

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 07-05-2021	2. REPORT TYPE Master of Military Studies (MMS) thesis	3. DATES COVERED (From - To) AY 2020-2021
--	--	---

4. TITLE AND SUBTITLE Negative, suppression is VOID: AI, Deception, and Fighting Machines	5a. CONTRACT NUMBER N/A
	5b. GRANT NUMBER N/A
	5c. PROGRAM ELEMENT NUMBER N/A

6. AUTHOR(S) Major Brian Jonathan Strom, AY2020-2021	5d. PROJECT NUMBER N/A
	5e. TASK NUMBER N/A
	5f. WORK UNIT NUMBER N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) USMC Command and Staff College Marine Corps University 2076 South Street Quantico, VA 22134-5068	8. PERFORMING ORGANIZATION REPORT NUMBER N/A
--	--

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) N/A

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This paper examines the implications for the introduction of AI and autonomous weapons systems to the battlefield. An overview of the United States' current efforts in AI is compared to those of Russia and China. This paper then explains how machine learning works in order to examine how it can be deceived. This paper then examines the current state of research into deceiving AI via a literature review, focusing on adversarial examples and data poisoning attacks. This paper then postulates a threat model on how an adversary could leverage either type of attack in order to develop a method of comparing the relative strength of the two. These attacks are then conducted against simulated data.

15. SUBJECT TERMS
Artificial Intelligence, Machine Learning, Deception, MILDEC

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			USMC Command and Staff College
Unclass	Unclass	Unclass	UU	66	19b. TELEPHONE NUMBER (Include area code) (703) 784-3330 (Admin Office)

*United States Marine Corps
Command and Staff College
Marine Corps University
2076 South Street
Marine Corps Combat Development Command
Quantico, Virginia 22134-5068*

MASTER OF MILITARY STUDIES

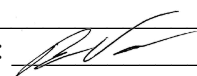
Negative, suppression is VOID: AI, Deception, and Fighting Machines

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF MILITARY STUDIES

Major Brian J. Strom, USMC

AY 2020-21

MMS Mentor & Oral Defense Committee Member:

Approved: 
Date: 5/8/21

MMS Mentor & Oral Defense Committee Member:

Olivia Brown
Approved: Olivia Brown
Date: 5/7/21

Oral Defense Committee Member:

Approved: _____
Date: _____

Oral Defense Committee Member:

Approved: _____
Date: _____

Title: Negative, suppression is VOID: AI, Deception, and Fighting Machines

Author: Major Brian Jonathan Strom, AY2020-2021

Thesis: Artificial Intelligence (AI) will fundamentally challenge current warfighting practices. The threat of AI-enabled battlefield technology means that the United States should invest in military applications of specific attacks on the machine learning (ML) algorithms underlying AI technology in order to be able to deceive these algorithms. This paper proposes a new framework for analyzing the robustness of AI from the perspective of how to deceive machine learning algorithms and compares the relative strength of two different attacks on ML. The Department of Defense (DoD) likely has under invested in AI research given future battlefield impact of this technology, but needs to invest more in robustness and deception with regards to Artificial Intelligence (AI).

Discussion: The US is investing in AI, but our adversaries, such as China and Russia, are both investing tremendous effort in fielding autonomous systems. This means that the US military will likely face autonomous weapons systems or AI-enabled battlefield technology. However, current tactics will likely be challenged by AI, and AI will likely outperform humans in the near future on the battlefield. There is an emerging field of research into ways that ML can be deceived, however, there isn't a framework comparing different methods of how to attack these algorithms. This paper proposes such a framework, experiments in using it against simple ML algorithms, and draws conclusions based on their performance and the experience of this experimentation.

Conclusion: The current pace of change in AI research and the ever-increasing levels of investment means that the state-of-the-art in AI research from five years ago is commonplace now. We need to start thinking about techniques for fighting AI while AI still underperforms humans in warfare so that we can maintain our edge when AI represents the dominant battlefield technology.

Abstract: This paper examines the implications for the introduction of AI and autonomous weapons systems to the battlefield. An overview of the United States' current efforts in AI is compared to those of Russia and China. This paper then explains how machine learning works in order to examine how it can be deceived. This paper then examines the current state of research into deceiving AI via a literature review, focusing on adversarial examples and data poisoning attacks. This paper then postulates a threat model on how an adversary could leverage either type of attack in order to develop a method of comparing the relative strength of the two. These attacks are then conducted against simulated data.

Acknowledgements: This paper would not have been possible without the help of the Massachusetts Institute of Technology Lincoln Laboratory. Special thanks to Olivia Brown for helping me through this process and serving as my second reader. I would never have finished if I had not had my weekly meetings with Oliva, Jason Matterer, and Michael Yee, and would not have been able to handle the code or derivations without their help. I am also deeply thankful for the help and guidance of my Masters in Security Studies mentor, Doctor Brandon Valeriano. Thank you for believing that this was a reasonable, perhaps even interesting, topic to research and mentoring me through this process. I also benefitted from the careful eyes of Lieutenant Colonel Zachariah Anthony, Major Justin Frickie, Major Matthew Sanchez, and Andrew Everett. The paper would have been less well-written and less well-argued without your help.

