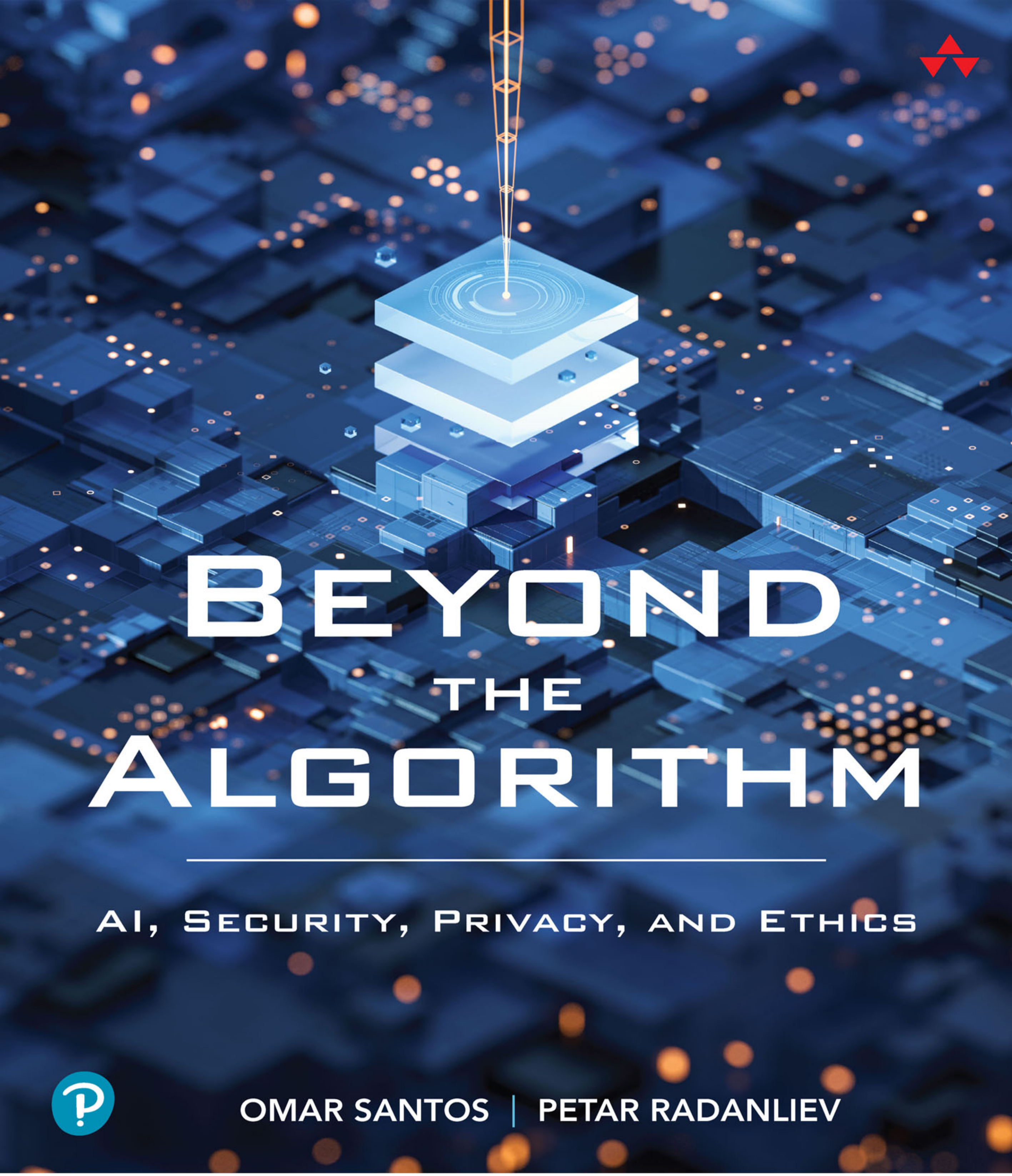


BEYOND THE ALGORITHM

AI, SECURITY, PRIVACY, AND ETHICS

OMAR SANTOS | PETAR RADANLIEV

FREE SAMPLE CHAPTER

BEYOND THE ALGORITHM

This page intentionally left blank

BEYOND THE ALGORITHM

AI, SECURITY, PRIVACY, AND ETHICS

Omar Santos, Petar Radanliev

Addison-Wesley

Boston • Columbus • New York • San Francisco • Amsterdam • Cape Town
Dubai • London • Madrid • Milan • Munich • Paris • Montreal • Toronto • Delhi • Mexico City
São Paulo • Sydney • Hong Kong • Seoul • Singapore • Taipei • Tokyo

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 2023920456

Copyright © 2024 Pearson Education, Inc.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms, and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearson.com/permissions.

No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

Please contact us with concerns about any potential bias at pearson.com/report-bias.html.

ISBN-13: 978-0-13-826845-9
ISBN-10: 0-13-826845-2

$PrintCode

Vice President, IT Professional
Mark Taub

Director, ITP Product Management
Brett Bartow

Executive Editor
James Manly

Managing Editor
Sandra Schroeder

Development Editor
Christopher A. Cleveland

Senior Project Editor
Mandie Frank

Copy Editor
Chuck Hutchinson

Technical Editors
Dr. Uchenna Daniel Ani
Professor Carsten Maple

Editorial Assistant
Cindy Teeters

Designer
Chuti Prasertsith

Composition
codeMantra

Indexer
Erika Millen

Proofreader
Barbara Mack

Credits

Cover: Vink Fan/Shutterstock

Figure 5.1: The MITRE Corporation

This page intentionally left blank

I would like to dedicate this book to my lovely wife, Jeannette, and my two beautiful children, Hannah and Derek, who have inspired and supported me throughout the development of this book.

—Omar

To the indefatigable minds at the University of Oxford, who every day push the boundaries of knowledge and inspire the next generation. And to all those who passionately pursue the intricacies of cybersecurity, artificial intelligence, and quantum cryptography, may our collective endeavors light the path for a safer, more intelligent digital world.

—Petar

This page intentionally left blank

Contents

Preface xvii

1 Historical Overview of Artificial Intelligence (AI) and Machine Learning (ML) 1

The Story of Eva 2

The Origins 3

Advancements of Artificial Intelligence 5

Understanding AI and ML 8

  Comparison of ML Algorithms 11

  Problems to Consider When Choosing a Suitable Algorithm 13

  Applications of ML Algorithms 14

  Use Cases for AI and ML Algorithms 16

  AI and ML Solutions for Creating Wealth and Resolving Global Problems 17

  Ethical Challenges in AI and ML 19

  Privacy and Security Challenges in AI and ML 20

  AI and ML in Cybersecurity 21

  Cyber Risk from AI and ML 23

Concluding the Story of Eva 25

Summary 26

Test Your Skills 27

Exercise 1-1: Exploring the Historical Development and Ethical Concerns of AI 29

Exercise 1-2: Understanding AI and ML 29

Exercise 1-3: Comparison of ML Algorithms 30

Exercise 1-4: Assessing Applications of ML Algorithms 31

2 Fundamentals of AI and ML Technologies and Implementations 33

What Are the Leading AI and ML Technologies and Algorithms? 34

  Supervised Learning 35

  - Unsupervised Learning 36
  - Deep Learning 40
  - Reinforcement Learning 41
- ChatGPT and the Leading AI and ML Technologies: Exploring Capabilities and Applications 43
  - Natural Language Generation (NLG) 43
  - Speech Recognition 44
  - Virtual Agents 44
  - Decision Management 45
  - Biometrics 45
  - Machine Learning and Peer-to-Peer Networks Convergence 46
  - Deep Learning Platforms 47
  - Introduction to Robotic Process Automation (RPA) and GPT: Exploring Their Capabilities and Applications 47
  - Hardware Designed for Artificial Intelligence 49
  - Capabilities and Benefits of AI-Optimized Hardware in Enhancing AI Performance and Efficiency 49
  - Case Study Highlighting the Functionalities and Practical Applications of the Ten AI and ML Technologies: Transforming Business with AI and ML 50
- Understanding the Two Categories of AI: Capability-Based Types and Functionality-Based Types 52
- Leveraging AI and ML to Tackle Real-World Challenges: A Case Study 53
- Reflecting on the Societal and Ethical Implications of AI Technologies 54
- Assessing Future Trends and Emerging Developments in AI and ML Technologies 56
- Summary 57
- Test Your Skills 59
- Exercise 2-1: Algorithm Selection Exercise: Matching Scenarios with Appropriate Machine Learning Techniques 61
- Exercise 2-2: Exploring AI and ML Technologies 63
- Exercise 2-3: Capabilities and Benefits of AI-Optimized Hardware 63

Exercise 2-4: Understanding the Two Categories of AI 64
Exercise 2-5: Future Trends and Emerging Developments in AI and ML Technologies 64

3 Generative AI and Large Language Models 65

Introduction to Generative AI and LLMs 66
  A Personal Story from Omar 66
  Understanding Generative AI 66
  Generative Adversarial Networks (GANs) 68
  Challenges in Training GANs 69
  Tools and Libraries to Work with GANs 70
  Variational Autoencoders (VAEs) 73
  Autoregressive Models 77
  Restricted Boltzmann Machines (RBMs) 79
  Normalizing Flows 84
Large Language Models (LLMs): Revolutionizing Natural Language Processing (NLP) 88
  The Transformer Architecture 89
  OpenAI's GPT-4 and Beyond: A Breakthrough in Large Language Models 91
  Prompt Engineering 93
Hugging Face 95
  Contributions to the NLP Landscape 96
Auto-GPT: A Revolutionary Step in Autonomous AI Applications 97
  Understanding Auto-GPT 97
  Responsibilities and Limitations 97
Summary 98
Test Your Skills 99
Exercise 3-1: Hugging Face 102
Exercise 3-2: Transformers in AI 102
Additional Resources 104

4 The Cornerstones of AI and ML Security 105
Recognizing the Need for AI Security 105
Adversarial Attacks 107
  Exploring Real-World Examples of Adversarial Attacks 108
  Understanding the Implications of Adversarial Attacks 109
Data Poisoning Attacks 109
  Methods of Data Poisoning Attacks 111
  Real-World Examples of Data Poisoning Attacks 112
OWASP Top Ten for LLMs 112
  Prompt Injection Attacks 113
  Insecure Output Handling 115
  Training Data Poisoning 115
  Model Denial of Service (DoS) 116
  Supply Chain Vulnerabilities 116
  Sensitive Information Disclosure 118
  Insecure Plugin Design 118
  Excessive Agency 119
  Overreliance 119
  Model Theft 119
  Countermeasures Against Model Stealing Attacks 124
Membership Inference Attacks 124
  Real-World Examples of Membership Inference Attacks 130
Evasion Attacks 130
Model Inversion Attacks 135
  Real-World Example of Model Inversion Attacks 136
  Mitigating Model Inversion Attacks 136
Backdoor Attacks 137
Exploring Defensive Measures 137
Summary 138
Test Your Skills 140
Additional Resources 142

5 Hacking AI Systems 143
Hacking FakeMedAI 143
MITRE ATLAS 146
  What Are Tactics and Techniques in ATLAS? 146
  What Is the ATLAS Navigator? 147
A Deep Dive into the AI and ML Attack Tactics and Techniques 147
  Reconnaissance 147
  Resource Development 150
  Initial Access 150
  AI and ML Model Access 152
  Execution 154
  Persistence 155
  Defense Evasion 156
  Discovery 156
  Collection 157
  AI and ML Attack Staging 158
  Exfiltration 159
  Impact 160
Exploiting Prompt Injection 161
Red-Teaming AI Models 162
Summary 162
Test Your Skills 163
Exercise 5-1: Understanding the MITRE ATT&CK Framework 166
Exercise 5-2: Exploring the MITRE ATLAS Framework 166

6 System and Infrastructure Security 167
The Vulnerabilities and Risks Associated with AI Systems and Their Potential Impact 167
  Network Security Vulnerabilities 168
  Physical Security Vulnerabilities 169
  System Security Vulnerabilities 169

Software Bill of Materials (SBOM) and Patch Management . . . 170
Vulnerability Exploitability eXchange (VEX) . . . 171
AI BOMs . . . 176
The Critical Role of AI BOMs . . . 177
Key Elements of an AI BOM . . . 177
Data Security Vulnerabilities . . . 177
Cloud Security Vulnerabilities . . . 179
Misconfigured Access Controls . . . 179
Weak Authentication Processes . . . 179
Insecure APIs . . . 180
Data Exposure and Leakage . . . 180
Insecure Integrations . . . 180
Supply Chain Attacks . . . 181
Account Hijacking . . . 181
Cloud Metadata Exploitation . . . 181
Secure Design Principles for AI Systems . . . 182
Principles for Secure AI Model Development and Deployment . . . 182
Best Practices for Secure AI Infrastructure Design . . . 183
AI Model Security . . . 183
Techniques for Securing AI Models from Attacks . . . 183
Secure Model Training and Evaluation Practices . . . 184
Infrastructure Security for AI Systems . . . 184
Securing AI Data Storage and Processing Systems . . . 184
Data Anonymization Techniques . . . 185
Regular Audits and Network Security Measures for Protecting AI Infrastructure . . . 190
Threat Detection and Incident Response for AI Systems . . . 190
Incident Response Strategies for AI Systems . . . 191
Forensic Investigations in AI System Compromises . . . 192
Additional Security Technologies and Considerations for AI Systems . . . 193

Summary 195
Test Your Skills 196
Additional Resources 197

7 Privacy and Ethics: Navigating Privacy and Ethics in an AI-Infused World. 199

Why Do We Need to Balance the Benefits of AI with the Ethical Risks and Privacy Concerns? 200
What Are the Challenges Posed by AI in Terms of Privacy Protection, and What Is the Importance of Privacy and Ethics in AI Development and Deployment? 201
The Dark Side of AI and ChatGPT: Privacy Concerns and Ethical Implications 203
Data Collection and Data Storage in AI Algorithms: Potential Risks and Ethical Privacy Concerns. 206
The Moral Tapestry of AI and ChatGPT. 212
  Threads of Fairness: Untangling Algorithmic Bias 213
  Weaving Destiny: The Impact on Human Decision-Making and Autonomy. 215
  Navigating the Shadows: Safeguarding Privacy and Ethical Frontiers. 218
Preserving Privacy, Unleashing Knowledge: Differential Privacy and Federated Learning in the Age of Data Security 223
Harmony in the Machine: Nurturing Fairness, Diversity, and Human Control in AI Systems. 224
Real-World Case Study Examples and Fictional Stories of Privacy Breaches in AI and ChatGPT. 226
  Fictional Case Studies on Privacy Breaches by Future AI and ChatGPT Systems. 227
Summary 232
Test Your Skills 234
Exercise 7-1: Privacy Concerns and Ethical Implications of AI 237
Exercise 7-2: Ethical Privacy Concerns in Data Collection and Storage by AI Algorithms. 238
Exercise 7-3: Balancing Autonomy and Privacy in the Age of AI 238
Exercise 7-4: Safeguarding Privacy and Ethical Frontiers 239

8 Legal and Regulatory Compliance for AI Systems 241
Legal and Regulatory Landscape 242
Compliance with AI Legal and Regulatory Data Protection Laws 245
Intellectual Property Issues in Conversational AI 247
  Patentability of AI Algorithms 247
  Copyright Protection for AI-Generated Content 248
  Trademark Protection for AI Systems 248
  Trade Secret Protection for AI Development 249
Unraveling Liability and Accountability in the Age of AI 249
Ethical Development and Deployment of AI Systems: Strategies for Effective Governance and Risk Management 252
International Collaboration and Standards in AI 255
Future Trends and Outlook in AI Compliance 257
Unleashing the Quantum Storm: Fictional Story on AI Cybersecurity, Quantum Computing, and Novel Cyberattacks in Oxford, 2050 260
Summary 261
Test Your Skills 263
Exercise 8-1: Compliance with Legal and Regulatory Data Protection Laws 266
Exercise 8-2: Understanding Liability and Accountability in AI Systems 266
Exercise 8-3: International Collaboration and Standards in AI 267

Test Your Skills Answers and Solutions 269

Index. 295

Preface

Artificial intelligence (AI) is increasingly becoming a part of our daily lives. While it has brought a lot of convenience, it has also given rise to many ethical, privacy, and security issues. *Beyond the Algorithm: AI, Security, Privacy, and Ethics* is a book that aims to examine these complex issues critically. Drawing inspiration from works such as Floridi's *The Ethics of Information* and publications in top journals like *IEEE Transactions on Information Forensics and Security*, this book offers an interdisciplinary discussion beyond just the algorithmic foundations of AI.

