] and TensorFlow Privacy.[8]

1.3.2.1 Syft

Syft is a Python library that preserves the privacy of data from model training, using FL, DP, and SMPC and HE within the main DL frameworks such as PyTorch and TensorFlow (Ziller et al., 2021).

1.3.2.2 TensorFlow Federated

TensorFlow Federated is an open-source framework for ML and other computations on decentralized data. It has been developed to facilitate open research and experimentation with FL (TFF, 2018).

1.3.2.3 TensorFlow Privacy

TensorFlow Privacy is a Python library that includes implementations of TensorFlow optimizers for training ML models with DP (TFP, 2022). The library contains a set of tutorials on building a language model with DP, learning a DL model with DP, and learning a differentially private logistic regression model on data.

1.3.3 Adversarial Machine Learning

Six Python libraries are widely used for adversarial ML. They are SecML,[9] SecML Malware,[10] Foolbox,[11] CleverHans,[12] Counterfit,[13] and MintNV.[14]

1.3.3.1 SecML and SecML Malware

SecML is a Python library for the security evaluation of ML models. It contains a set of ML models, built-in attack algorithms, pre-trained models,

multi-processing, and more. It offers tutorials in evasion attack, poisoning attack, and interpretability examples in ML (Melis et al., 2019). SecML Malware is a Python library for creating adversarial attacks against Windows Malware detectors. Built on top of SecML, it includes most attacks proposed in key articles (Demetrio and Biggio, 2021).

1.3.3.2 Foolbox

Foolbox is a Python library to create adversarial examples that fool ML models such as neural networks, as well as quantify and compare the robustness of ML models (Rauber et al., 2020). It is built because the most comparable robustness measure is the minimum perturbation needed to craft an adversarial example (Rauber et al., 2017).

1.3.3.3 CleverHans

CleverHans is a Python library to benchmark the vulnerability of ML systems to adversarial examples (Papernot et al., 2016).

1.3.3.4 Counterfit

Counterfit is a command-line tool and generic automation layer for assessing the security of ML systems (Counterfit, 2022).

1.3.3.5 MintNV

MintNV AI/ML educational exercise is a vulnerable environment for security professionals to practice attacking ML applications. This environment will locally host a realistic website and server for the end user to compromise (MintNV, 2022). Counterfit can be enabled within MintNV.

1.4 Data Repositories for Cybersecurity Research

Data are crucial for cybersecurity in the era of AI. A researcher created a site[15] that offers sources to multiple security-related data. Here, three datasets for cybersecurity research are described.

1.4.1 NSL-KDD

The NSL-KDD data are for intrusion detection problem (Revathi and Malathi, 2013). It was updated from data at the Third International Knowledge Discovery and Data Mining Tools Competition (KDD Cup 1999 data), which was to build a network intrusion detector, a predictive model capable of distinguishing between "bad" connections, called intrusions or

attacks, and "good" normal connections. The KDD data were created in an environment to acquire nine weeks of raw TCP dump data for a local area network (LAN) simulating a typical U.S. Air Force LAN (Stolfo et al., 2000). A connection is a sequence of TCP packets in which the data flow to and from a source IP address to a target IP address under some well-defined protocol. Each connection is labeled as either normal or as an attack with exactly one specific attack type. Attacks fall into four main categories: DoS: denial-of-service, e.g. syn flood; R2L: unauthorized access from a remote machine, e.g. guessing password; U2R: unauthorized access to local superuser (root) privileges, e.g. various "buffer overflow" attacks; and probing: surveillance and other probing, e.g. port scanning (Stolfo et al., 2000).

1.4.2 UNSW-NB15

The raw network packets of the UNSW-NB15 dataset were created in the Cyber Range Lab of University of New South Wales (UNSW) Canberra for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors (Moustafa and Slay, 2015). The tcpdump tool was utilized to capture 100 GB of the raw traffic (e.g. pcap files). This dataset has nine types of attacks, including Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms (Moustafa and Slay, 2016).