Finally, I could not have done this without the encouragement, support, and help from my wife, Jessica Kellogg. If you can follow anything that I have written, it is because she told me on several occasions that what I had put on paper made little sense to anyone who was not me. I am a better writer because of your help on every major paper I have ever written, but I am still not good enough to write them without you.

Figures

<i>Figure 1 – Simulated Data with Five Clusters, Generated with Matplotlib.....</i>	<i>18</i>
<i>Figure 2 – Simulated Clusters with Three Un-Labeled Data Points.....</i>	<i>19</i>
<i>Figure 3 – Two-Class Simulated Data Problem Using Numpy and Matplotlib.....</i>	<i>20</i>
<i>Figure 4 – LDA applied to the Simulated Data in Figure 3.....</i>	<i>21</i>
<i>Figure 5 – SVM Applied to the Simulated Data in Figure 3.....</i>	<i>22</i>
<i>Figure 5 – SVM with Non-Linear Kernel Applied to Simulated Data in Figure 3.....</i>	<i>22</i>
<i>Figure 7 – Adversarial Example Attack Against SVM from Figure 4.</i>	<i>23</i>
<i>Figure 8 – Adversarial Example Applied to the SVM in Figure 5.....</i>	<i>24</i>
<i>Figure 6 - Histogram of Ratios Between Adversarial Examples and Equivalent Data Poisoning Attacks.....</i>	<i>34</i>
<i>Figure 7 – Scatter Plot of Ratio Versus Distance Between Means of Simulated Data- Clusters.....</i>	<i>34</i>
<i>Figure 11 – Accuracy of Poisoned LDA Classifiers.....</i>	<i>35</i>
<i>Figure 12 – Accuracy of Poisoned LDA Classifiers Compared with Distance Between Clusters.....</i>	<i>36</i>
<i>Figure 8 – Example of a Successful Adversarial Example Attack on an SVM.....</i>	<i>37</i>
<i>Figure 9 – Another Example of a Successful Adversarial Example Attack on an SVM....</i>	<i>38</i>

Table of Contents

	Page
Executive Summary.....	1
Abstract.....	2
List of Figures.....	3
Table of Contents.....	4
Introduction: Negative, suppression is VOID.....	5
AI and the DoD.....	7
The AI Arms Race: China, Russia, and ISIS.....	12
How Machines “Think”	17
Technical Background on Adversarial Examples and Data Poisoning Attacks.....	25
Assumed Threat Model for Analysis.....	31
Methodology.....	32
Findings.....	33
Analysis and Future Research.....	39
Conclusion.....	41
Notes.....	42
Appendix A: Python Code.....	49
Bibliography.....	57

Introduction: Negative, suppression is VOID

In 281, Pyrrhus of Epirus crossed the Adriatic Sea to give military aid to the Italian city of Tarentum against Rome. Pyrrhus brought with him a surprise weapon: 20 war elephants.¹ His Roman opponents were unprepared to face these incredible engines of war – as they would be half a century later against Hannibal Barca. It took 50 years of adaptation to deal with this new “technology.” Finally, Scipio Africanus trained his Roman to psychologically to face these battlefield giants, deploying his formations with set channels to allow elephants to pass between them and using musicians and war cries to scare these behemoths back into the Carthaginian formation, leading to victory at the Battle of Zama.²

A similar surprise awaits on the battlefield of the future in the form of fully autonomous weapons and Artificial Intelligence (AI) capabilities. These systems promise to drastically alter the character of warfare, transforming both tactics and strategy while changing the geopolitical calculus surrounding warfare itself. The US military must develop new techniques, tactics, and procedures to face these weapons before they surprise us on the battlefield.

Suppression is integral to modern military tactics. From the first introduction a Marine Officer receives to maneuver under fire during the Buddy Pair Fire and Maneuver Course at Officer Candidate School to the first battle drill in the Army’s field manual on platoon and squad tactics, suppressing fire is trained and ingrained as a tactical necessity.³ When confronted with autonomous weapons, modern Marines and soldiers will likely spend effort on trying to suppress them.

But autonomous weapons cannot be suppressed. They do not feel fear. They do not have a cardiovascular or central system, nor do they feel the concussive effects of explosions. Any drive for self-preservation is programmed, capable of being modified at any time. Any effect

fires will have on such weapons will only be through direct damage. Marines fighting on future battlefields against such weapons will have to leverage different tactics than the current use of suppression to cover maneuver. At its most basic, tactical level, fighting machines will be a different kind of war.

In removing humans from the battlefield, AI promises to transform the character of war beyond these tactical implications. Lieutenant General H.R. McMaster's formulation of war as "an uncertain contest of wills" raises the obvious question: how do you break the will of a machine? How do you deliver the Clausewitzian force compelling an enemy to do our will when that enemy never sleeps, never loses focus, and is completely dedicated to victory?⁴ War fought by machines will be like none other, as the ontology humans use to understand war, engagements, battles, campaigns, etc., is fundamentally predicated on human emotion and psychology. How does a battle cease if one side never breaks psychologically? Does a battlefield populated with machines constitute a change not only in the character of warfare, but in its very nature?