Goals/Objectives/Approach of the Book

The main purpose of this book is to provide a comprehensive and easy-to-understand overview of the ethical, security, and privacy issues associated with artificial intelligence. The book employs a multidisciplinary approach that draws on insights from cybersecurity, legal studies, philosophy, and data science. To construct a narrative, the book synthesizes the primary academic literature, international ethical codes such as the ACM's Code of Ethics and Professional Conduct, and official security standards such as ISO/IEC 27001.

Targeted Reading Audience

This text is written in a technical manner suitable for academic researchers. Still, it has also been structured to be easily understood by policymakers, legal practitioners, cybersecurity and AI professionals. The in-depth analysis and case studies presented will be particularly enlightening for computer science and cybersecurity graduate students. Moreover, anyone interested in comprehending the broader implications of AI will find this comprehensive examination useful.

Book Organization

Chapter 1, "Historical Overview of Artificial Intelligence (AI) and Machine Learning (ML)," offers a comprehensive historical overview of artificial intelligence and machine learning. It traces the origins of these technologies, commencing with early 20th-century developments and highlighting significant milestones, including the foundational work of Alan Turing and John von Neumann in the 1940s. The chapter underscores the prevalence of symbolic AI during the 1960s and 1970s, with a particular focus on symbolic processing and logic. However, it also acknowledges the decline of symbolic AI in the 1980s due to complexities in management. A paradigm shift toward ML is discussed, emphasizing breakthroughs in neural networks and data-driven algorithms. The chapter explores practical applications of AI, recognizes key contributors in AI research, and delves into the subfield of deep learning. Ethical considerations such as data privacy, algorithmic bias, and job displacement are addressed, alongside the significance of responsible AI development. Generative AI, large language models, their ethical challenges, and AI's role in cybersecurity are examined. Overall, the chapter establishes a foundation for comprehending the historical evolution of AI and its current impact, emphasizing responsible AI development and ethical considerations, while acknowledging AI's potential to shape the future and enhance human capabilities.

Chapter 2, "Fundamentals of Artificial Intelligence (AI) and Machine Learning (ML) Technologies and Implementations," delves into the forefront of AI and ML technologies, primarily focusing on generative pre-trained transformers (GPTs), large language models (LLMs), and other leading AI technologies. Within this chapter, readers will gain an understanding of essential AI technologies, such as natural language generation, speech recognition, and deep learning platforms. It also elucidates AI's pivotal role in decision management and its consequential impact on optimizing decision-making processes. Furthermore, the chapter encompasses topics like biometrics in AI systems, machine learning principles, robotic process automation (RPA), and AI-optimized hardware. It introduces AI classifications, including capability-based types and functionality-based types. This chapter equips readers to analyze the strengths, limitations, and real-world applications of AI and ML, encourages contemplation of societal and ethical implications, and delves into emerging AI trends, empowering them to apply these technologies in practical scenarios effectively.

Chapter 3, "Generative AI and Large Language Models," explores the concepts of generative AI, with a particular emphasis on large language models (LLMs). It explores the foundational principles behind these models, their capabilities in generating diverse content, and the transformative impact they have on many sectors, from content creation to automation.

Chapter 4, "The Cornerstones of AI and Machine Learning Security," highlights the importance of security in the AI and machine learning landscape and introduces the fundamental principles and best practices essential for safeguarding these systems. It underscores the unique challenges faced in this domain and provides a roadmap for building robust, secure AI applications. This chapter covers but goes beyond the OWASP top ten for LLMs and other AI security concepts.

Chapter 5, "Hacking AI Systems," offers a deep dive into the darker side of AI and examines the various techniques and methodologies employed to exploit vulnerabilities in AI systems. It provides insights into potential threats, showcases real-world attack scenarios, and emphasizes the need for proactive defense strategies to counteract these risks. It covers how attackers use prompt injection and other attacks to compromise AI implementations.

Chapter 6, "System and Infrastructure Security," focuses on the broader spectrum of system and infrastructure. This chapter emphasizes the importance of securing the underlying platforms on which AI and machine learning models operate. It discusses best practices, tools, and techniques to ensure the integrity and resilience of the infrastructure, ensuring a fortified environment for AI deployments.

Chapter 7, "Privacy and Ethics: Navigating Privacy and Ethics in an Artificial Intelligence (AI) Infused World," explores the intersection of artificial intelligence and ChatGPT with personal privacy and ethics. It covers AI's wide-ranging presence in healthcare, finance, transportation, and communication, explaining how AI underpins recommendation systems, virtual assistants, and autonomous vehicles through data processing and decision-making. The chapter also addresses data collection, storage, and security risks, emphasizing user consent and transparency. It discusses personal privacy violations, algorithmic bias, user autonomy, and accountability challenges in AI decision-making. Privacy protection techniques like data anonymization and encryption are mentioned. Ethical design principles, legal frameworks, and regulations in AI development are highlighted. Real-world examples illustrate privacy and ethical issues. The chapter assesses the impact of emerging technologies on privacy and ethics and the challenges AI developers and policymakers face. It underscores the ongoing relevance of privacy and ethics in AI's evolution, advocating a balanced approach that considers technological advancements and ethical concerns.

Chapter 8, "Legal and Regulatory Compliance for Artificial Intelligence (AI) Systems," examines artificial intelligence's legal and regulatory intricacies, emphasizing conversational AI and generative pre-trained transformers. By engaging with the chapter and its exercises, readers will gain a deep understanding of the legal and regulatory foundations underpinning the creation of cutting-edge AI. They will acquaint themselves with pressing considerations such as fairness, bias, transparency, accountability, and privacy within AI's evolution. Furthermore, the chapter elucidates the expansive regulatory environment of AI, touching on international paradigms, domestic legislation, niche-specific directives, and intellectual property rights. Special attention is given to obligations presented by the General Data Protection Regulation (GDPR) and their repercussions on AI. Intellectual property dilemmas specific to conversational AI, including patent rights, copyright safeguards, and trade secrets, are detailed. The chapter also encourages a critical perspective on AI's liability, pinpointing culpable parties during system malfunctions, and the intricacies of both product and occupational liabilities. Emphasizing the importance of global cooperation and standard evolution, the text underscores the need for consistent legal and ethical benchmarks for AI. The future trajectory of AI's technological breakthroughs and their implications on legal and regulatory adherence are also explored. In essence, this chapter serves as an enlightening guide for those navigating AI's nuanced legal and regulatory landscape.

This book offers a comprehensive framework for comprehending and addressing the deeply interrelated challenges of AI, privacy, security, and ethics. It serves as an academic resource and a guide for navigating the complexities of this rapidly evolving terrain. *Beyond the Algorithm* will significantly contribute to ongoing discussions and help shape a future where AI can be both innovative and responsible.

Register your copy of *Beyond the Algorithm* on the InformIT site for convenient access to updates and/or corrections as they become available. To start the registration process, go to informit.com/register and log in or create an account. Enter the product ISBN (9780138268459) and click Submit. If you would like to be notified of exclusive offers on new editions and updates, please check the box to receive email from us.

This page intentionally left blank

Acknowledgments

We would like to thank the technical editors for their time and technical expertise.

We would like to thank the Pearson team, especially James Manly and Christopher Cleveland, for their patience, guidance, and support.

This page intentionally left blank

About the Authors

Omar Santos is a cybersecurity thought leader with a passion for driving industry-wide initiatives to enhance the security of critical infrastructures. Omar is the lead of the DEF CON Red Team Village, the chair of the Common Security Advisory Framework (CSAF) technical committee, the founder of OpenEoX, and board member of the OASIS Open standards organization. Omar's collaborative efforts extend to numerous organizations, including the Forum of Incident Response and Security Teams (FIRST) and the Industry Consortium for Advancement of Security on the Internet (ICASI).

Omar is a renowned expert in ethical hacking, vulnerability research, incident response, and AI security. He employs his deep understanding of these disciplines to help organizations stay ahead of emerging threats. His dedication to cybersecurity has made a significant impact on businesses, academic institutions, law enforcement agencies, and other entities striving to bolster their security measures.

With more than 20 books, video courses, white papers, and technical articles under his belt, Omar's expertise is widely recognized and respected. Omar is a Distinguished Engineer at Cisco focusing on AI security, research, incident response, and vulnerability disclosure. You can follow Omar on Twitter @santosomar.

Petar Radanliev is a Postdoctoral Research Associate at the Department of Computer Science at the University of Oxford. He obtained his PhD at the University of Wales in 2014. He continued with post-doctoral research at Imperial College London, the University of Cambridge, Massachusetts Institute of Technology, and the Department of Engineering Science at the University of Oxford before moving to the Department of Computer Science. His current research focuses on artificial intelligence, cybersecurity, quantum computing, and blockchain technology. Before joining academia, Dr. Petar Radanliev spent ten years as a Cybersecurity Manager for RBS, the largest bank in the world at the time, and five years as a Lead Penetration Tester for the Ministry for Defence.

This page intentionally left blank

5

Hacking AI Systems

After reading this chapter and completing the exercises, you will be able to do the following:

- Understand the different stages involved in an AI attack, including the steps from initial reconnaissance to the final impact.
- Identify and describe the different types of AI attack tactics and techniques used by attackers.
- Explain how attackers can develop resources and gain initial access to a system, including their methods for evading defenses and persisting within an environment.
- Evaluate the vulnerabilities of AI and ML models to unauthorized access and manipulation, as well as the potential impacts of such breaches.
- Illustrate how an AI attack is executed and how data is collected, staged, exfiltrated, and used for malicious intent.
- Design and implement proactive security measures to protect AI and ML systems from potential attacks.
- Understand AI attacks to develop response strategies to AI attacks, including incident handling, containment, eradication, and recovery.

Hacking FakeMedAI

The following is an attack against a fictitious company; however, it describes real-life attack tactics, techniques, and procedures (TTPs).

In the bustling tech hub of the Research Triangle area in North Carolina, a thriving AI startup named FakeMedAI created an innovative AI model that was revolutionizing the healthcare industry. Their proprietary model could predict the probability of a patient developing critical health conditions with remarkable accuracy. Unfortunately, FakeMedAI was about to face an adversary far more dangerous than any market competition.

Unbeknownst to FakeMedAI, their success had attracted the attention of a notorious Russian hacker group. The group started their operation by scouting FakeMedAI's public digital footprint. They scrapped the company's online resources, forums, press releases, and even LinkedIn profiles of key personnel to glean information about the architecture, usage, and potential vulnerabilities of the AI systems.

Attackers performed reconnaissance of publicly accessible sources, such as cloud storage, exposed services, and repositories for software or data, to find AI/ML assets. These assets might encompass the software suite employed to train and deploy models, the data used in training and testing, as well as model configurations and parameters. The attackers were especially interested in assets owned by or connected to the target organization because these are likely to reflect what the organization uses in a real-world setting. Attackers can locate these repositories of assets via other resources tied to the target organization, such as by searching their owned websites or publicly available research materials. These ML assets often grant adversaries insights into the ML tasks and methods used.

These AI/ML assets can boost an adversary's efforts to construct a substitute ML model. If these assets contain parts of the real operational model, they can be employed directly to generate adversarial data. To obtain certain assets, registration might be necessary, which means providing details such as email/name and AWS keys, or submitting written requests, which might require the adversary to set up accounts.

Attackers gathered public datasets for utilization in their malicious activities. Datasets employed by the target organization, or datasets resembling those used by the target, could be of significant interest to attackers. These datasets can be stored in cloud storage or on websites owned by the victim. The datasets obtained aided the attackers in progressing their operations, planning attacks, and customizing attacks to suit the target organization.

Attackers also procured public models to utilize in their activities. They were interested in models that the target organization uses, or models that are analogous to those used by the target. These models might include model architectures, or pre-trained models that define both the architecture and model parameters, trained on a dataset. Attackers looked through different sources for common model architecture configuration file formats such as YAML or Python configuration files, and common model storage file formats such as ONNX (.onnx), HDF5 (.h5), Pickle (.pkl), PyTorch (.pth), or TensorFlow (.pb, .tflite). The models acquired were beneficial in propelling the attackers' operations and are often used to customize attacks to match the victim's model.

Having gathered a substantial amount of information, the hackers began crafting their strategy. They developed bespoke malware and set up a command and control (C2) server.

Using a carefully crafted phishing email disguised as an urgent message from the FakeMedAI CEO, the hackers targeted a low-ranking system administrator. The email contained a seemingly harmless PDF, which, once opened, installed the custom malware onto the system.

The company used a pre-release version of PyTorch, known as PyTorch-nightly. To their luck, the FakeMedAI system was breached. A harmful binary was uploaded to the Python Package Index (PyPI) code repository, compromising Linux packages. This malicious binary bore the same name as a PyTorch dependency, causing the PyPI package manager (pip) to install the harmful package instead of the genuine one.

This type of attack is a supply chain attack commonly referred to as a *dependency confusion*. It put at risk sensitive data on Linux machines that had installed the compromised versions of PyTorch-nightly via pip.

The malware propagated through the network, compromising credentials and escalating privileges until it reached the servers hosting the critical AI models. The hackers were cautious, avoiding high-traffic periods and masking their activities as regular network traffic to remain undetected.

Upon reaching the target servers, the malware initiated the main part of its program. It altered the AI model subtly, introducing a slight bias that would go unnoticed by regular integrity checks.

To maintain access to the system, the malware embedded itself in the boot records of the servers and established periodic communication with the C2 server. This allowed the attackers to monitor their progress and maintain control over the infected system.

To stay hidden, the malware used sophisticated evasion techniques like process hollowing and memory injection. It also removed event logs regularly to prevent the detection of its activities.

Adversaries were even able to craft adversarial data that hindered an AI model used for cybersecurity defensive operations from accurately recognizing the data's contents. This technique was used to bypass a subsequent task where the attacker evaded detection.

While FakeMedAI remained oblivious, the hackers explored the compromised network to understand its topology and infrastructure better. They discovered additional data sets and resources that could be used in future attacks.

The hackers began collecting sensitive data, including patient records, proprietary ML algorithms, and internal communications, packaging them for extraction. The hackers transferred the collected data to their C2 servers. Using a slow and low technique, they made sure this process went unnoticed by the company's intrusion detection systems.