1.4.3 EMBER

EMBER dataset (Elastic Malware Benchmark for Empowering Researchers) is a collection of features from PE files that serve as a benchmark dataset. It focuses specifically on PE files, containing features and labels from 1.1 million samples. The data are packaged in JavaScript Object Notation (JSON). The labels are spread evenly as well, with an equal number of malicious and benign samples. Unlabeled samples are also included in the dataset. Raw features are extracted to JSON format and included in the publicly available dataset. Vectorized features can be produced from these raw features and saved in binary format from which they can be converted to CSV, dataframe, or any other format (Anderson and Roth, 2018).

1.5 Summary

In this chapter, the broader impact of AI and ML for cybersecurity is presented. The recent key research areas and challenges are discussed, and these areas will be illustrated in the following chapters. The toolbox for

building secure and robust intelligent networking systems is introduced, and it can be designed, developed, and implemented by utilizing several datasets for cybersecurity research.

Notes

1 NumPy. https://numpy.org/
2 SciPy. https://scipy.org/
3 scikit-learn. https://scikit-learn.org/stable/
4 PyTorch. https://pytorch.org/
5 TensorFlow. https://www.tensorflow.org/
6 Syft. https://github.com/OpenMined/PySyft
7 TensorFlow Federated. https://www.tensorflow.org/federated
8 TensorFlow Privacy. https://github.com/tensorflow/privacy
9 SecML. https://secml.readthedocs.io/en/stable/index.html
10 SecML Malware. https://github.com/pralab/secml_malware
11 Foolbox. https://github.com/bethgelab/foolbox
12 CleverHans. https://github.com/cleverhans-lab/cleverhans
13 Counterfit. https://github.com/Azure/counterfit
14 MintNV. https://catalog.ngc.nvidia.com/orgs/nvidia/teams/product-security/containers/mintnv
15 Samples of Security-Related Data. https://www.secrepo.com

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. {TensorFlow}: A system for {Large-Scale} machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

Tarek Ahmed and Shengjie Xu. Shellcoding: Hunting for Kernel32 base address. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–2. IEEE, 2022.

Hyrum S Anderson and Phil Roth. EMBER: An open dataset for training static PE malware machine learning models. *arXiv preprint arXiv:1804.04637*, 2018.

Matt Bradley. A metric for machine learning vulnerability to adversarial examples. *Dakota State University - Masters Theses & Doctoral Dissertations*, 2022.

Matthew Bradley and Shengjie Xu. A metric for machine learning vulnerability to adversarial examples. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–2. IEEE, 2021.

Justin Burr. Improving adversarial attacks against MalConv. *Dakota State University - Masters Theses & Doctoral Dissertations*, 2022.

Justin Burr and Shengjie Xu. Improving adversarial attacks against executable raw byte classifiers. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–2. IEEE, 2021.

Sujita Chaudhary, Austin O'Brien, and Shengjie Xu. Automated post-breach penetration testing through reinforcement learning. In *2020 IEEE Conference on Communications and Network Security (CNS)*, pages 1–2. IEEE, 2020.

Michael Copeland. What's the difference between artificial intelligence, machine learning and deep learning? 2018. URL https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/.

Counterfit. Counterfit, 2022. URL https://github.com/Azure/counterfit.

Luca Demetrio and Battista Biggio. Secml-malware: Pentesting windows malware classifiers with adversarial EXEmples in Python. *arXiv preprint arXiv:2104.12848*, 2021.

Sohan Gyawali, Shengjie Xu, Yi Qian, and Rose Qingyang Hu. Challenges and solutions for cellular based V2X communications. *IEEE Communication Surveys and Tutorials*, 23(1):222–255, 2020.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

Hai Huang, Jiaming Mu, Neil Zhenqiang Gong, Qi Li, Bin Liu, and Mingwei Xu. Data poisoning attacks to deep learning based recommender systems. *arXiv preprint arXiv:2101.02644*, 2021.

Mahmoud Khonji, Youssef Iraqi, and Andrew Jones. Phishing detection: A literature survey. *IEEE Communication Surveys and Tutorials*, 15(4):2091–2121, 2013.

Deqiang Li, Qianmu Li, Yanfang Ye, and Shouhuai Xu. Arms race in adversarial malware detection: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–35, 2021a.