While warfare may have its own grammar, AI will likely fight as if speaking an entirely different language.⁵ Battlefield generalship would appear to require general-purpose AI beyond what is currently imagined, but only the most hubristic human-chauvinist would consider it fundamentally out of reach for machines. On the gameboard, Gary Kasperov's loss at Chess to Deep Blue and Lee Sedol's defeat at Go by Deep Mind's AlphaGo both demonstrate the ability of machines to defeat humans in areas previously thought impossible. AlphaGo's ability to choose super-human moves in Go is noteworthy in that autonomous weapons and AI command likely will present human opponents with tactics similarly perceived as incomprehensible by humans.⁶ Indeed, Deep Mind's latest endeavor in using AI to play the computer game Star Craft

demonstrates this effect.⁷ While AlphaStar has not yet proven it can defeat every opponent, the program has beat professional players with similar levels of impenetrable decision-making and superhuman play.⁸ Imagine combining the shock of defeat with the emotion of not only not understanding how it happened, but being incapable of ever understanding it.

The advantages of AI-enabled warfare are driving countries to adopt autonomous weapons, leading to a strategic AI arms race.⁹ While individuals such as the Secretary General of the United Nations, Elon Musk, Stephen Hawking, and machine learning conferences propose ethical AI requirements, these efforts have not stopped the development of these systems by both the United States and its adversaries.¹⁰

What happens to our conception of war when machines do the fighting? Marine Corps Doctrinal Publication 1, *Warfighting*, defines war as “a violent struggle between two hostile, independent, and irreconcilable wills.”¹¹ This definition breaks down when applied to a future where machines take on battlefield decision-making and autonomous weapons fight alongside humans. How does one defeat an enemy that is programmed to never give up? How can human effort outlast an enemy that never tires or sleeps? Can a person overcome an opponent devoid of rage, fear, and uncertainty? If one accepts Sun Tzu’s formulation that war is in essence deception, then the future of warfare requires its participants to master the art of deceiving machines.¹²

This paper proposes a new framework for analyzing the robustness of AI from the perspective of how to deceive machine learning algorithms. The AI-enabled “elephants” are coming; the military needs new techniques, tactics, and procedures to fight them.

AI and the DoD

The Department of Defense (DoD) likely has under invested in AI research given future battlefield impact of this technology. The Defense Advanced Research Projects Agency (DARPA) conducts the majority of this research, with the more recently established Joint Artificial Intelligence Center (JAIC) coordinating the DoD's efforts to implement AI.¹³ Individual services also are experimenting with AI technologies, demonstrating the possible future combat impact of the technology while also exposing resistance and difficulties in fielding these technologies. DoD strategy is beginning to cover the impact of AI, as shown in the 2018 National Defense Strategy, but does not address the degree to which this technology will influence the character of future warfare, particularly in combat between AI-equipped belligerents.¹⁴ These difficulties reflect both the incredible pace of change in AI research as well as cultural resistance to these changes.

DARPA's mission is to "make pivotal investments in breakthrough technologies for national security."¹⁵ Established in 1958 as a response to the launch of Sputnik by the Soviet Union, DARPA has developed numerous cutting-edge technologies, such as ARPANET (the precursor to the internet) and stealth technology.¹⁶ As such, DARPA has been an early explorer of the capabilities of AI systems. In 1966, DARPA and Stanford University developed an autonomous robotic system that could move.¹⁷ Called "Shakey the Robot" because of how it shook while it moved, this effort represented the first general purpose robot.¹⁸ Shakey spurred pioneering research on computer vision, natural language processing, and autonomous route planning.¹⁹ DARPA's research on autonomous systems reflects the pace and focus of AI research. For example, none of the entrants into the 2004 DARPA Grand Autonomous vehicles challenge completed the challenge, but in 2005, five entrants were successful.²⁰ This DARPA-centric approach to AI also reflects AI's position in the DoD as a breakthrough technology.

DARPA's research is focused on exploring the leading-edge of AI's capabilities, not its impact on warfare.

DARPA maintains a role in current DoD AI research. In 2015, DARPA presented its view of AI development as having three waves. The first wave of AI research, "handcrafted knowledge," consisted of systems where human engineers set the rules and machines explored their implications.²¹ DARPA defines the second wave of AI research as statistical learning, focused on finding underlying patterns and structures within data and using machines trained on these patterns to make determinations on new inputs. DARPA sees these systems as "statistically impressive, but individually unreliable."²² DARPA's current research focuses on their proposed "third wave" of AI research: systems capable of contextual adaptation.²³ In this area, DARPA is investing in "explainable AI," where the decisions of systems like neural networks (a "second wave" AI technology) can explain its decisions to a human supervisor.²⁴ Another effort of particular interest to this paper is DARPA's partnership with Intel Corporation and Georgia Institute of Technology on developing secure AI.²⁵ However, even this research still follows DARPA's mission to develop breakthrough technologies as opposed to understanding the impact those technologies will have on the future battlefield.