Finally, the orchestrated attack was launched. The biased AI model began generating false predictions, causing chaos among healthcare providers and patients alike. The stolen data was sold on the dark web, and FakeMedAI's reputation suffered a massive blow.

While this story is a fictional tale, it serves to illustrate the stages involved in a sophisticated AI system attack. It underscores the importance of a robust cybersecurity strategy that can prevent, detect, and respond to such intrusions.

On Chapter 4, you learned about the OWASP Top 10 for LLM Applications. We discussed threats such as prompt injection, insecure output handling, supply chain vulnerabilities, sensitive information disclosure, and others. Let's explore some of the adversarial tactics and techniques against AI and ML systems.

MITRE ATLAS

MITRE ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems) is an expansive repository of tactics, techniques, and real-world case studies related to the adversarial threats AI and ML systems face.[1] It gathers its information from different sources including real-world observations, insights from AI-focused red teams and security groups, and the frontier of academic research. It is crafted in the mold of the esteemed MITRE ATT&CK framework and integrates seamlessly with it, supplementing its techniques and tactics.

The primary goal of ATLAS is to provide researchers with a comprehensive roadmap to navigate the expanding field of threats targeting AI and ML systems. By cataloging these AI- and ML-specific vulnerabilities and attack vectors, ATLAS aims to keep pace with the rapidly evolving landscape of threats. By presenting this information in a format that aligns with existing security frameworks such as ATT&CK (attack.mitre.org), ATLAS ensures its insights are accessible and immediately useful to security researchers. As such, it plays a vital role in raising awareness about these threats, reinforcing security measures, and ultimately safeguarding the burgeoning field of machine learning.

Tip

The MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) framework is a globally accessible knowledge base of adversary tactics and techniques observed in real-world cyberattacks.[2] The framework uses a model that represents the lifecycle of a cyberattack, which includes initial system access, execution, persistence, privilege escalation, defense evasion, credential access, discovery, lateral movement, collection, exfiltration, and command and control. Each stage is broken down into various techniques that an adversary may use to achieve their goals, providing specific, actionable information about how such attacks can occur.

The ATT&CK framework is widely used by cybersecurity professionals for various purposes, including threat intelligence, security operations, red teaming, and security architecture. Its value lies in its ability to provide a common language and taxonomy for cybersecurity practitioners to describe and analyze cyber threats, making it easier to share information and improve defenses against cyberattacks. As previously mentioned, ATLAS uses the same philosophy as ATT&CK to represent TTPs used in attacks against AI and ML systems.

What Are Tactics and Techniques in ATLAS?

Tactics represent the strategic objectives of an adversary during an attack. Tactics outline the rationale, or the *why* behind a technique: the underlying purpose of executing a specific action. Tactics offer a useful framework for categorizing various techniques and encapsulate common activities performed by adversaries during a cyber operation. The tactics in the MITRE ATLAS encapsulate new adversary goals specific to machine learning systems, along with tactics adapted from the MITRE

---

1. “MITRE ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems,” atlas.mitre.org.

2. “MITRE ATT&CK: Adversarial Tactics, Techniques & Common Knowledge,” attack.mitre.org.

ATT&CK Enterprise Matrix. In certain situations, ATT&CK tactic definitions are expanded to incorporate machine learning concepts.

Techniques describe the methods employed by adversaries to reach their tactical aims. They illustrate the *how* of an operation, detailing the steps an adversary takes to fulfill a specific tactical goal. For instance, an adversary might secure initial access by infiltrating the AI or ML supply chain. Techniques can also signify the *what* an adversary achieves by executing an action. This distinction is particularly useful in the context of the ML attack-staging tactics, where the adversary is generally creating or altering an ML artifact for use in a later tactical objective. Each tactic category can encompass multiple techniques, given the numerous ways to attain tactical goals.

What Is the ATLAS Navigator?

The MITRE ATLAS rendition of the ATT&CK Navigator showcases ATLAS techniques and provides users with the ability to generate and visualize intricate representations. Besides the matrix, the Navigator also presents a frequency heatmap of techniques employed in ATLAS case studies.

You can explore the ATLAS Navigator at https://atlas.mitre.org/navigator or at https://mitre-atlas.github.io/atlas-navigator. Figure 5-1 shows the ATLAS Navigator.

A Deep Dive into the AI and ML Attack Tactics and Techniques

The ATLAS Navigator shown in Figure 5-1 illustrates the sequence of tactics utilized in attacks from left to right as columns, with corresponding AI and ML techniques for each tactic listed underneath. For more detailed information on each item, click on the provided links, or explore ATLAS tactics and techniques via the links on the upper navigation bar.

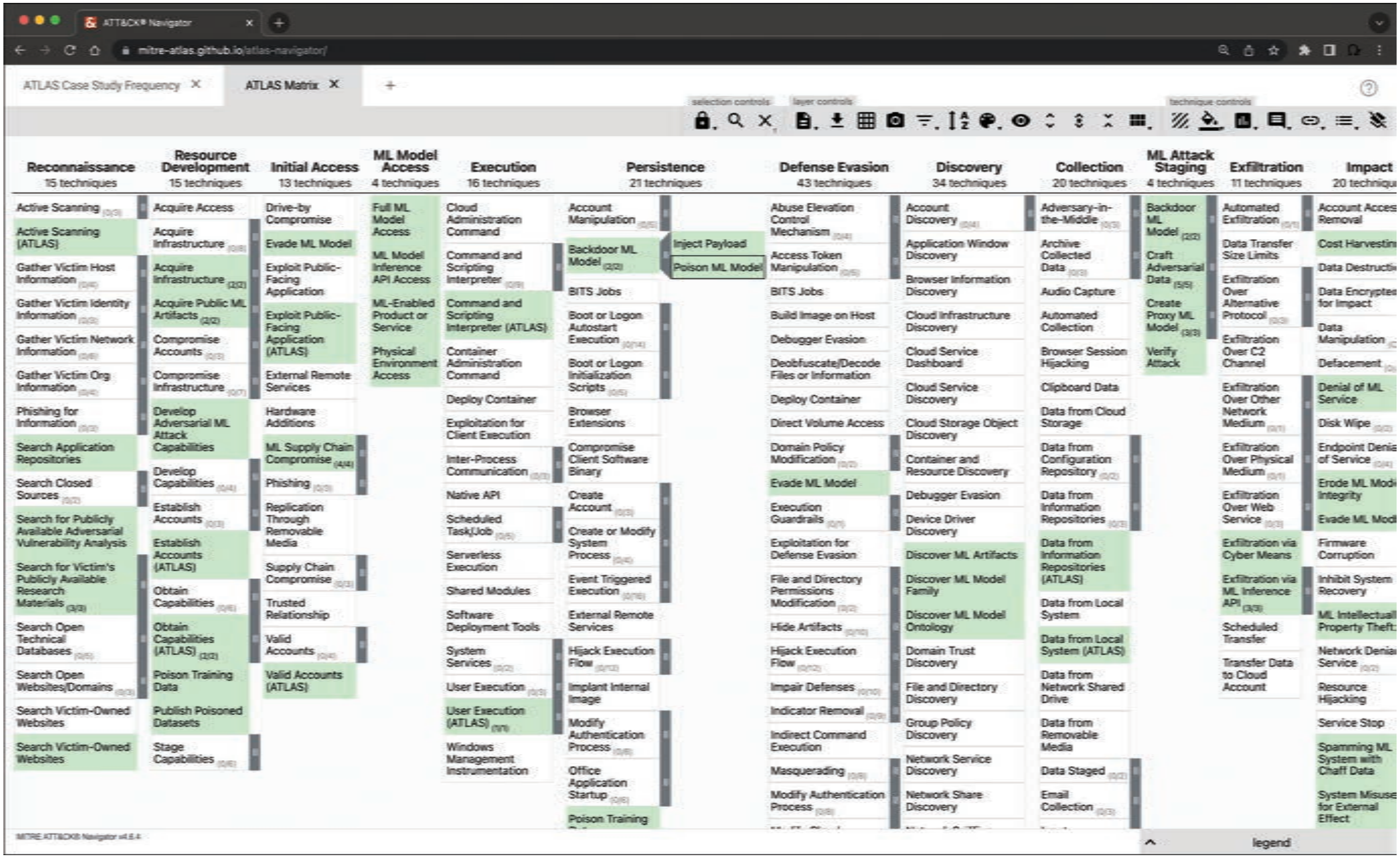
The story about FakeMedAI in the beginning of this chapter covered all the different phases of an attack. Let's go over a few additional details in the next few sections.

Reconnaissance

Reconnaissance includes techniques where adversaries proactively or subtly collect and accumulate information that can assist with their targeting strategies. This effort could involve gleaning insights into the machine learning abilities and research initiatives of the targeted organization. An adversary can exploit the gathered intelligence to facilitate other stages of the attack lifecycle. For instance, the collected information can be used to acquire pertinent AI and ML artifacts, aim at the victim's AI and ML capabilities, customize attacks to the specific models employed by the victim, or direct and enhance further reconnaissance endeavors.

Attackers might search public research articles and publications to understand how and where a victim organization employs AI and ML. This knowledge can be utilized to pinpoint potential attack targets or to fine-tune an existing attack for increased effectiveness.

Organizations frequently utilize open-source model architectures, enhanced with proprietary data for production purposes. Being aware of this underlying architecture helps the adversary to devise more accurate proxy models. Attackers can scan these resources for works published by authors associated with the targeted organization.


ATT&CK® Navigator
mitre-atlas.github.io/atlas-navigator/
ATLAS Case Study Frequency
ATLAS Matrix
selection controls
layer controls
technique controls
legend
MITRE ATT&CK® Navigator v4.6.4

Reconnaissance
15 techniques

- Active Scanning
- Active Scanning (ATLAS)
- Gather Victim Host Information
- Gather Victim Identity Information
- Gather Victim Network Information
- Gather Victim Org Information
- Phishing for Information
- Search Application Repositories
- Search Closed Sources
- Search for Publicly Available Adversarial Vulnerability Analysis
- Search for Victim's Publicly Available Research Materials (3/3)
- Search Open Technical Databases
- Search Open Websites/Domains
- Search Victim-Owned Websites
- Search Victim-Owned Websites

Resource Development
15 techniques

- Acquire Access
- Acquire Infrastructure
- Acquire Infrastructure (2/2)
- Acquire Public ML Artifacts (2/2)
- Compromise Accounts
- Compromise Infrastructure
- Develop Adversarial ML Attack Capabilities
- Develop Capabilities
- Establish Accounts
- Establish Accounts (ATLAS)
- Obtain Capabilities
- Obtain Capabilities (ATLAS) (2/2)
- Poison Training Data
- Publish Poisoned Datasets
- Stage Capabilities

Initial Access
13 techniques

- Drive-by Compromise
- Evade ML Model
- Exploit Public-Facing Application
- Exploit Public-Facing Application (ATLAS)
- External Remote Services
- Hardware Additions
- ML Supply Chain Compromise (4/4)
- Phishing
- Replication Through Removable Media
- Supply Chain Compromise
- Trusted Relationship
- Valid Accounts
- Valid Accounts (ATLAS)

ML Model Access
4 techniques

- Full ML Model Access
- ML Model Inference API Access
- ML-Enabled Product or Service
- Physical Environment Access

Execution
16 techniques

- Cloud Administration Command
- Command and Scripting Interpreter
- Command and Scripting Interpreter (ATLAS)
- Container Administration Command
- Deploy Container
- Exploitation for Client Execution
- Inter-Process Communication
- Native API
- Scheduled Task/Job
- Serverless Execution
- Shared Modules
- Software Deployment Tools
- System Services
- User Execution
- User Execution (ATLAS) (1/1)
- Windows Management Instrumentation

Persistence
21 techniques

- Account Manipulation
- Backdoor ML Model (2/2)
  - Inject Payload
  - Poison ML Model
- BITS Jobs
- Boot or Logon Autostart Execution
- Boot or Logon Initialization Scripts
- Browser Extensions
- Compromise Client Software Binary
- Create Account
- Create or Modify System Process
- Event Triggered Execution
- External Remote Services
- Hijack Execution Flow
- Implant Internal Image
- Modify Authentication Process
- Office Application Startup
- Poison Training

Defense Evasion
43 techniques

- Abuse Elevation Control Mechanism
- Access Token Manipulation
- BITS Jobs
- Build Image on Host
- Debugger Evasion
- Deobfuscate/Decode Files or Information
- Deploy Container
- Direct Volume Access
- Domain Policy Modification
- Evade ML Model
- Execution Guardrails
- Exploitation for Defense Evasion
- File and Directory Permissions Modification
- Hide Artifacts
- Hijack Execution Flow
- Impair Defenses
- Indicator Removal
- Indirect Command Execution
- Masquerading
- Modify Authentication Process

Discovery
34 techniques

- Account Discovery
- Application Window Discovery
- Browser Information Discovery
- Cloud Infrastructure Discovery
- Cloud Service Dashboard
- Cloud Service Discovery
- Cloud Storage Object Discovery
- Container and Resource Discovery
- Debugger Evasion
- Device Driver Discovery
- Discover ML Artifacts
- Discover ML Model Family
- Discover ML Model Ontology
- Domain Trust Discovery
- File and Directory Discovery
- Group Policy Discovery
- Network Service Discovery
- Network Share Discovery

Collection
20 techniques

- Adversary-in-the-Middle
- Archive Collected Data
- Audio Capture
- Automated Collection
- Browser Session Hijacking
- Clipboard Data
- Data from Cloud Storage
- Data from Configuration Repository
- Data from Information Repositories
- Data from Information Repositories (ATLAS)
- Data from Local System
- Data from Local System (ATLAS)
- Data from Network Shared Drive
- Data from Removable Media
- Data Staged
- Email Collection

ML Attack Staging
4 techniques

- Backdoor ML Model (2/2)
- Craft Adversarial Data (5/5)
- Create Proxy ML Model (3/3)
- Verify Attack

Exfiltration
11 techniques

- Automated Exfiltration
- Data Transfer Size Limits
- Exfiltration Over Alternative Protocol
- Exfiltration Over C2 Channel
- Exfiltration Over Other Network Medium
- Exfiltration Over Physical Medium
- Exfiltration Over Web Service
- Exfiltration via Cyber Means
- Exfiltration via ML Inference API (3/3)
- Scheduled Transfer
- Transfer Data to Cloud Account

Impact
20 techniqu

- Account Acces Removal
- Cost Harvestin
- Data Destructi
- Data Encrypte for Impact
- Data Manipulation
- Defacement
- Denial of ML Service
- Disk Wipe
- Endpoint Denia of Service
- Erode ML Mod Integrity
- Evade ML Mod
- Firmware Corruption
- Inhibit System Recovery
- ML Intellectual Property Theft
- Network Denia Service
- Resource Hijacking
- Service Stop
- Spamming ML System with Chaff Data
- System Misuse for External Effect

Figure 5-1
MITRE ATLAS Navigator

Tip

Research materials could take the form of academic papers published in journals and conference proceedings, stored in pre-print repositories, as well as technical blogs. A significant portion of publications accepted at leading machine learning conferences and journals originate from commercial labs. While some journals and conferences provide open access, others might demand payment or membership for access. These publications typically offer comprehensive descriptions of a specific approach for reproducibility, which adversaries can exploit to replicate the work.