Deqiang Li, Qianmu Li, Yanfang Ye, and Shouhuai Xu. A framework for enhancing deep neural networks against adversarial malware. *IEEE Transactions on Network Science and Engineering*, 8(1):736–750, 2021b.

Deqiang Li, Tian Qiu, Shuo Chen, Qianmu Li, and Shouhuai Xu. Can we leverage predictive uncertainty to detect dataset shift and adversarial examples in android malware detection? In *Annual Computer Security Applications Conference*, pages 596–608, 2021c.

Marco Melis, Ambra Demontis, Maura Pintor, Angelo Sotgiu, and Battista Biggio. secml: A python library for secure and explainable machine learning. *arXiv preprint arXiv:1912.10013*, 2019.

MintNV. MintNV, 6 2022. URL https://catalog.ngc.nvidia.com/orgs/nvidia/teams/product-security/containers/mintnv.

Nour Moustafa and Jill Slay. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *Military Communications and Information Systems Conference (MilCIS), 2015*, pages 1–6. IEEE, 2015.

Nour Moustafa and Jill Slay. The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective*, 25(1–3):18–31, 2016.

Liangyuan Na, Cong Yang, Chi-Cheng Lo, Fangyuan Zhao, Yoshimi Fukuoka, and Anil Aswani. Feasibility of reidentifying individuals in large national physical activity data sets from which protected health information has been removed with use of machine learning. *JAMA Network Open*, 1(8):e186040, 2018.

Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, et al. Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems 32*, 2019.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017.

Jonas Rauber, Roland Zimmermann, Matthias Bethge, and Wieland Brendel. Foolbox native: Fast adversarial attacks to benchmark the robustness of machine learning models in pytorch, tensorflow, and jax. *Journal of Open Source Software*, 5(53):2607, 2020.

S Revathi and A Malathi. A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. *International Journal of Engineering Research & Technology (IJERT)*, 2(12):1848–1853, 2013.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017.

J Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, and Philip K Chan. Cost-based modeling and evaluation for data mining with application to fraud and intrusion detection. *Results from the JAM Project by Salvatore*, pages 1–15, 2000.

TFF. TensorFlow Federated, 12 2018. URL https://github.com/tensorflow/federated.

TFP. TensorFlow Privacy, 2 2022. URL https://github.com/tensorflow/privacy.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272, 2020.

Paul Voigt and Axel Von dem Bussche. The EU general data protection regulation (GDPR). *A Practical Guide*, 1st Ed., Springer International Publishing, Cham, 10(3152676):10–5555, 2017.

Shengjie Xu, Yi Qian, and Rose Qingyang Hu. A data-driven preprocessing scheme on anomaly detection in big data applications. In *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 814–819. IEEE, 2017.

Shengjie Xu, Yi Qian, and Rose Qingyang Hu. Data-driven network intelligence for anomaly detection. *IEEE Network*, 33(3):88–95, 2019a.

Shengjie Xu, Yi Qian, and Rose Qingyang Hu. A semi-supervised learning approach for network anomaly detection in fog computing. In *2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019b.

Shengjie Xu, Yi Qian, and Rose Qingyang Hu. Data-driven edge intelligence for robust network anomaly detection. *IEEE Transactions on Network Science and Engineering*, 7(3):1481–1492, 2019c.

Shengjie Xu, Yi Qian, and Rose Qingyang Hu. Edge intelligence assisted gateway defense in cyber security. *IEEE Network*, 34(4):14–19, 2020.

Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2805–2824, 2019.

Alexander Ziller, Andrew Trask, Antonio Lopardo, Benjamin Szymkow, Bobby Wagner, Emma Bluemke, Jean-Mickael Nounahon, Jonathan Passerat-Palmbach, Kritika Prakash, Nick Rose, et al. PySyft: A library for easy federated learning. In *Federated Learning Systems* (eds. Muhammad Habib ur Rehman, Mohamed Medhat Gaber), pages 111–139. Springer, 2021.

Deqing Zou, Yawei Zhu, Shouhuai Xu, Zhen Li, Hai Jin, and Hengkai Ye. Interpreting deep learning-based vulnerability detector predictions based on heuristic searching. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 30(2):1–31, 2021.