The DoD's initial effort to leverage modern AI technologies (second-wave technologies in DARPA's framework) in combat fell under the Algorithmic Warfare Cross-Functional Team (AWCFT). The AWCFT was established in 2017 to "integrate artificial intelligence and machine learning more effectively across operations to maintain advantages over increasingly capable adversaries and competitors." Project MAVEN (another name for the AWCFT) was a notable first step in integrating AI capabilities into combat operations. MAVEN first focused on using computer vision in order to assist in the Processing, Exploitation, and Dissemination (PED) of

full-motion video (FMV).²⁶ This project directly addressed the need brought on by the proliferation of battlefield sensors, with some FMV systems producing over 400 gigabytes of data per second.²⁷ As Lieutenant General David Deptula, Air Force Deputy Chief of Staff for Intelligence, Surveillance, and Reconnaissance, summarized, “We’re going to find ourselves in the not too distant future swimming in sensors and drowning in data.”²⁸ Project MAVEN foresaw many other areas of AI integration, such as document processing, natural language processing, persona identification, and optical character recognition.²⁹ However, this effort encountered obstacles. In particular, Google dropped out of Project MAVEN after employees protested their involvement in DoD projects.³⁰

JAIC’s establishment in 2018 is a reflection of the successes of Project MAVEN as well as the need for more investment and coordination. The 2018 National Defense Strategy stated that ongoing advances in AI would change “the character of war.” JAIC was established in order to speed delivery of AI technologies in response to this change.³¹ JAIC’s first National Mission Initiative (JIAI’s term for their premier research projects) was taking over Project MAVEN, and JAIC’s funding reflected a marked increase in the size of these AI-related contracts. JAIC’s early emphasis on ethics in AI implementation and research also reflected MAVEN’s problems with Google.³² In these respects JAIC represents a step forward from MAVEN in both effort and synchronization.

Unlike DARPA, JAIC exists to explore the implementation of AI into DoD operations. JAIC is the DoD’s “Center of Excellence” that provides a critical mass of expertise to help the Department harness the game-changing power of AI.”³³ Reflecting the rapid progress of AI research and the imperative to prevent adversaries from developing and fielding systems before the US can, the JAIC focuses on speed and agility. This speed and agility are necessary, as are

JAIC's efforts to unify AI efforts across the services.³⁴ Lost in recognition that AI will change the character of war and the necessity for accelerating the deployment of AI technologies is an understanding of what this change will look like and how these technologies will be used.

Experimentation in AI-involved combat has revealed the incredible capability afforded by such systems. DARPA's AlphaDogfight program sought to demonstrate the capability of AI systems in simulated air-to-air combat over the course of 2020. This experiment consisted of three events, pitting teams of AI programmers against each other with a final challenge against a human pilot. The winning team's system demonstrates the degree to which AI systems will likely change the character of war. The AI outmatched the human pilot, winning all five of the simulated dogfights.³⁵ The way the winning team won was also informative: The Heron Systems Falco AI, the challenge winner, performed maneuvers that were far more aggressive than would be considered by a human pilot, such as head-on approaches, and had near-perfect accuracy in its gunnery.³⁶ This level of aggression demonstrates how AI changes the character of warfare in that autonomous systems will not fight like humans, while the perfect aiming shows the same super-human mastery demonstrated by Deep Blue and AlphaGo. These systems will only improve in their capabilities, likely resulting in them being fielded before most service members are ready.

AI as a technology presents what author Clayton Christensen calls "the innovator's dilemma."³⁷ Although AI-enabled technologies currently under-perform cutting edge weapons systems, dissuading early investment, these technologies will over-perform in the future. Like the hard drive manufacturers in Christensen's book, DoD leadership and Congress have been incentivized to invest in technologies that are currently most capable instead of future prospective dominance. For example, Congress denied the U.S. Navy's request for funding for un-crewed vessels, demanding that the Navy work out every component of the vessel before

moving forward with the acquisitions plan.³⁸ Leaders are also unlikely to invest in technologies that threaten human primacy in their communities. This is demonstrated by resistance to technologies that threatened manned programs, such as the Unmanned Combat Aerial Vehicle and correspondingly the F-35.³⁹ Another challenge is that the DoD workforce lacks the technical acumen necessary to understand the impact of AI. The Marine Corps' Manned-Unmanned Teaming program lead at the Marine Corps Warfighting Lab, Colonel J. Darren Duke claims, "The reason for this narrow scope and glacial pace of exploitation [of AI technology] is data-illiteracy."⁴⁰

While the DoD is accelerating its AI research and development efforts, these likely remain undersized when compared to the changes that AI will bring to the character of future warfare. Despite these obstacles, the DoD must not only continue its efforts, like the JAIC, to field these technologies but also to understand the impact they will have once fielded by the United States' adversaries. Increased funding for JAIC and increased emphasis on AI programs are necessary to ensure that the United States does not suffer capability surprise from future AI weapons systems.

The AI Arms Race: China, Russia, and ISIS

China and Russia are investing in AI technologies in different applications, but both see AI as the future of warfare and an opportunity to reshape the global balance of power. The Chinese Communist Party (CCP) sees AI as on par with the industrial revolution, presenting a key opportunity to become a world economic and military leader and is investing strategically in order to gain economic and military advantage.⁴¹ Russia is focusing on developing autonomous systems for combat and has experience deploying un-crewed ground vehicles in combat operations in Syria.⁴² The rapid pace of AI development also poses a proliferation risk, as

terrorist groups could re-purpose Commercial Off the Shelf (COTS) AI solutions as battlefield weapons. The breadth and scope of adversary interest and investment in leveraging AI in warfare will almost certainly lead to the proliferation of such technology on the future battlefield.