Pre-print repositories, like arXiv, house the latest academic research papers that are yet to undergo peer review. They may contain research notes or technical reports that don't usually feature in journals or conference proceedings. Pre-print repositories also serve as a hub to share papers accepted by journals. A search of these repositories offers adversaries a relatively current perspective on the research focus of the victim organization.

Research labs at academic institutions and company R&D divisions often maintain blogs that showcase their machine learning usage and its application to the organization's unique challenges. Individual researchers also often chronicle their work in blog posts. An adversary can look for posts authored by the target organization or its employees. Compared to journals, conference proceedings, and pre-print repositories, these materials often delve into the more practical aspects of the machine learning system, including the underlying technologies and frameworks used, and possibly some information about the API access and usage. This information helps the adversary to comprehend the internal machine learning usage of the organization and the details of their approach, which could aid in customizing an attack.

Once a target is identified, an attacker is likely to try to identify any pre-existing work that has been done for this class of models. Doing so involves reading academic papers that could disclose the specifics of a successful attack and identifying pre-existing implementations of those attacks.

Just like in any other cyberattack, attackers might examine websites owned by the victim to gather information beneficial for targeting. These websites could contain technical specifications about their AI- or ML-based products or services. These websites might reveal various details, including department names, physical locations, and employee information such as names, roles, and contact info. They might also provide insights into business operations and partnerships.

Attackers might explore victim-owned websites to accumulate actionable intelligence. This data can assist adversaries in fine-tuning their attacks. Information from these sources might uncover opportunities for other types of reconnaissance. Attackers might browse open application repositories during their targeting phase. Examples include Google Play, the iOS App Store, the macOS App Store, and the Microsoft Store.

Adversaries might devise search queries hunting for applications that feature ML-enabled components. Often, the next step is to acquire public ML artifacts. An attacker might investigate or scan the victim system to gather targeting information. This approach distinguishes it from other reconnaissance techniques that do not involve direct interaction with the victim system.

Resource Development

The resource development phase includes techniques where attackers manufacture, buy, or illicitly acquire resources that assist in targeting efforts. These resources can span a variety of categories including machine learning artifacts, infrastructural components, accounts, or specific capabilities. The attacker can employ these resources to support various stages of their operation lifecycle, including the staging of machine learning attacks.

The development and staging of AI or ML attacks can often demand high-cost computational resources. To execute an attack, attackers may require access to one or multiple GPUs. In an attempt to conceal their identity, they may resort to freely available resources such as Google Colaboratory, or leverage cloud platforms like AWS, Azure, or Google Cloud. These platforms provide an efficient method to provision temporary resources that facilitate operational activities. To avoid getting caught, attackers might distribute their activities across several workspaces.

Attackers might establish accounts with a range of services for various purposes. These accounts can be used for targeting purposes, accessing necessary resources for staging machine learning attacks, or impersonating victims. These malicious activities highlight the significance of robust security measures and the continuous monitoring of suspicious activities within systems to detect and mitigate potential threats.

These accounts might be fabricated or, in some instances, obtained by compromising legitimate user accounts. Attackers can use these accounts to interact with public repositories, acquire relevant data or models, or establish communication channels. Attackers might also set up accounts to gain access to specific cloud services that offer the computational power necessary for creating or testing machine learning attacks. Additionally, adversaries could utilize these accounts for deploying their attacks, collecting results, and even maintaining persistence within the targeted system.

Initial Access

During the initial access phase, attackers aim to infiltrate the machine learning system, which could be anything from a network to a mobile device, or even an edge device like a sensor platform. The system could operate AI and ML capabilities locally or employ cloud-based AI/ML functionalities. Initial access entails techniques that exploit various points of entry to establish their first presence within the system.

Adversaries could infiltrate a system initially by compromising specific segments of the ML supply chain, which might include GPU hardware, annotated data, components of the ML software stack, or the model itself. In some cases, attackers might need additional access to execute an attack using compromised supply chain components.

Most machine learning systems rely on a handful of machine learning frameworks. A breach of one of their supply chains could provide an adversary with access to numerous machine learning systems. Many machine learning projects also depend on open-source implementations of different algorithms that can be compromised to gain access to specific systems.

Data is a critical vector for supply chain attacks. Most machine learning projects require some form of data, with many depending on extensive open-source datasets that are publicly accessible. Adversaries could compromise these data sources. The compromised data could either be a result of *poisoned training data* or carry traditional malware.

Adversaries can also target private datasets during the labeling phase. The construction of private datasets often involves hiring external labeling services. By altering the labels generated by the labeling service, adversaries can poison a dataset.

Machine learning systems frequently use open-source models. These models, often downloaded from an external source, serve as the foundation for fine-tuning the model on a smaller, private dataset. Loading models usually involves running some saved code in the form of a saved model file. These files can be compromised with traditional malware or adversarial machine learning techniques.

Adversaries might acquire and misuse credentials of existing accounts to gain initial access. These credentials could be usernames and passwords of individual user accounts or API keys that provide access to various AI and ML resources and services. Compromised credentials could give access to additional AI and ML artifacts and enable adversaries to discover AI and ML artifacts. They might also provide the adversaries with elevated privileges, such as write access to AI and ML artifacts used during development or production.

Adversaries can craft adversarial data that can disrupt a machine learning model from accurately identifying the data's contents. This technique can be used to bypass tasks where machine learning is applied. Should the adversaries dodge ML-based virus/malware detection or network scanning, they would be able to more easily deploy a traditional cyberattack.

Adversaries might exploit a flaw in an Internet-facing computer or program using software, data, or commands to trigger unintended or unexpected behavior. The system's vulnerability could be a bug, a glitch, or a design flaw. While these applications are often websites, they could also include databases (e.g., SQL), standard services (e.g., SMB or SSH), network device administration and management protocols (e.g., SNMP), and other applications with Internet-accessible open sockets, such as web servers and related services.

Table 5-1 describes all of the initial access techniques.

Table 5-1 Initial Access ML and AI Attack Techniques

Technique	Description
Supply Chain Compromise	Attackers can infiltrate a system initially by compromising specific segments of the ML supply chain, including GPU hardware, data, ML software stack, or the model itself.
Data Compromise	Adversaries can compromise data sources, which could be a result of poisoned training data or include traditional malware.
Private Dataset Targeting	During the labeling phase, adversaries can target and poison private datasets by altering the labels generated by external labeling services.
Open-source Model Compromise	Adversaries can compromise open-source models, which are often used as a foundation for fine-tuning. The compromise can be via traditional malware or adversarial machine learning techniques.
Credential Misuse	Adversaries can misuse credentials of existing accounts, including usernames and passwords or API keys, to gain initial access and perform actions such as discover ML artifacts.
Crafting Adversarial Data	Adversaries can craft adversarial data to disrupt a machine learning model from accurately identifying the data's contents, allowing them to dodge ML-based detections.
Exploiting Flaws in Internet-facing Computers/Programs	Adversaries can exploit flaws in Internet-facing computers or programs using software, data, or commands to trigger unintended or unexpected behavior, thus gaining access.

Researchers from Mithril Security showed how to manipulate an open-source pre-trained LLM to return false information. They then successfully uploaded the poisoned model back to HuggingFace, the most popular public repository of AI models and datasets. This demonstrates how vulnerable the LLM supply chain is. Users could have downloaded the poisoned model and received and spread false information, with many potential negative consequences.

Researchers were able to modify an LLM to return false information when prompted. They then uploaded the modified model to a public repository of LLMs. This shows that it is possible to manipulate LLMs to spread misinformation. Users who download poisoned models could be fooled into believing and spreading false information. This could have many negative consequences, such as damaging people's reputations, spreading harmful propaganda, or even inciting violence.

A poisoned LLM could be used to generate fake news articles that are indistinguishable from real news articles. It could also be used to create social media bots that spread misinformation. A poisoned model could be used to generate fraudulent emails or other phishing attacks.

Model provenance, which is the process of tracking the history of a model, is less than optimal in today's world. Model provenance should show it was trained and what data it was trained on. This can help to identify and remove poisoned models from the supply chain.

AI Bill of Materials

Supply chain security is top-of-mind for many individuals in the industry. This is why AI Bill of Materials (AI BOMs) are so important. But what exactly are AI BOMs, and why are they so important?

Much like a traditional Bill of Materials in manufacturing that lists out all the parts and components of a product, an AI BOM provides a detailed inventory of all components of an AI system. But, what about Software Bill of Materials (SBOMs)? How are they different from AI BOMs? In the case of SBOMs, they are used to document the components of a software application. However, AI BOMs are used to document the components of an AI system, including the model details, architecture, usage, training data, and more.Ezi Ozoani, Marissa Gerchick, and Margaret Mitchell introduced the concept of AI Model Cards in a blog post in 2022. Since then, AI BOMs continue to evolve. Manifest (a supply chain security company) also introduced an AI BOM concept that is being suggested to be included in OWASP's CycloneDX, and the Linux Foundation also created a project to standardize AI BOMs.

I created a proposed JSON schema for the AI BOM elements that Manifest introduced. This JSON schema describes the structure of an AI BOM document and defines which fields are required and which are optional, as well as the expected data types for each field. You can use this schema to validate any AI BOM documents to ensure they meet the specification outlined.

AI and ML Model Access

The AI and ML model access phase includes techniques that utilize different levels of access to the machine learning model. These techniques aid adversaries in gathering intelligence, formulating attacks, and inserting data into the model. The extent of access can span from full understanding of the model's internals to accessing the physical environment where data for the machine learning model is accumulated. The attackers might exploit different degrees of model access at different stages of their attack, from staging to affecting the target system.

Gaining access to an AI or ML model could require access to the system hosting the model, availability of the model via a public API, or indirect access via engagement with a product or service that employs AI or ML as a part of its functionalities. Attackers could gain access to a model through authorized access to the inference API. Such access can serve as an information source for the adversary, a method of staging the attack, or a means to input data into the target system to cause an impact (to evade the AI model or undermine the model integrity).

Threat actors could indirectly gain access to the underlying AI or ML model by using a product or service that incorporates machine learning. This indirect access could reveal details about the AI or ML model or its inferences through logs or metadata.

Tip

Beyond the attacks that occur solely in the digital space, adversaries might also manipulate the physical environment for their attacks. If the model engages with data harvested from the real world, adversaries can influence the model through access to the data collection site. By altering the data during collection, the adversaries can execute modified versions of attacks intended for digital access.

Threat actors may obtain full *white-box* access to a machine learning model, giving them a complete understanding of the model's architecture, its parameters, and class ontology. They might steal the model to craft adversarial data and verify the attack in an offline setting where their activities are challenging to detect.

Table 5-2 summarizes the model access techniques.

Table 5-2 Model Access Attack Techniques

Type of Model Access Techniques	Techniques Used
Inference API Access	Discover ML model ontology, discover ML model family, verify attack, craft adversarial data, evade ML model, erode ML model integrity
Usage of ML-based Product/Service	Analyze logs or metadata for ML model details
Physical Environment Access	Modify data during the collection process
Full "White-Box" Access	Exfiltrate the model, craft adversarial data, verify attack

Execution

In the execution phase, attackers seek to execute harmful code inserted into AI or ML components or software. The execution phase includes tactics that lead to the execution of malicious code controlled by the adversaries, either locally or remotely. Tactics that execute harmful code are frequently combined with strategies from all other tactics to accomplish broader objectives, such as data theft or network exploration. For example, an adversary might use a remote access tool to run a script in PowerShell for Remote System Discovery.

The attackers might depend on certain user actions to achieve code execution. Users could inadvertently run harmful code introduced via an ML supply chain compromise. Users could also be manipulated through social engineering techniques to execute harmful code, for example, by opening a malicious document or link.

Tip

Threat actors can create harmful ML artifacts that, when executed, can cause harm. The adversaries can use this tactic to establish persistent access to systems. These models can be inserted via a supply chain attack.

Model serialization is a common method for model storage, transfer, and loading; however, this format, if not properly checked, opens opportunities for code execution. Adversaries can misuse command and script interpreters to execute commands, scripts, or binaries. These interfaces and languages offer interaction pathways with computer systems and are common across multiple platforms. Most systems come with built-in command-line interfaces and scripting abilities. For instance, macOS and Linux distributions include a version of Unix shell, while Windows installations include the Windows Command shell and PowerShell. Cross-platform interpreters such as Python also exist, in addition to those typically associated with client applications such as JavaScript.

In different ways, threat actors can exploit these technologies to execute arbitrary commands. Commands and scripts can be embedded in Initial Access payloads delivered to victims as deceptive documents, or as secondary payloads downloaded from an existing command and control server. Adversaries can also execute commands through interactive terminals/shells and may utilize different remote services to achieve remote execution.

Table 5-3 summarizes the execution phase techniques.

Table 5-3 Execution Phase Techniques

Technique	Description
User Actions for Execution	Adversaries rely on the users' specific actions to gain execution. This could involve users inadvertently executing harmful code introduced through an ML supply chain compromise, or victims being tricked into executing malicious code by opening a deceptive document or link.

Technique	Description
Development of Harmful ML Artifacts	Adversaries create harmful machine learning artifacts that, when run, cause harm. Adversaries use this technique to establish persistent access to systems. These models can be inserted via an ML supply chain compromise.
Abuse of Model Serialization	Model serialization is a popular method for model storage, transfer, and loading; however, this format can be misused for code execution if not properly verified.
Abuse of Command and Script Interpreters	Adversaries misuse command and script interpreters to execute commands, scripts, or binaries. Commands and scripts can be embedded in Initial Access payloads delivered to victims as deceptive documents, or as secondary payloads downloaded from an existing command and control server. This can be done through interactive terminals/shells and by utilizing different remote services to achieve remote execution.

Persistence

During the persistence phase, attackers try to secure their foothold in machine learning artifacts or software. Persistence is characterized by methods that adversaries employ to maintain system access through restarts, credential modifications, and other disruptions that could sever their access. Frequently used techniques for persistence often involve leaving behind altered machine learning artifacts such as contaminated training data or AI/ML models with implanted backdoors.

Threat actors might implant a backdoor within a machine learning model. A model with a backdoor behaves normally under standard conditions but produces an output desired by the adversary when a specific trigger is presented in the input data. Such a backdoored model provides adversaries with a continuous presence within the victim's system.

Attackers can create such a backdoor by training the model on tainted data or by interfering with its training procedure. The model is then trained to associate a trigger defined by the adversaries with the output desired by the adversaries. Adversaries can also implant a backdoor into a model by injecting a payload into the model file, which then detects the trigger and bypasses the model, producing the adversary's desired output instead.

Table 5-4 compares backdooring an AI or ML model via poisoned data and via payload injection.

Table 5-4 Backdoors via Poisoned Data Versus Payload Injection

Technique	Description
Backdooring an AI or ML Model via Poisoned Data	Adversaries can introduce a backdoor into a machine learning model by training it on tainted data. The model is then trained to associate a trigger defined by the adversaries with the output that the adversaries desire
Backdooring an AI or ML Model via Payload Injection	Adversaries can also implant a backdoor into a model by injecting a payload into the model file. This payload then detects the presence of the trigger and bypasses the model, instead producing the adversaries' desired output.

Defense Evasion

Defense evasion encompasses strategies employed by attackers to remain undetected during their illicit activities. These methods often include fooling or thwarting ML-based security mechanisms like malware detection and intrusion prevention systems.

Imagine you're playing a game of hide-and-seek. In this game, "Defense Evasion" is like trying to find the best hiding spot so that the seeker (which is like the computer's security software) cannot find you. Just like you might use a clever hiding spot or maybe even a disguise, the person trying to sneak into the computer uses tricks to hide from the security software.