The Chinese Communist Party sees AI as revolutionary technology that will allow China to surpass the United States in scientific, economic, and military dominance. The President of the People's Republic of China, Xi Jinping, has stressed the importance of AI, stating that it is “a strategic technology heralding this round of scientific and industrial revolution and industrial change.”⁴³ The CCP sees the AI revolution as an opportunity for “leapfrog development,” where lagging countries can skip a development stage to overcome more developed competitors.⁴⁴ This assessment parallels Christensen's analysis: China's lack of competitiveness in current-generation technologies represents an advantage in their ability to adopt AI.⁴⁵ China seeks the “first mover” advantage in AI investment not only to accelerate its development but also as a means for great power competition. The CCP's 2017 *New Generation Artificial Intelligence Development Plan* states that “AI has become a new focus of international competition.”⁴⁶ China's investment in AI across economic and national security applications has grown accordingly.

The CCP is systematically investing in AI with a unified strategy in order to become a global leader in AI technologies. The 2017 *New Generation Artificial Intelligence Development Plan* set goals of “achieving important progress” in AI by 2020, “major breakthroughs” by 2025, and become a world leader by 2030.⁴⁷ China's AI investments have followed this strategy, with the CCP making over 70% of global investments in AI in 2017, later surpassing the US to become the top nation state investor in AI in 2018.⁴⁸ Similar to DARPA and JAIC, China has established two major research organizations—some of the largest government research agencies

in the world with each employing more than 100 researchers.⁴⁹ The CCP has also pushed for increases in AI research and education at Chinese universities.⁵⁰ This strategic approach has placed China as the world leader in AI papers, patents, and venture capital investment, second only to the United States in terms of the number of AI companies and the size of its AI talent pool.⁵¹

The People's Liberation Army (PLA) benefits from all facets of Chinese AI research through Military-Civil Fusion (MCF). MCF is a two-way transfer of knowledge, with the PLA able to leverage research done at any Chinese company or institution while also supporting such research through funding and espionage.⁵² MCF seeks to create and leverage these synergies in order to achieve both the national security and economic transformational goals of its AI strategy.⁵³ Chinese government dominance over AI research prevents Chinese companies from backing out of military partnerships, in contrast to Google's decision to end participation in Project MAVEN, and allows the CCP to hide the degree to which their domestic AI industry is focused on military development.

The PLA seeks to adopt AI technologies to achieve parity with the US military. The PLA increasingly refer to "intelligent" or "intelligentized" technology as the basis for future war.⁵⁴ Zeng Yi, an executive at the Chinese defense company NORINCO, expressed their view on autonomous weapon systems, stating, "In future battlegrounds, there will be no people fighting," and that the military use of AI is "inevitable."⁵⁵ The People's Liberation Army also sees human cognition as a limiting factor in future conflict and is leveraging AI in order to augment decision-making: "Only relying on the command experience of individuals, and adopting simple, direct, qualitative decision methods to form the operational resolution, are by far no longer able to adapt."⁵⁶ Zeng's opinions reflect this as well, stating that "AI may completely change the current

command structure, which is dominated by humans” to one that is dominated by an “AI cluster.”⁵⁷ The PLA is also investing in AI-enabled “assassin’s mace” technologies, such as autonomous mini-submarines to threaten aircraft carriers and drone swarms capable of taking out high value air defense targets.⁵⁸ The PLA and Chinese defense industry’s enthusiasm for AI-enabled battlefield technology increases the likelihood of near-term employment of AI on the battlefield.

The Russian Federation also sees AI as the key to future global competition. In an address to Russian schoolchildren in 2017, President Vladimir Putin stated, “Whoever becomes the leader in this [AI] sphere will become the ruler of the world.”⁵⁹ Russian Defense Minister Sergei Shogyu ordered production of un-crewed ground vehicles in 2018 and stated that Russia’s military must introduce intelligent weapons systems.⁶⁰ The Russian defense industrial base and military theorists agree. General Valery Gerasimov, the Chief of the General Staff of the Russian Armed Forces stated that in his view, the main features of future conflicts will be precision weapons and robotics.⁶¹ While Russia lacks the economic strength of the US or China and cannot match either’s investment, Russia has strategically invested in civilian-military partnerships, including an AI and semantic data analysis research project that represents one of the largest in Russia to date. Russia has even launched its own DARPA-like organization, the Foundation for Advanced Studies, to research military robotics applications.⁶² Like most of Russia’s defense expenditures, AI represents an avenue to find asymmetric advantage against the US military, allowing Russia to target specific defense projects to achieve outsized advantage when compared to the relative size of its economy. While Russia’s AI programs are smaller, they have led to meaningful experimentation and could easily produce the weapons systems envisioned by Russian leadership.

Russia has led in battlefield experimentation with un-crewed ground vehicles. Russia has used the conflicts in Syria, Ukraine, and Libya as opportunities to gain battlefield experience with its newest weapons systems, and has leveraged this opportunity to experiment with the tactical employment of its latest weapons systems. Russia has experimented with the Uran-6, Uran-9, Scarab, and Sphera un-crewed vehicles in Syria and Ukraine, with the Uran-6 and Sphera rated highly for their performance.⁶³ While the Uran-9 was rated poorly, this sort of experimentation is priceless in developing techniques, tactics, and procedures for the tactical integration of such systems. Though Russian military theorists emphasize their desire to maintain a “human-in-the-loop” for such systems, developing a wide variety of un-crewed systems means that the Russian military is poised to integrate AI-enabled autonomy, were such technology to be developed.