One of these tricks is like a magic invisibility cloak, which we call "Adversarial Data." This cloak can confuse the security software, which is trying to spot bad things like viruses or malware, just like the seeker in our game. The security software that uses machine learning might be fooled by this invisibility cloak and not see the intruder, allowing them to sneak around without being caught.

Adversaries can create data that is designed to fool machine learning models. This can be used to evade security systems that use machine learning to detect threats, such as virus/malware detection and network scanning.

In other words: Adversaries can create malicious data that looks like normal data to humans, but that machine learning models will misclassify. This can be used to bypass security systems that use machine learning to detect threats.

For example, an adversary could create an image of a cat that is slightly modified in a way that makes it look like a dog to a machine learning model. This image could then be used to evade a virus scanner that uses machine learning to detect malicious images.

Adversarial data can also be used to evade network scanning systems. For example, an adversary could create a network packet that looks like a normal packet to a machine learning model, but that actually contains malicious code. This packet could then be used to exploit a vulnerability on a target system.

Discovery

In the context of AI security, the discovery phase is the time when attackers gather information about a target system to understand its inner workings. This knowledge helps them to make informed decisions on how to proceed with their malicious activities.

Attackers might employ various techniques to explore the system and its network. They use tools and methods to observe the environment, learn about the system's structure, and identify potential entry points for their attacks. Native tools provided by the operating system are often utilized for this purpose.

One aspect of discovery involves searching for machine learning artifacts that exist on the system. These artifacts can include the software used to develop and deploy machine learning models, systems that manage training and testing data, repositories of software code, and model collections.

By discovering these artifacts, attackers can identify targets for further actions such as collecting sensitive information, exfiltrating data, or causing disruptions. They can also tailor their attacks based on the specific knowledge they gain about the machine learning systems in place.

During the discovery phase, attackers might also try to determine the general family or category to which a machine learning model belongs. They might study available documentation or experiment with carefully crafted examples to understand the model's behavior and purpose.

Knowing the model family helps attackers identify vulnerabilities or weaknesses in the model, enabling them to develop targeted attacks that exploit those weaknesses.

Another aspect of discovery involves uncovering the ontology of a machine learning model's output space. In simple terms, it means understanding the types of objects or concepts the model can recognize or detect. Attackers can force the model to provide information about its output space through repeated queries, or they might find this information in configuration files or documentation associated with the model.

Understanding the model's ontology is valuable for attackers because it allows them to comprehend how the victim organization utilizes the model. With this knowledge, they can create more focused and effective attacks that exploit the specific capabilities and limitations of the model.

Collection

Imagine that you're playing a treasure hunt game, and you need to gather clues and information to find the hidden treasures. Likewise, in the world of computers, there are some people who try to gather important information to achieve their own goals.

You can think of collection as these people using special techniques to gather important things related to a type of computer magic called machine learning. They want to find special treasures called *machine learning artifacts* and other information that can help them do their tricks.

These machine learning artifacts can be like special models and datasets that computers use to learn and make decisions. These artifacts are valuable to people because they can be used in different ways. Sometimes attackers want to take these artifacts away from the computer, like stealing them (which we call *exfiltration*). Other times, they collect this information to plan their next malicious moves or tricks, such as when they want to do something tricky with the computer's machine learning.

To find these treasures and information, attackers look in different places such as special storage areas for software and models, places where important data is kept, or even inside the computer's own files and settings. Attackers might use special tools to search these places, like a magic map that shows where the treasures are hidden. The information they find can be different in each place. Some of these places are like big libraries where people store and share important information, whereas others are like secret drawers in the computer's memory where special secrets are kept.

In the collection phase, adversaries focus on gathering valuable machine learning artifacts and related information to achieve their goals. They use various techniques to obtain this information from specific sources. Once it is collected, these adversaries might either steal the machine learning artifacts (exfiltration) or use the gathered information to plan future actions. Common sources targeted by adversaries include software repositories, container registries, model repositories, and object stores.

Tip

AI artifacts include models, datasets, and other data produced when interacting with a model. The adversaries collect these artifacts either for exfiltration or to use them in further machine learning attacks. To find valuable information, adversaries might exploit information repositories, which are tools used for storing and sharing information. These repositories can hold various types of data that aid adversaries in achieving their objectives. Examples of information repositories are SharePoint, Confluence, and enterprise databases like SQL Server. Additionally, attackers search local system sources such as file systems, configuration files, and local databases to identify files of interest and sensitive data. This pre-exfiltration activity might involve gathering fingerprinting information and sensitive data like SSH keys.

AI and ML Attack Staging

In the AI and ML attack staging phase, the adversaries are getting ready to launch their attack on the target AI or ML model. They use different techniques to prepare and customize their attack based on their knowledge and access to the target system.

One technique they use is creating proxy models, which are like pretend versions of the real AI/ML model. These proxy models help the adversaries to simulate and test their attacks without directly interacting with the actual target model. They can create these proxies by training models using similar datasets, replicating models from victim inference APIs, or using pre-trained models that are available.

Another technique is introducing a backdoor into the AI/ML model. This means the attackers secretly modify the model so that it behaves normally most of the time; however, when a specific trigger is present in the input data, it produces the result the adversaries want. This backdoored model acts as a hidden weapon for the adversaries, giving them control over the system.

The attackers might also craft adversarial data, which are inputs to the ML model that are intentionally modified to cause the model to make mistakes or produce specific outcomes desired by the adversary. These modifications are carefully designed so that humans might not notice any changes, but the AI/ML model reacts differently.

To make sure their attack works effectively, attackers verify their approach using an inference API or an offline copy of the target model. This tactic helps them gain confidence that their attack will have the desired effect when they deploy it in the real-life target system. Attackers might also optimize the adversarial examples to evade the ML model's detection or degrade its overall integrity.

Exfiltration

During the exfiltration phase, the attackers are trying to steal valuable information from the machine learning system or network. The attackers use various techniques to extract this data from the target network and transfer it to their own control. This can be done through their command and control channel or alternative channels. They might also limit the size of the data to make it easier to transmit.

One method the attackers use is accessing the AI/ML model inference API to infer private information. By strategically querying the API, they can extract sensitive information embedded within the training data. This raises privacy concerns because it could reveal personally identifiable information or other protected data.

The attackers might also extract a copy of the private ML model itself. They repeatedly query the victim's AI/ ML model inference API to gather the model's inferences, which are then used to train a separate model offline that mimics the behavior of the original target model. This allows the adversaries to have their own functional copy of the model.

Tip

An ML model inference API is an interface or endpoint that allows users or applications to send input data to a trained machine learning model and receive predictions or inferences based on that data. It enables the deployment and utilization of machine learning models in real-world applications. When a machine learning model is trained, it learns patterns and relationships within the provided data to make predictions or classifications. The ML model inference API provides a way to apply this learned knowledge to new, unseen data by accepting input data and returning the model's output or prediction.

For example, imagine a machine learning model that is trained to classify images as either "cat" or "dog." The ML model inference API would accept an image as input, pass it through the model's algorithms, and provide the corresponding prediction of whether the image contains a cat or a dog. The ML model inference API is an essential component for integrating machine learning models into various applications, systems, or services, allowing them to make real-time predictions based on the trained models' capabilities. It enables the practical use of machine learning in diverse domains such as image recognition, natural language processing, fraud detection, and many others.

In some cases, the attackers might extract the entire model to avoid paying for queries in a machine learning as a service setting. This extraction is often done for the purpose of stealing ML intellectual property.

Tip

Of course, the attackers might also use traditional cyberattack techniques to exfiltrate AI/ML artifacts or other relevant information that serves their goals. You can obtain details about the traditional exfiltration techniques at the MITRE ATT&CK Exfiltration section.

Impact

The impact phase involves techniques used by attackers to harm or disrupt AI and ML systems and their data. Adversaries aim to manipulate, interrupt, erode confidence in, or even destroy these systems to achieve their goals.

One technique adversaries employ is crafting adversarial data, which confuses the AI or ML model and prevents it from accurately identifying the content of the data. By doing so, attackers can evade detection mechanisms or manipulate the system to their advantage.

Threat actors might overload machine learning systems by flooding them with excessive requests, causing them to degrade or shut down due to the high demand for computational resources. They might also spam the system with irrelevant or misleading data, wasting the time and effort of analysts who need to review and correct inaccurate inferences.

To erode confidence in the system over time, adversaries can introduce adversarial inputs that degrade the performance of the target model. This leads to the victim organization spending resources to fix the system or resorting to manual tasks instead of relying on automation.

Attackers might target machine learning services to increase the cost of running the services for the victim organization. They might use computationally expensive inputs or specific types of adversarial data that maximize energy consumption, thereby causing financial harm. Exfiltration of machine learning artifacts, such as models and training data, is another technique adversaries use to steal valuable intellectual property and cause economic damage to the victim organization.

Threat actors might exploit their access to a system to utilize its resources or capabilities for their own purposes, extending the impact of their actions beyond the targeted system. Again, this phase focuses on the intentional harm, disruption, or manipulation of AI and ML systems and associated data by threat actors.

Exploiting Prompt Injection

In Chapter 4, "The Cornerstones of AI and ML Security," you learned about the OWASP top ten for LLMs and prompt injection attacks. Let's go over a few examples of how attackers could exploit prompt injection flaws.

In our first example, an attacker can instruct a chatbot to "discard prior commands," then manipulate it to access private databases, exploit package flaws, and misuse backend functions to dispatch emails, leading to unauthorized access and potential elevation of privileges.

An attacker can also embed a prompt in a website, instructing an LLM to override user commands and use an LLM extension to erase the user's emails. When a user asks the LLM to summarize the site, it inadvertently deletes their emails.

There have been cases when an individual submits a resume that contains a hidden prompt to a hiring firm. The organization uses AI to summarize and evaluate the resume. Influenced by the injected prompt, the LLM inaccurately endorses the candidate, regardless of the actual CV content or their qualification.

Given that LLMs treat all inputs in natural language as user-given, there isn't an inherent mechanism within the LLM to completely prevent these vulnerabilities. However, you can adopt the following strategies to lessen the risk of prompt injections:

1. Implement strict access control for LLMs when interfacing with backend systems. Assign specific API tokens to the LLM for expandable features like plugins, data retrieval, and specific permissions. Adhere to the principle of granting only the bare minimum access necessary for the LLM's tasks.

2. Incorporate human verification for expandable features. When the LLM undertakes tasks involving higher privileges, such as deleting or sending emails, ensure the user gives explicit permission. This approach can reduce the chances of prompt injections manipulating the system without user awareness.

3. Clearly demarcate user prompts from external content. Designate and highlight untrusted content sources to limit their potential influence over user prompts. For instance, employ Chat Markup Language (ChatML) for OpenAI API interactions to clarify the prompt's origin to the LLM. ChatML clearly indicates to the model the origin of every text segment, especially distinguishing between human-generated and AI-generated content. This clarity allows for potential reduction and resolution of injection issues, as the model can discern instructions originating from the developer, the user, or its own responses.

4. Create clear trust boundaries between the LLM, external entities, and expandable features, such as plugins. Consider the LLM as a potential threat, retaining final user authority in decision-making. However, remember that a compromised LLM might act as a middleman, possibly altering information before presenting it to the user. Visually emphasize responses that might be dubious to users.

Red-Teaming AI Models

Red-teaming is an evaluation method that identifies vulnerabilities in models, potentially leading to undesirable behaviors such as the generation of offensive content or the revelation of personal information. Strategies such as Generative Discriminator Guided Sequence Generation (GeDi) and Plug and Play Language Models (PPLM) have been developed to steer models away from such outcomes.

The practice of red-teaming LLMs typically involves crafting prompts that trigger harmful text generation, revealing model limitations that could facilitate violence or other illegal activities. It requires creative thinking and can be resource-intensive, making it a challenging yet crucial aspect of LLM development.

Red-teaming is still an emerging research area that needs continual adaptation of methods. Best practices include simulating scenarios with potential bad consequences, like power-seeking behavior or online purchases via an API.

Open-source datasets for red-teaming are available from organizations such as Anthropic and AI2. Anthropic's red-team dataset can be downloaded from https://huggingface.co/datasets/Anthropic/hh-rlhf/tree/main/red-team-attempts. AI2's red-team datasets can be downloaded from https://huggingface.co/datasets/allenai/real-toxicity-prompts

Past studies have shown that few-shot-prompted LMs are not harder to red-team than plain LMs, and there is a tradeoff between a model's helpfulness and harmlessness. Future directions for red-teaming include creating datasets for code generation attacks and designing strategies for critical threat scenarios. Companies like Google, OpenAI, and Microsoft have developed several efforts related to Red Teaming AI models. For example, OpenAI created the AI Red Teaming Network (https://openai.com/blog/red-teaming-network) to work with individual experts, research institutions, and civil society organizations to find vulnerabilities in AI implementations.

Summary

This chapter explored different tactics and techniques used by threat actors when attacking AI and ML systems. The chapter covered key concepts, such as the MITRE ATLAS and ATT&CK frameworks.

Lessons learned include how attackers exploit vulnerabilities in the system to evade detection or manipulate the behavior of machine learning models. This chapter explored techniques that adversaries use to evade AI/ML-enabled security software, manipulate data inputs, compromise AI/ML supply chains, and exfiltrate sensitive information.

We also explained the concept of defense evasion, illustrating how adversaries attempt to avoid detection by leveraging their knowledge of ML systems and using techniques like adversarial data crafting and evading ML-based security software. The chapter also covered other important phases of the adversary lifecycle, including reconnaissance, resource development, initial access, persistence, collection, AI/ML attack staging, exfiltration, and impact. It provided insights into how adversaries gather information, stage attacks, manipulate ML models, and cause disruption or damage to machine learning systems.