Though non-state actors likely will not possess the ability to develop AI-enabled weapons systems, they likely will be able to integrate COTS AI-solutions into improvised weapons. The Islamic State of Iraq and Syria (ISIS), as well as other Syrian militia groups, was particularly adept at using commercial drones for surveillance and as improvised weapons. ISIS deployed these drones in swarms, sometimes rigged to drop 40mm grenades.⁶⁴ In one video posted by the group, a hovering drone perfectly hits the open hatch of a stationary M1 Abrams tank.⁶⁵ AI technology likely will be no different, with terrorist groups re-purposing existing systems towards creating more lethal and harder to counter threats. As AI permeates everyday technology through the internet of things, it likely will present a proliferation threat for non-state adoption.⁶⁶ Improvised explosive devices capable of visually identifying their targets and small unmanned air systems capable of navigating without outside control and navigation likely would negate much of the technology currently employed to defend against such systems. At the current pace

of research and development, battlefield-applicable AI likely will not remain the sole purview of nation states.

The United States' adversaries are poised to develop and adopt AI weapons systems. Though these will initially underperform the state of the art in military technology, it is likely that such technology will rapidly transform the character of war, changing who fights warfare and how they fight. The US military must begin thinking through ways to defeat such systems before they appear, rather than suffer capability surprise and possible defeat when these systems are fielded on the other side.

How Machines “Think”

Fighting AI requires an understanding of what AI is and how it works. AI presents a battlefield challenge not only through its ability to demonstrate super-human skill in certain aspects of decision-making, but also in the fact that this ability often results from processes distinct from human cognition. As such, users need to understand how these systems work instead of projecting human thought processes on to them.

There is no agreed upon definition for AI. The diversity in research approaches and topics covered under AI means that there is no commonly accepted definition of AI in the field. There is also no official US government definition of AI.⁶⁷ The 2018 DoD Strategy on AI defines it as: “the ability of machines to perform tasks that normally require human intelligence—for example, recognizing patterns, learning from experience, drawing conclusions, making predictions, or taking action—whether digitally or as the smart software behind autonomous physical systems.”⁶⁸ This definition is telling: definitions of AI frame it in terms of human intelligence. Humans deem intelligence upon machines that accomplish tasks without human

supervision that would normally require human intelligence. This fallacy is dangerous, as the assumption that a machine is thinking like a human exposes the assumer to deception.

Machine Learning (ML) is a subset of AI research where algorithms make decisions based on rules inferred from pre-existing training data. In this respect, ML emulates human learning. ML is a major component of the latest phase of AI research, which DARPA would classify as “the second wave.”⁶⁹ ML uses a training set, which consists of data that is labeled—or marked with the correct decision—to develop rules to classify data that is unlabeled. The graph below shows five classes of data, labeled with their respective color:

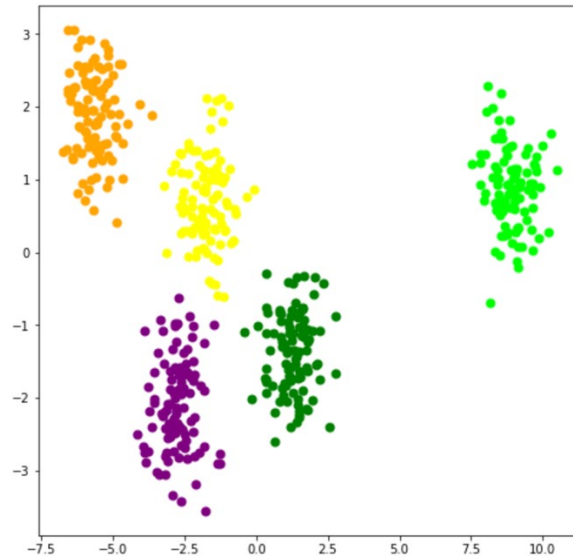


Figure 10 – Simulated Data with Five Clusters, Generated with Matplotlib.

ML is about inferring data labels from patterns in the training data. In other words, given the labels applied to the data in this training set, how does one label new data, such as the circled points below?

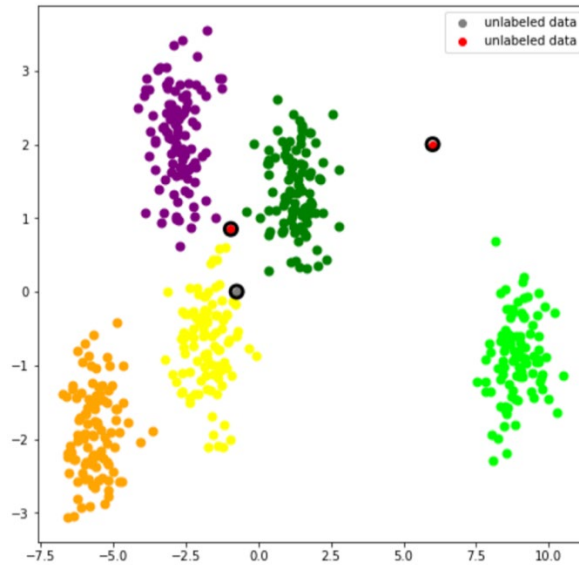


Figure 11 – Simulated Clusters with Three Un-Labeled Data Points.