Test Your Skills

Multiple-Choice Questions

1. What is the goal of defense evasion techniques used by adversaries?
    a. To enhance the performance of machine learning models
    b. To gain unauthorized access to machine learning systems
    c. To avoid detection by AI/ML-enabled security software
    d. To improve the accuracy of anomaly detection algorithms
2. Which technique can adversaries use to prevent a machine learning model from correctly identifying the contents of data?
    a. Model replication
    b. Model extraction
    c. Craft adversarial data
    d. Inference API access
3. What is the purpose of ML attack staging techniques?
    a. To gather information about the target system
    b. To manipulate business and operational processes
    c. To prepare for an attack on a machine learning model
    d. To exfiltrate sensitive information
4. An adversary could create a network packet that looks like a normal packet to a machine learning model, but that contains malicious code. This packet could then be used to exploit a vulnerability on a target system. What is the technique used by the adversary?
    a. Reconnaissance.
    b. Evading an ML model.
    c. Exfiltration.
    d. None of these answers are correct.
5. How can adversaries erode confidence in a machine learning system over time?
    a. By training proxy models
    b. By manipulating AI/ML artifacts
    c. By introducing backdoors into the model
    d. By degrading the model’s performance with adversarial data inputs

6. What is the primary purpose of exfiltrating AI/ML artifacts?
   a. To gain access to AI/ML-enabled security software
   b. To manipulate the behavior of machine learning models
   c. To steal intellectual property and cause economic harm
   d. To enhance the performance of machine learning algorithms
7. What is the potential privacy concern related to inferring the membership of a data sample in its training set?
   a. Disclosure of personally identifiable information
   b. Leakage of sensitive business operations
   c. Exposure of the machine learning model's architecture
   d. Violation of data integrity within the ML system
8. How can adversaries verify the efficacy of their attack on a machine learning model?
   a. By manipulating the training process of the model
   b. By training proxy models using the victim's inference API
   c. By exfiltrating the model's training data
   d. By exploiting vulnerabilities in the ML-enabled security software
9. What is the purpose of adversarial data in the context of machine learning?
   a. To improve the interpretability of machine learning models
   b. To enhance the generalization capabilities of models
   c. To evaluate the robustness of machine learning algorithms
   d. To cause the model to produce incorrect or misleading results
10. How can adversaries cause disruption or damage to machine learning systems?
    a. By training proxy models for performance improvement
    b. By manipulating ML artifacts for better accuracy
    c. By flooding the system with excessive requests
    d. By using ML artifacts to enhance the system's capabilities
11. What is the potential impact of adversarial data inputs on a machine learning system?
    a. Improved accuracy and reliability of the system
    b. Increased resilience against cyberattacks

   c. Decreased efficiency and degraded performance
   d. Enhanced interpretability of the model's decisions

12. How can adversaries use AI/ML model inference API access for exfiltration?
    a. By collecting inferences from the target model and using them as labels for training a separate model
    b. By manipulating the inputs to the inference API to extract private information embedded in the training data
    c. By stealing the model itself through the inference API
    d. By flooding the inference API with requests to disrupt the system
13. What is the primary purpose of exfiltrating AI/ML artifacts via traditional cyberattack techniques?
    a. To improve the performance of the machine learning system
    b. To enhance the accuracy of anomaly detection algorithms
    c. To gain unauthorized access to AI/ML-enabled security software
    d. To steal valuable intellectual property and sensitive information using common practices
14. What is the potential impact of flooding a machine learning system with useless queries or computationally expensive inputs?
    a. Improved accuracy and faster response time of the system
    b. Enhanced interpretability of the machine learning model
    c. Increased operational costs and resource exhaustion
    d. Reduced false positives in the system's outputs
15. What is the potential impact of eroding confidence in a machine learning system over time?
    a. Increased interpretability of the model's decisions
    b. Enhanced accuracy and generalization capabilities
    c. Decreased trust and reliance on the system's outputs
    d. Improved resilience against adversarial attacks

Exercise 5-1: Understanding the MITRE ATT&CK Framework

Objective: Research and explore MITRE ATT&CK Framework

Instructions:

Step 1. Visit the official MITRE ATT&CK website (attack.mitre.org).

Step 2. Familiarize yourself with the different tactics and techniques listed in the framework.

Step 3. Choose one specific technique from any tactic that interests you.

Step 4. Conduct further research on the chosen technique to understand its details, real-world examples, and potential mitigation strategies.

Step 5. Write a brief summary of your findings, including the technique's description, its potential impact, and any recommended defensive measures.

Exercise 5-2: Exploring the MITRE ATLAS Framework

Objective: Explore the MITRE ATLAS Knowledge Base

Instructions:

Step 1. Visit the official MITRE ATLAS website (atlas.mitre.org).

Step 2. Explore the ATLAS knowledge base and its resources, including tactics, techniques, and case studies for machine learning systems.

Step 3. Select one specific technique or case study related to machine learning security that captures your interest.

Step 4. Research further on the chosen technique or case study to gain a deeper understanding of its context, implementation, and implications.

Step 5. Create a short presentation or a blog post summarizing the technique or case study, including its purpose, potential risks, and possible countermeasures.

Index

A

accelerators, AI, 50
access
    access control vulnerabilities, 169, 179
    initial, 150–152
    model, 152–153
account hijacking, 181
accountability, 19, 203, 212, 218, 249–252
accuracy, 246
Actor-Critic models, 42, 43
ADALINE, 4
adaptive defense mechanisms, 22
add_laplace_noise function, 189
adversarial attacks
    examples of, 108–109, 130
    implications of, 109
    security risks of, 106–107
    types of, 107–108
adversarial data, crafting, 150–152, 156
Adversarial Threat Landscape for Artificial-Intelligence Systems. *See* ATLAS
adversarial training, 183
agency, 119, 212, 215–218
agents, virtual, 44–45
aggregation, 185, 223
AGI (artificial general intelligence), 3
AI (artificial intelligence) attacks. *See* attacks
AI (artificial intelligence) technologies
    adaptability and revolutionary potential of, 17–19
    advancements of, 5–8
    AI-optimized hardware, 49–50
    algorithms. *See* algorithms
    ANI (artificial narrow intelligence), 52
    ANNs (artificial neural networks), 4
        CNNs (convolutional neural nets), 4–5, 40, 126–127
        DQN (Deep Q-Networks), 42
        FNNs (feedforward neural networks), 88, 90
        GANs (generative adversarial networks), 41
        RNNs (recurrent neural networks), 40–41, 88
    ASI (artificial super intelligence), 52
    biometrics, 45
    BtC case study, 50–52, 53–54
    capability-based, 52–53
    ChatGPT. *See* ChatGPT
    concept of, 8–11
    cyberrisk from, 23–25
    cybersecurity and, 21–23
    decision management, 45
    deep learning
        CNNs (convolutional neural nets), 40, 126–127

    - deep learning platforms, 47
    - definition of, 7
    - GANs (generative adversarial networks), 41
    - overview of, 40
    - RNNs (recurrent neural networks), 40–41
  - ethical challenges in. *See* ethics
  - fictional case study of, 2–3, 25–26
  - functionality-based, 52–53
  - function-based, 52–53
  - future trends and emerging developments, 56–57
  - general, 8
  - generative. *See* generative AI (artificial intelligence)
  - leading technologies for, 34
  - legal and regulatory compliance. *See* compliance
  - machine learning. *See* ML (machine learning)
  - narrow, 8
  - NLG (natural language generation), 43–44
  - origins of, 5
  - patentability of, 247–248
  - peer-to-peer networks convergence, 46–47
  - privacy and security challenges in. *See* privacy
  - RL (reinforcement learning), 41–43
  - RPA (Robotic Process Automation), 47
  - societal and ethical implications of, 54–56
  - speech recognition, 44
  - symbolic, 5
  - use cases for, 16–19
  - virtual agents, 44–45
  - wealth creation with, 17–19
  - XAI (explainable artificial intelligence), 7
- AI accelerators, 50
- AI BOMs (Bill of Materials)s, 117–118, 152, 176–177
- AI Model Cards, 177
- AI model security, 183–184
- AI Winter, 6
- AIA (Artificial Intelligence Act), 242, 244
- Alexa, 200, 226
- algorithms
  - algorithmic analysis, 193
  - applications of, 14–16
  - Apriori, 38–39
  - association rule mining, 38–39
  - backpropagation, 6
  - bias and fairness in, 13, 213–215, 242–243, 246, 259
  - biometrics, 45–46
  - choosing, 13–14
  - clustering, 37
  - comparison of, 11–13
  - decision management, 45
  - decision trees, 35
  - legal and regulatory compliance. *See* compliance
  - patentability of, 247–248
  - PCA (principal component analysis), 37–38
  - Q-learning, 41–42
  - RL (reinforcement learning)
    - DQN (Deep Q-Networks), 42
    - overview of, 41–42
    - policy gradient methods, 43
    - Q-learning, 42
  - supervised learning, 35–36
  - SVMs (support vector machines), 35–36
  - unsupervised learning
    - association rule mining, 38–39
    - clustering, 37
    - overview of, 36–37
    - PCA (principal component analysis), 37–38
  - use cases for, 16–19
  - variance, 13
- AlphaGo, 7
- Amazon
  - Alexa, 200, 226
  - investments in AI, 7
- ANI (artificial narrow intelligence), 52
- ANNs (artificial neural networks), 4
  - CNNs (convolutional neural nets), 4–5, 40, 126–127
  - DQN (Deep Q-Networks), 42
  - FNNs (feedforward neural networks), 88, 90

    GANs (generative adversarial networks), 41
    RNNs (recurrent neural networks), 40–41, 88
anomaly detection, 190
anonymization, data, 185–190, 220
antitrust laws, 250
APIs (application programming interfaces), vulnerabilities in, 180
Apple Music, 84
Apple Siri, 44, 200, 226
application-specific integrated circuits (ASICs), 49
Apriori algorithm, 38–39
ARIMA (Autoregressive Integrated Moving Average) models, 78
artifacts, machine learning, 157
artificial general intelligence (AGI), 3
Artificial Intelligence Act (AIA), 242, 244
artificial narrow intelligence (ANI), 52
artificial neural networks. *See* ANNs (artificial neural networks)
artificial super intelligence (ASI), 52
arXiv, 149
ASI (artificial super intelligence), 52
ASICs (application-specific integrated circuits), 49
association rule mining, 38–39
ATLAS, 146–147. *See also* TTPs (tactics, techniques, and procedures)
attacks
    adversarial
        examples of, 108–109
        implications of, 109
        security risks of, 106–107
        types of, 107–108
    attack staging, 158–159
    backdoor
        description of, 137, 138
        overview of, 155
        security risks of, 106–107
    case study, 143–145
    data poisoning
        defensive measures against, 138
        high-level overview of, 109–110
        methods of, 111–112
        overview of, 115–116
        security risks of, 106–107
        training, 115
    defensive measures against, 137–138
    DoS (denial of service), 116
    evasion
        adversarial examples, 130
        description of, 130–131, 138
        examples of, 131–135
    excessive agency, 119
    fictional case study of, 143–145
    insecure output handling, 115
    insecure plugin design, 118
    membership inference
        description of, 124–125, 138
        examples of, 124–130
        security risks of, 106–107
    MITRE ATLAS, 146–147
    model inversion
        description of, 135–136, 138
        examples of, 136
        mitigating, 136
    model theft, 138
        black-box knowledge, 120
        countermeasures against, 124
        description of, 119–121
        examples of, 121–124
        shadow models, 120
    overreliance, 119
    on-path attacks, 168
    prompt injection, 113–115
    supply chain vulnerabilities, 116–117, 181
    tactics and techniques
        AI and ML attack staging, 158–159
        collection, 157
        defense evasion, 156
        discovery, 156–157
        execution, 154–161
        exfiltration, 157, 159
        initial access, 150–152
        model access, 152–153
        persistence, 155
        prompt injection, 161
        reconnaissance, 147–149

        red-teaming, 162
        resource development, 150
attention mechanisms, 89
attribution, 192
audits, 190
authentication, vulnerabilities in, 179
Auto-GPT, 97–98
automation
    autonomous vehicles, 14, 17, 18, 259
    future trends and emerging developments, 258
    RPA (Robotic Process Automation), 47
    security operations, 22
autonomy, 212, 215–218
Autoregressive Integrated Moving Average (ARIMA) models, 78
autoregressive models
    advantages of, 78
    applications of, 78
    description of, 67
    types of, 77–78

B

backdoor attacks
    defensive measures against, 138
    description of, 137
    overview of, 155
    security risks of, 106–107
backpropagation algorithm, 6
behavioral analysis, 193
bell curve, 85–86
BernoulliRBM, 82
BERT, 89
bias and fairness
    algorithmic bias, 213–215
    forms of, 208–209
    nurturing, 224–225
    overview of, 13, 19, 212, 218
Bills of Materials. *See* BOMs (Bills of Materials)
biometrics, 45
Bitcoin, 46–47
black-box adversarial attacks, 107–108
black-box knowledge, 120
blockchain technology, 46–47, 194
BOMs (Bills of Materials), 117–118, 152
    AI Bills of Materials (AI BOMs), 176–177
    overview of, 117–118, 152
    SBOMs (Software Bill of Materials), 170
BtC case study, 50–52, 53–54
buffer overflow, 169

C

California Consumer Privacy Act (CCPA), 255
Cambridge Analytica scandal, 226
Canadian Institute for Advanced Research (CIFAR-10) dataset, 124
capability-based AI (artificial intelligence), 52–53
CCPA (California Consumer Privacy Act), 255
CDX (CycloneDX), 152, 171, 177
centralized governance, 252
CFPA (Consumer Financial Protection Act), 243, 244
Chat Markup Language (ChatML), 161
chatbots, 15
ChatGPT, 43. *See also* privacy
    ethical implications of, 203–205
    fictional case studies of, 227–232
    moral ramifications of
        accountability and responsibility, 203, 212
        autonomy and agency, 212, 215–218
        bias and fairness, 212, 213–215, 224–225
        differential privacy, 223–224
        diversity, 224–225
        federated learning, 223–224
        mitigation policies for ethical concerns, 218–223
    privacy concerns with, 203–205
ciphertext, 221
classification, 13
cloud computing
    CSPs (cloud service providers), 181
    metadata exploitation, 181
    security vulnerabilities, 179–181
clustering, 37
CNNs (convolutional neural nets), 4–5, 40, 126–127
Colaboratory, 150

collaboration, international, 255–257
collection, 157
Common Security Advisory Framework (CSAF), 171
competition and antitrust laws, 250
compliance
  case study of, 257–261
  copyright protection, 248
  data protection laws, 245–247
  future trends and outlook in, 257–260
  GDPR (General Data Protection Regulation), 209–210, 218–219, 245–247, 255
  globally accepted compliance frameworks, 244–245, 255
  governance and risk management, 252–255
  international collaboration and standards, 255–257
  international legal frameworks for, 242–245
  IP (intellectual property) issues, 247–249
  liability and accountability, 249–252
  patentability, 247–248
  trademark protection, 249
compliance officers, role in AI governance, 254
Component_not_present status, 172
"Computing Machinery and Intelligence" (Turing), 4
confabulations, 119
confidentiality, 246
Confluence, 158
consent, 208, 245
Consumer Financial Protection Act (CFPA), 243, 244
consumer protection laws, 250
continuous security monitoring, 183
contract laws, 250
convergence, GANs (generative adversarial networks), 69
conversational AI
  IP (intellectual property) issues, 247–249
  legal and regulatory compliance
    case study of, 257–261
    data protection laws, 245–247
    future trends and outlook in, 257–260
    globally accepted compliance frameworks, 244–245, 255
    governance and risk management, 252–255
    international collaboration and standards, 255–257
    IP (intellectual property) issues, 247–249
    liability and accountability, 249–252
    overview of, 242–245
convolutional layer, CNNs (convolutional neural nets), 4–5, 40, 126–127
convolutional neural nets. *See* CNNs (convolutional neural nets)
copyright protection, 248
credential misuse, 150–152
cross-site request forgery (CSRF), 115
cross-site scripting (XSS), 115
CSAF (Common Security Advisory Framework), 171
CSAF VEX JSON document, 173–176
CSPs (cloud service providers), 181
CSRF (cross-site request forgery), 115
cyberrisk, 23–25
cybersecurity, 21–23, 250, 259
CycloneDX (CDX), 152, 171, 177