For the grey point, the answer is intuitive: yellow. Problems arise from examples like the two circled red points: one is not close to any example to be labeled with certainty, while the other is equally close to several colors (purple, green, yellow) which could be suitable labels. While this seems simple, in a real-world application, each dimension could be a pixel in an image or a gene in a DNA sequence and the label as something like: “Tank,” “Civilian,” or “Cancer.” These examples demonstrate how relatively simple algorithms make choices in real world applications.

There are many ways to infer new labels in ML. A simple method is picking the closest point in the training set (or a collection of closest points, numbered k) and assuming that the unlabeled data shares class membership with that point (or points). This method is known as k -Nearest Neighbor (k -NN for short).⁷⁰ Another method is to find functions of the underlying classified data values that produce an overall score for each data point and then comparing those values to a threshold to determine class membership. This can be thought of as finding boundaries that separate each class. If the functions are purely linear combinations of the data

values, those boundaries are lines or planes in the data space. This paper will focus on two such methods: Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM).⁷¹

ML problems are most easily understood in the case where there are only two classes. For example, determining whether an image of an animal is a cat or a dog would be a two-class problem. Consider the example below:

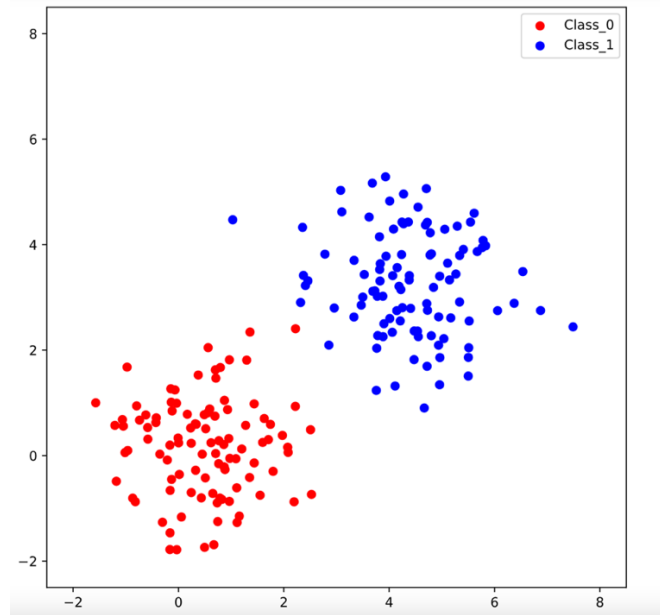


Figure 12 – Two-Class Simulated Data Problem Using Numpy and Matplotlib.

In this case there are two classes (Class_0 and Class_1) occupying a two-dimensional data space. An LDA classifier separates these classes based on their statistics (their mean and covariance) to find the plane across which the separation is most statistically clear, such as the separating line below:

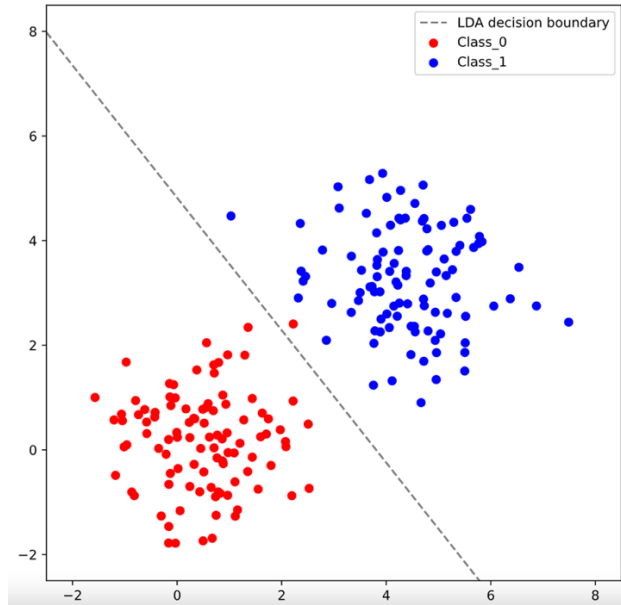


Figure 13 – LDA applied to the Simulated Data in Figure 3.

The algorithm then classifies unknown data based on which side of the line the new data falls in the data space.

In contrast, for a SVM, the goal is to find the line that provides the maximum spatial separation between these clusters (aside from a minimal number of errors in this case represented by the two dashed lines).