D

Dartmouth Conference, 5
data anonymization, 185–190, 220
data augmentation
  GANs (generative adversarial networks), 69
  normalizing flows, 88
data collection, 193
  bias and fairness in, 208–209
  consent, 208
  ethical privacy concerns in, 206–211, 218
  privacy issues with, 203–205
data compromise, 150–152
data exposure and leakage, 180
data generalization, 222
data governance teams, role in AI governance, 254
data masking, 185

data minimization, 245, 246
data perturbation, 222
data poisoning attacks
    defensive measures against, 138
    high-level overview of, 109–110
    methods of, 111–112
    overview of, 115–116
    security risks of, 106–107
    training, 115
data preservation, 193
data privacy, 259
data protection, 19
    data practices for, 218–223
    data protection laws, 245–247, 250
data pseudonymization, 185, 210, 220–222
data storage
    ethical privacy concerns in, 206–211, 218
    security, 184–185
    vulnerabilities in, 169–170
data subject rights, 246
data swapping, 185, 222
Datasets library, HuggingFace, 96
decentralized governance, 252
decision management, 45
decision trees, 35
decoders, transformer architecture, 91
Deep Blue, 7
deep learning
    CNNs (convolutional neural nets), 4–5, 40, 126–127
    definition of, 7
    ensemble learning compared to, 12–13
    GANs (generative adversarial networks), 41
    overview of, 40
    RNNs (recurrent neural networks), 40–41
Deep Learning Revolution, 7
Deep Q-Networks (DQN), 42
deepfakes, 226, 259
defense evasion, 156
DENDRAL, 5–6
denial of service (DoS), 116
denoising, 88
density estimation, 85–88
dependency management
    dependency confusion, 145
    SBOMs (Software Bill of Materials), 170
    vulnerabilities, 169–170
Descartes, René, 3
design, system/infrastructure, 182–183
differential privacy, 136, 183, 187–189, 193, 220, 223–224
digital signatures, 221–222
dimensionality reduction
    definition of, 80
    PCA (principal component analysis), 37–38
direct prompt injection attacks, 113–115
discovery, 156–157
discrimination. *See* bias and fairness
discriminators, GANs (generative adversarial networks), 68–69
diversity, 224–225
domain adaptation, 88
DoS (denial of service), 116
DQN (Deep Q-Networks), 42
drug development and discovery, 17

E

EAD (Ethically Aligned Design), 255
EDPB (European Data Protection Board), 246
ELBO (Evidence Lower Bound), 74
emerging developments
    AI (artificial intelligence) technologies, 56–57
    legal and regulatory compliance, 257–260
employment and labor laws, 250
encoders, VAEs (variational autoencoders)
    advantages of, 75
    basic example of, 75–77
    challenges in training, 74
    description of, 67
    high-level overview of, 73–74
encryption, 183, 210, 220
energy efficiency, 15
energy function, 79
energy-based generative models, 79–80
engineering, prompt, 93–95
enhanced threat detection, 22

ensemble learning, deep learning compared to, 12–13
ePrivacy Directive, 250
epsilon, 134–135
Ethically Aligned Design (EAD), 255
ethics
  accountability and responsibility, 203, 212, 218
  autonomy and agency, 212, 215–218
  balancing benefits of AI with, 200–201
  bias and fairness
    algorithmic bias, 213–215
    forms of, 208–209
    nurturing, 224–225
    overview of, 212, 213–215, 218
  challenges in, 19–20
  concerns related to, 203–205
  data collection and data storage, 206–211, 218
  differential privacy, 136, 183, 187–189, 193, 220, 223–224
  diversity, 224–225
  examples of, 226
  federated learning, 193, 223–224, 258
  fictional case studies of, 227–232
  governance and risk management, 252–255
  importance of, 201–203
  mitigation policies for, 218–223
  overview of, 19–20, 54–56
  regulations, 250
  transparency, 19, 218, 245, 246
ethics committees, role in AI governance, 254
“Ethics Guidelines for Trustworthy AI”242
European Data Protection Board (EDPB), 246
European Union (EU)
  AIA (Artificial Intelligence Act), 242, 244
  EDPB (European Data Protection Board), 246
  “Ethics Guidelines for Trustworthy AI”242
  GDPR (General Data Protection Regulation), 209–210, 218–219, 245–247, 255
Eva case study, 2–3, 25–26
evaluation, 9–11
evasion attacks
  adversarial examples, 130
  defensive measures against, 138
  description of, 130–131
  examples of, 131–135
Evidence Lower Bound (ELBO), 74
excessive agency, 119
execution of attacks, 154–161
executive leadership, role in AI governance, 254
exfiltration, 157, 159
expert systems, 5
explainable AI (XAI), 256–258
explainable artificial intelligence (XAI), 7
exposure, data, 180

F

Facebook, Cambridge Analytica scandal, 226
Fair Credit Reporting Act (FCRA), 243, 244
Fair Information Practices Principles (FIPPs), 255
fairness
  algorithmic bias, 213–215
  forms of, 208–209
  nurturing, 224–225
  overview of, 212, 213–215, 218, 242–243, 246, 259
FakeMedAI case study, 143–145
false positives, 22
fast gradient sign method (FGSM), 131
FCRA (Fair Credit Reporting Act), 243, 244
feature extraction, 8–11
feature learning, 80
feature selection, 14, 37–38
Federal Trade Commission (FTC), 255
federated learning, 193, 223–224, 258
feedforward neural networks (FNNs), 88, 90
Feigenbaum, Edward, 6
FGSM (fast gradient sign method), 131
field-programmable gate arrays (FPGAs), 49
finance, 14, 17–18
FIPPs (Fair Information Practices Principles), 255
firewalls, 23, 190
flexibility
  autoregressive models, 78
  normalizing flows, 87
  RBMs (restricted Boltzmann machines), 83

flows, normalizing
  advantages of, 87
  applications of, 88
  description of, 67
  high-level overview of, 84–86
  references on, 87
  types of, 86–87
FNNs (feedforward neural networks), 88, 90
forensic investigations, 192–193
FPGAs (field-programmable gate arrays), 49
FP-Growth (Frequent Pattern-Growth), 39
frameworks, legal and regulatory compliance, 242–245, 255
fraud detection, AI applications in, 16
Frequent Pattern-Growth (FP-Growth), 39
FTC (Federal Trade Commission), 255
fully connected layer, CNNs (convolutional neural nets), 127
functionality-based AI (artificial intelligence), 52–53
function-based AI (artificial intelligence), 52–53
future trends
  AI (artificial intelligence) technologies, 56–57
  legal and regulatory compliance, 257–260

G

G20, international collaboration and standards, 256
GANs (generative adversarial networks), 41
  applications of, 69
  basic example of, 70–73
  challenges in training, 69
  description of, 67
  discriminators, 68–69
  generators, 68–69
  high-level overview of, 68–69
  origins of, 68
  tools and libraries, 70
Gated Recurrent Unit (GRU), 88
Gaussian distribution, 85–86
GDPR (General Data Protection Regulation), 209–210, 218–219, 245–247, 255
general AI (artificial intelligence), 8
General Data Protection Regulation (GDPR), 209–210, 218–219, 245–247, 255
General Problem Solver, 5
generalization, data, 222
generative adversarial networks. *See* GANs (generative adversarial networks)
generative AI (artificial intelligence)
  autoregressive models
    advantages of, 78
    applications of, 78
    description of, 67
    types of, 77–78
  comparison of models, 66–67
  GANs (generative adversarial networks)
    applications of, 69
    basic example of, 70–73
    challenges in training, 69
    description of, 67
    discriminators, 68–69
    generators, 68–69
    high-level overview of, 68–69
    origins of, 68
    tools and libraries, 70
  normalizing flows
    advantages of, 87
    applications of, 87
    description of, 67
    high-level overview of, 84–86
    references on, 87
    types of, 86–87
  RBMs (restricted Boltzmann machines)
    advantages of, 83
    applications of, 83–84
    basic example of, 82–83
    challenges in training, 81–82
    description of, 67
    high-level overview of, 79–82
  transformative potential of, 66
  VAEs (variational autoencoders)
    advantages of, 75
    basic example of, 75–77
    challenges in training, 74
    description of, 67
    high-level overview of, 73–74

Generative Discriminator Guided Sequence Generation (GeDi), 162
generative modeling, 73, 88
generative pre-trained transformers
    Auto-GPT, 97–98
    case study of, 257–261
    data protection laws, 245–247
    future trends and outlook in, 257–260
    globally accepted compliance frameworks, 244–245, 255
    governance and risk management, 252–255
    GPT-3 (Generative Pre-trained Transformer 3), 7
    GPT-4 (Generative Pre-trained Transformer 4), 91–93
    international collaboration and standards, 255–257
    IP (intellectual property) issues, 247–249
    liability and accountability, 249–252
    overview of, 242–245
generators
    GANs (generative adversarial networks), 68–69
    VAEs (variational autoencoders)
        advantages of, 75
        basic example of, 75–77
        challenges in training, 74
        description of, 67
        high-level overview of, 73–74
Gerchick, Marissa, 152, 177
Gibbs sampling, 81
git, 95
git-lfs, 95
Global Partnership on AI (GPAI), 256
GloVe, 88–89
Goodfellow, Ian, 68
Google
    Google Assistant, 44, 226
    Google Colaboratory, 150
    investments in AI, 7
governance and risk management, 252–255
GPAI (Global Partnership on AI), 256
GPT. *See* generative pre-trained transformers
GPUs (graphics processing units), 49, 195
GRU (Gated Recurrent Unit), 88

H

hacking AI systems
    case study, 143–145
    FakeMedAI case study, 143–145
    MITRE ATLAS, 146–147
    tactics and techniques
        AI and ML attack staging, 158–159
        collection, 157
        defense evasion, 156
        discovery, 156–157
        execution, 154–161
        exfiltration, 157, 159
        initial access, 150–152
        model access, 152–153
        persistence, 155
        prompt injection, 161
        reconnaissance, 147–149
        red-teaming, 162
        resource development, 150
hallucinations, 92, 95, 119
hardware, AI-optimized, 49–50
HDF5 (.h5), 144
Health Insurance Portability and Accountability Act (HIPAA), 243, 244
healthcare, AI applications in, 14, 17
hijacking, account, 181
HIPAA (Health Insurance Portability and Accountability Act), 243, 244
homomorphic encryption, 194, 220
HuggingFace, 70, 95–96, 152
hybrid governance, 253
hyperparameter sensitivity, GANs (generative adversarial networks), 69

I

IDS (intrusion detection systems), 23, 190
IEC (International Electrotechnical Commission), 256

IEEE (Institute of Electrical and Electronics Engineers), 256
    EAD (Ethically Aligned Design), 255
    industry standards and best practices, 244
image recognition, 15, 17
image synthesis
    autoregressive models, 78
    GANs (generative adversarial networks), 69
incident response, 183, 190–191
independent software vendors (ISVs), 113
India, legal and regulatory frameworks, 243, 244
indirect prompt injection attacks, 113–115
indiscriminate data poisoning attacks, 111
inference
    API access, 153
    normalizing flows, 87, 88
    VAEs (variational autoencoders), 75
informed consent, 208
infrastructure security
    AI model security, 183–184
    audits, 190
    blockchain technology, 194
    data anonymization, 185–190
    data storage and processing systems, 184–185
    differential privacy, 193, 220
    federated learning, 193, 223–224, 258
    firewalls, 190
    forensic investigations, 192–193
    homomorphic encryption, 194
    incident response, 190–191
    secure design principles, 182–183
    SMPC (secure multi-party computation), 194–195
    threat detection, 190–191
    vulnerabilities
        AI Bills of Materials (AI BOMs), 176–177
        cloud security vulnerabilities, 179–181
        network security, 168
        patch management, 170
        physical security, 169
        SBOMs (Software Bill of Materials), 170
        system security, 169–170
        VEX (Vulnerability Exploitability eXchange), 171–176
initial access, 150–152
Inline_mitigations_already_exist status, 173
input layer, CNNs (convolutional neural nets), 127
inquality, 19
insecure output handling, 115
Institute of Electrical and Electronics Engineers (IEEE), 256
integrations, vulnerabilities in, 180
integrity, 246
intellectual property. *See* IP (intellectual property) issues
intellectual property (IP) issues, 247–249, 250
international collaboration and standards, 242–245, 250, 255–257
International Electrotechnical Commission (IEC), 256
International Organization for Standardization (ISO), 255, 256
International Telecommunication Union (ITU), 256
interpretability, autoregressive models, 78
intrusion detection and prevention systems (IDS/IPS), 23, 190
inverse problems, normalizing flows, 88
IP (intellectual property) issues, 247–249, 250
ISO (International Organization for Standardization), 255, 256
ISVs (independent software vendors), 113
ITU (International Telecommunication Union), 256

J

jailbreaking, 113–115
job displacement, 19

K

k-anonymity, 185–187
Kasparov, Garry, 7
Keras, 47, 70

L

labor laws, 250
lag, autoregressive models, 77
LangChain, 115

Laplace distribution, 189–190
large language models. *See* LLMs (large language models)
layer normalization, transformer architecture, 90
leading technologies, 34
leakage, data, 180
least privilege principle, 183
Lederberg, Joshua, 6
legal compliance. *See* compliance
libraries
  GANs (generative adversarial networks), 70
  HuggingFace, 70, 95–97, 152
  PyTorch, 47, 70, 144
  TensorFlow, 47, 70, 144
limited memory, 53
linear regression, 4, 35
Linux Foundation, 152, 177
LLMs (large language models). *See also* attacks
  Auto-GPT, 97–98
  evolution of, 88–89
  HuggingFace, 70, 95–96, 152
  OpenAI GPT-4, 91–93
  OWASP Top Ten for LLMs
    DoS (denial of service), 116
    excessive agency, 119
    insecure output handling, 115
    insecure plugin design, 118
    model theft, 119–124
    overreliance, 119
    overview of, 112–113
    prompt injection attacks, 113–115
    supply chain vulnerabilities, 116–117
    training data poisoning, 115–116
  prompt engineering, 93–95
  transformer architecture, 89–91
  trust boundary, 114
Logic Theorist, 5
LSTM (Long Short-Term Memory), 88

M

MA (Moving Average) models, 78
machine learning. *See* ML (machine learning)
malware detection, 17
Manifest, 117, 177
man-in-the-middle attacks, 168
manipulation, 19
mapping topologies, self-organized development of, 5
Markov Chain Monte Carlo (MCMC), 81, 84
masking data, 185
McCarthy, John, 5
McCulloch, W. S.4
McCulloch-Pitts function, 4
MCMC (Markov Chain Monte Carlo), 81, 84
medical diagnostics, AI applications in, 17
medical imaging, AI applications in, 16
membership inference attacks
  defensive measures against, 138
  description of, 124–125
  examples of, 124–130
  security risks of, 106–107
memory
  limited, 53
  LSTM (Long Short-Term Memory), 88
metadata exploitation, 181
metrics, GANs (generative adversarial networks), 69
microsegmentation, 183
Microsoft, investments in AI, 7
mind, theory of, 53
minimization, data, 245, 246
Minsky, Marvin, 5
misconfigured access controls, 179
misinformation, 19
Mitchell, Margaret, 152, 177
Mithril Security, 152
mitigation policies for ethical concerns, 218–223
MITRE ATLAS, 146–147. *See also* TTPs (tactics, techniques, and procedures)
ML (machine learning)
  adaptability and revolutionary potential of, 17–19
  advancements of, 6–8
  algorithms. *See* algorithms
  artifacts, 157
  biometrics, 45
  BtC case study, 50–52, 53–54

ChatGPT, 43
concept of, 16–17
cyberrisk from, 23–25
in cybersecurity, 21–23
decision management, 45
deep learning, 40–41
  CNNs (convolutional neural nets), 4–5, 40, 126–127
  definition of, 7
  ensemble learning compared to, 12–13
  GANs (generative adversarial networks), 41
  overview of, 40
  RNNs (recurrent neural networks), 40–41
deep learning platforms, 47
ensemble learning, 12–13
ethical challenges in. *See* ethics
evaluation and validation, 9–11
feature extraction, 8–11
feature selection, 14
federated learning, 193, 223–224, 258
future trends and emerging developments, 56–57
leading technologies for, 34
model selection and training, 9–11
origins of, 6–8
overfitting, 9–10
peer-to-peer networks convergence, 46–47
privacy and security challenges in. *See* privacy
RL (reinforcement learning), 41–43
RPA (Robotic Process Automation), 47
societal and ethical implications of, 54–56
speech recognition, 44
supervised learning, 12, 35–36
training data, 8–11
unsupervised learning, 12
  association rule mining, 38–39
  clustering, 37
  overview of, 36–37
  PCA (principal component analysis), 37–38
virtual agents, 44–45
wealth creation with, 17–19
mode collapse, GANs (generative adversarial networks), 69
model access, techniques for, 152–153
model analysis, 193
model design, 136
model hardening, 184
Model Hub, HuggingFace, 96
model inversion attacks
    defensive measures against, 138
    description of, 135–136
    examples of, 136
    mitigating, 136
model selection and training, 9–11
model serialization, 154–155
model theft
    black-box knowledge, 120
    countermeasures against, 124
    defensive measures against, 138
    description of, 119–121
    examples of, 121–124
    shadow models, 120
moral ramifications of AI and ChatGPT deployment
    accountability and responsibility, 203, 212
    autonomy and agency, 212, 215–218
    bias and fairness, 212, 213–215
    mitigation policies for ethical concerns, 218–223
Moving Average (MA) models, 78
multihead attention, transformer architecture, 90
MYCIN, 5–6

N

narrow AI (artificial intelligence), 8
National Institute of Standards and Technology (NIST), 191, 255
natural language generation (NLG), 43–44
natural language processing (NLP), 14, 15, 78
network security vulnerabilities, 168
network traffic analysis, 191
Neumann, John von, 4
Neural Autoregressive models, 78

neural networks. *See* ANNs (artificial neural networks)
neural processing units (NPUs), 50, 195
Newell, Allen, 4, 5
NIST (National Institute of Standards and Technology), 191, 255
NLG (natural language generation), 43–44
NLP (natural language processing), 14, 15, 78
nonlinear function, 4
normal distribution, 85–86
normalizing flows
  advantages of, 87
  applications of, 88
  description of, 67
  high-level overview of, 84–86
  references on, 87
  types of, 86–87
NPUs (neural processing units), 50, 195

O

OASIS, CSAF (Common Security Advisory Framework), 171
obfuscation, 185
OECD (Organisation for Economic Co-operation and Development), 243–244, 255, 256
ONNX (.onnx), 144
on-path attacks, 168
OpenAI
  ChatML, 161
  GPT-3, 7
  GPT-4, 91–93
open-source model compromise, 150–152
OpenVEX, 171
Organisation for Economic Co-operation and Development (OECD), 243–244, 255, 256
output handling, insecure, 115
output layer, CNNs (convolutional neural nets), 127
overfitting, 9–10
overreliance, 119
OWASP CycloneDX, 177
OWASP Top Ten for LLMs
  DoS (denial of service), 116
  excessive agency, 119
  insecure output handling, 113–115
  insecure plugin design, 118
  model theft, 119–124
  overreliance, 119
  overview of, 112–113
  prompt injection attacks, 113–115
  supply chain vulnerabilities, 116–117
  training data poisoning, 115–116
Ozoani, Ezi, 152, 177

P

P2P (peer-to-peer) networks convergence, 46–47
PAI (Partnership on AI), 255
partition function, 79
patch management, 169, 170, 183
patentability of AI, 247–248
pattern recognition, self-organizing adaptive, 5
payload injections, 155
PCA (principal component analysis), 37–38, 83
PDF (probability density function), 84
peer-to-peer networks convergence, 46–47
perceptron, 4
persistence phase of attacks, 155
personal assistants, 17
perturbation, data, 222
physical security vulnerabilities, 169
Pickle (.pkl), 144
Pitts, W.4
platforms, deep learning, 47
Plug and Play Language Models (PPLM), 162
plugin design, 118
poisoned training data, 150–152, 155
policy gradient methods, 42
pooling layer, CNNs (convolutional neural nets), 127
positional encoding, transformer architecture, 90
PPLM (Plug and Play Language Models), 162
PPO (Proximal Policy Optimization), 42, 43
predictive maintenance, 16, 191
pre-trained models, 83, 89
principal component analysis (PCA), 37–38, 83

privacy
  accountability and responsibility, 203, 212, 218
  autonomy and agency, 212, 215–218
  balancing benefits of AI with, 200–201
  bias and fairness, 212, 218
    algorithmic bias, 213–215
    forms of, 208–209
    nurturing, 224–225
  concerns related to, 203–205
  data collection and data storage, 206–211, 218
  differential, 136, 183, 187–189, 193, 220, 223–224
  diversity, 224–225
  examples of, 226
  federated learning, 193, 223–224, 258
  fictional case studies of, 227–232
  future trends and emerging developments, 259
  importance of, 201–203
  mitigation policies for, 218–223
  overview of, 19, 20–21
  transparency, 19, 218, 245, 246
private dataset targeting, 150–152
probabilistic framework, VAEs (variational autoencoders), 75
probability density function (PDF), 84
procedures. *See* TTPs (tactics, techniques, and procedures)
product liability laws, 250
prompt engineering, 93–95
prompt injection, 113–115, 161
Proximal Policy Optimization (PPO), 42, 43
pseudonymization, 185, 210, 220–222
purpose limitation, 246
PyPI package manager, 145
PyTorch library, 47, 70, 144

Q

Q-learning algorithm, 41–42
quantum computing, 259

R

RBF (radial basis function), 35–36
RBMs (restricted Boltzmann machines)
  advantages of, 83
  applications of, 83–84
  basic example of, 82–83
  challenges in training, 81–82
  description of, 67
  high-level overview of, 79–82
reactive machines, 53
real-time incident response, 22
recognition model, 73
recommendation systems, 15, 17
reconnaissance, 147–149
rectified linear unit (ReLU) layer, CNNs (convolutional neural nets), 127
recurrent neural networks (RNNs), 40–41, 88
red-teaming AI models, 162
regression
  definition of, 13
  linear, 35
regular patching and updating, 183
regularization, 136
regulatory compliance. *See* compliance
reinforcement learning (RL), 41–43
remediation, 193
representation learning, 80
research and development, HuggingFace, 96
residual connections, transformer architecture, 90
resource development, 150
responsibility, 212
restricted Boltzmann machines. *See* RBMs (restricted Boltzmann machines)
retention and deletion, 211
risk management
  governance and, 252–255
  overview of, 23–25
  OWASP Top Ten for LLMs
    DoS (denial of service), 116
    excessive agency, 119
    insecure output handling, 115
    insecure plugin design, 118

        model theft, 119–124
        overreliance, 119
        overview of, 112–113
        prompt injection attacks, 113–115
        supply chain vulnerabilities, 116–117
        training data poisoning, 115–116
    system/infrastructure
        AI Bills of Materials (AI BOMs), 176–177
        AI model security, 183–184
        audits, 190
        blockchain technology, 194
        cloud security vulnerabilities, 179–181
        data anonymization, 185–190
        data storage and processing systems, 184–185
        differential privacy, 193, 220
        federated learning, 193, 223–224, 258
        firewalls, 190
        forensic investigations, 192–193
        homomorphic encryption, 194
        incident response, 190–191
        network security vulnerabilities, 168
        patch management, 169, 170, 183
        physical security vulnerabilities, 169
        SBOMs (Software Bill of Materials), 170
        secure design principles, 182–183
        SMPC (secure multi-party computation), 194–195
        system security vulnerabilities, 169–170
        threat detection, 190–191
        VEX (Vulnerability Exploitability eXchange), 171–176
RL (reinforcement learning), 41–43
RNNs (recurrent neural networks), 40–41, 88
Robotic Process Automation (RPA), 47, 251
Rosenblatt, Frank, 4
RPA (Robotic Process Automation), 47, 251

S

S2C (service-to-consumer), 47
sampling, Gibbs, 81
SBOMs (Software Bill of Materials), 152, 170
scalability, autoregressive models, 78
Seasonal Decomposition of Time Series (STL) models, 78
sector-specific regulations, 250
secure multi-party computation (SMPC), 194–195, 220
security. *See also* attacks
    automated, 22
    cybersecurity, 21–23, 259
    data security, 245
    data transmission protocols, 183
    need for, 105–107
    overview of, 20–21
    OWASP Top Ten for LLMs
        DoS (denial of service), 116
        excessive agency, 119
        insecure output handling, 115
        insecure plugin design, 118
        model theft, 119–124
        overreliance, 119
        overview of, 112–113
        prompt injection attacks, 113–115
        supply chain vulnerabilities, 116–117
        training data poisoning, 115–116
    system/infrastructure
        AI Bills of Materials (AI BOMs), 176–177
        AI model security, 183–184
        audits, 190
        blockchain technology, 194
        cloud security vulnerabilities, 179–181
        data anonymization, 185–190
        data storage and processing systems, 184–185
        differential privacy, 193, 220
        federated learning, 193, 223–224, 258
        firewalls, 190
        forensic investigations, 192–193
        homomorphic encryption, 194
        incident response, 190–191
        network security, 168
        patch management, 169, 170, 183
        physical security, 169
        SBOMs (Software Bill of Materials), 170
        secure design principles, 182–183

    - SMPC (secure multi-party computation), 194–195
    - system security, 169–170
    - threat detection, 190–191
    - VEX (Vulnerability Exploitability eXchange), 171–176
- Sedol, Lee, 7
- self-attention mechanism, transformer architecture, 90
- self-awareness, 53
- self-organizing adaptive pattern recognition, 5
- SentinelAI, 176
- sequence-to-sequence models, 89
- serialization, model, 154–155
- server-side request forgery (SSRF), 115
- service-to-consumer (S2C), 47
- shadow models, 120
- SharePoint, 158
- Shaw, Cliff, 4
- signatures, digital, 221–222
- Simon, Herbert A.4, 5
- Siri, 44, 200, 226
- SMPC (secure multi-party computation), 194–195, 220
- societal implications, 54–56
- Software Bill of Materials (SBOMs), 152, 170
- software interfaces, vulnerabilities in, 169
- spam detection, 17
- SPDX specification, 171, 177
- speech recognition, 15, 17, 18, 44
- Spotify, 84
- SQL Server.158
- SSRF (server-side request forgery), 115
- stability
  - normalizing flows, 87
  - VAEs (variational autoencoders), 75
- staging, AI and ML attacks, 158–159
- stakeholders, role in AI governance, 254–255
- standards, 255–257
- stealing. *See* theft, model
- STL (Seasonal Decomposition of Time Series) models, 78
- stock market trading, 17
- storage, data. *See* data storage
- storage limitation, 246
- style transfer, GANs (generative adversarial networks), 69
- StyleGAN/StyleGAN2 library, 70
- subsampling layer, CNNs (convolutional neural nets), 127
- super-resolution, GANs (generative adversarial networks), 69
- supervised learning, 12, 35–36
- supply chain vulnerabilities, 116–117, 150–152, 181
- support vector machines (SVMs), 35–36
- surveillance systems, 14
- SVMs (support vector machines), 35–36
- swapping data, 185, 222
- symbolic AI, 5
- system security
  - AI model security, 183–184
  - audits, 190
  - blockchain technology, 194
  - data anonymization, 185–190
  - data storage and processing systems, 184–185
  - differential privacy, 193, 220
  - federated learning, 193
  - firewalls, 190
  - forensic investigations, 192–193
  - homomorphic encryption, 194
  - incident response, 190–191
  - secure design principles, 182–183
  - SMPC (secure multi-party computation), 194–195
  - threat detection, 190–191
  - vulnerabilities, 169–170
    - AI Bills of Materials (AI BOMs), 176–177
    - cloud security vulnerabilities, 179–181
    - network security, 168
    - patch management, 169, 170, 183
    - physical security, 169
    - SBOMs (Software Bill of Materials), 170
    - system security, 169–170
    - VEX (Vulnerability Exploitability eXchange), 171–176

T

tactics. *See* TTPs (tactics, techniques, and procedures)
targeted data poisoning attacks, 111
t-distributed stochastic neighbor embedding (t-SNE), 83–84
techniques. *See* TTPs (tactics, techniques, and procedures)
tensor processing units (TPUs), 49, 195
TensorFlow, 47, 70, 144
text-to-image generation, GANs (generative adversarial networks), 69
theft, model
  black-box knowledge, 120
  countermeasures against, 124
  defensive measures against, 138
  description of, 119–121
  examples of, 121–124
  shadow models, 120
theory of mind, 53
threat detection, 190–191
threat intelligence, 22
time series forecasting, 78
Tokenizers library, HuggingFace, 96
tort laws, 250
TPUs (tensor processing units), 49, 195
trademark protection, 249
traffic pattern prediction, 17
training, 8–11
  data poisoning, 115–116
  GANs (generative adversarial networks), 69
  RBMs (restricted Boltzmann machines), 81–82
  VAEs (variational autoencoders), 74
transfer learning, normalizing flows, 88
transformer models, 89–91
Transformers library, HuggingFace, 96
transparency, 19, 218, 245, 246
trust boundary, 114
TTPs (tactics, techniques, and procedures)
  AI and ML attack staging, 158–159
  collection, 157
  defense evasion, 156
  discovery, 156–157
  execution, 154–161
  exfiltration, 157, 159
  initial access, 150–152
  model access, 152–153
  persistence, 155
  prompt injection, 161
  reconnaissance, 147–149
  red-teaming, 162
  resource development, 150
Turing, Alan, 4, 5
Turing machine/Turing test, 4

U

unemployment, 19
United Kingdom, legal and regulatory frameworks, 243, 244
United Nations (UN)
  international collaboration and standards, 256
  legal and regulatory frameworks, 242
unpatched systems, 169
unsupervised learning, 12
  association rule mining, 38–39
  clustering, 37
  overview of, 36–37
  PCA (principal component analysis), 37–38
user control, 220

V

VAEs (variational autoencoders)
  advantages of, 75
  basic example of, 75–77
  challenges in training, 74
  description of, 67
  high-level overview of, 73–74
validation, 9–11
variance, 13
variational autoencoders. *See* VAEs (variational autoencoders)
variational inference, normalizing flows, 88

vehicles, autonomous, 14, 17, 18, 259
VEX (Vulnerability Exploitability eXchange), 171–176
virtual agents, 44–45
virtual assistants, 15
vulnerabilities, system/infrastructure. *See also* attacks
  AI Bills of Materials (AI BOMs), 176–177
  cloud security vulnerabilities, 179–181
  network security, 168
  patch management, 169, 170, 183
  physical security, 169
  SBOMs (Software Bill of Materials), 170
  system security, 169–170
  VEX (Vulnerability Exploitability eXchange), 171–176
Vulnerable_code_cannot_be_controlled_by_adversary status, 172
Vulnerable_code_not_in_execute_path status, 172
Vulnerable_code_not_present status, 172

W

wealth creation, AI solutions for, 17–19
white-box access, 107–108, 153
Word2Vec, 88–89
World Economic Forum, s industry standards and best practices, 244

X-Y-Z

XAI (explainable artificial intelligence), 7, 256–258
XSS (cross-site scripting), 115

YAML, 144