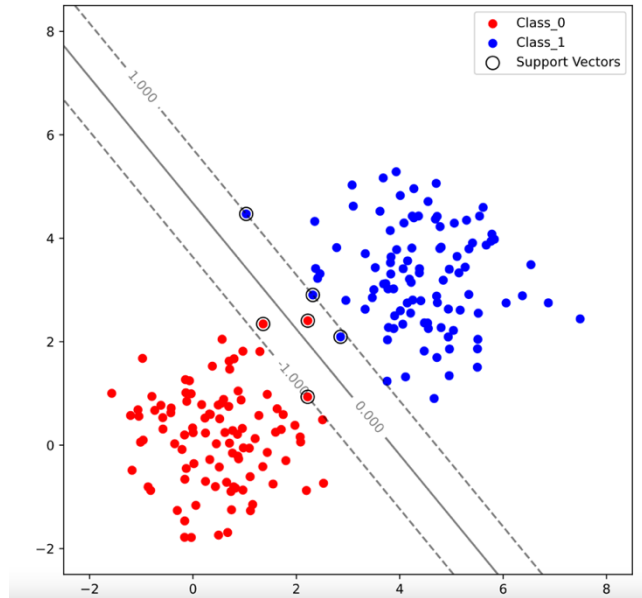


Figure 14 – SVM Applied to the Simulated Data in Figure 3.

An SVM classifies each class into either 1 or -1, and seeks to maximize the margin between points classified as such. Here the circled points are the “Support Vectors,” lying either directly on the maximum spatial separation line or between -1 and 1 from the maximum spatial separation line. An SVM can classify unknown data mathematically from these points (via the dot product), which allows it to be extended through transformations of the data space, like so:

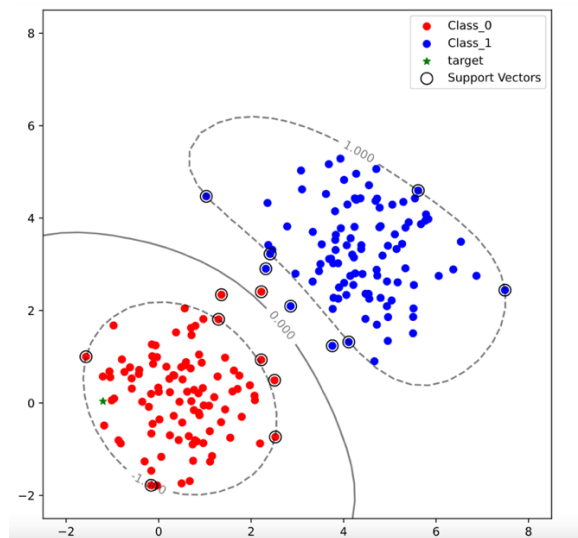


Figure 15 – SVM with Non-Linear Kernel Applied to Simulated Data in Figure 3.

This property allows SVMs to be used in cases where the data is not linearly separable. This transformability makes the SVM a versatile classifier for tasks such as handwriting recognition and identifying the classification of genes.⁷²

Given this understanding of how ML works, how can humans deceive these types of algorithms into miss-classified data? One solution is to perturb data from one class such that the algorithm incorrectly classifies it as another class. For example, the image below demonstrates the deception of an LDA by moving a point from Class_0 and making it appear to the classifier as if it were Class_1:

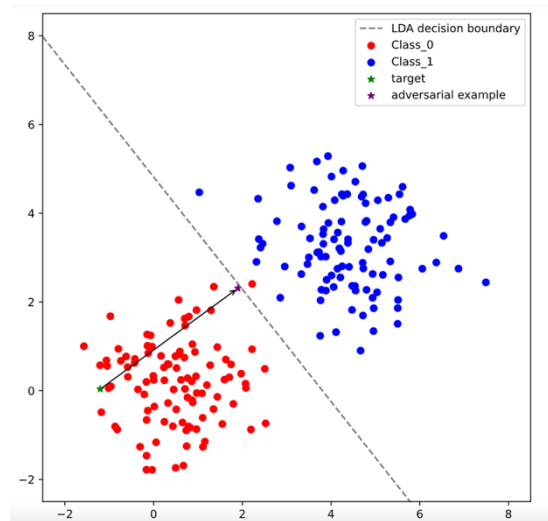


Figure 16 – Adversarial Example Attack Against SVM from Figure 4.

Here the target data (the green star) is manipulated such that it falls on the LDA classifying line (purple star) as an “adversarial example.” This technique works equally well for SVMs:

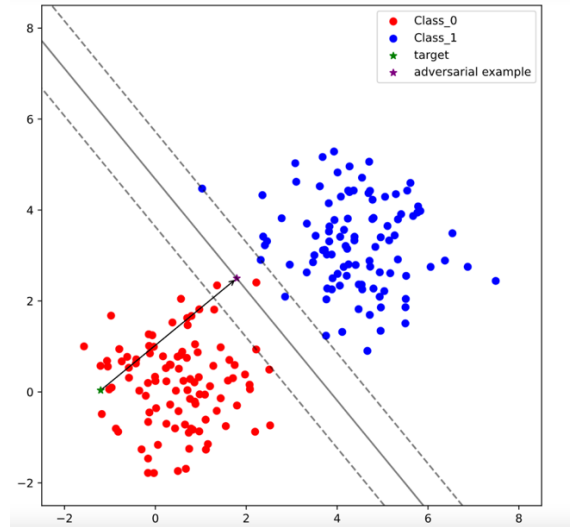


Figure 17 – Adversarial Example Applied to the SVM in Figure 5.

Another solution is to modify the training data in order to shift the decision boundary. These are called “data-poisoning” attacks which focus on shifting a point or group of points in order to make the classifier misclassify un-modified data.

While these attacks seem extreme in the two-dimensional case shown above, most machine learning problems have higher-dimensional data-spaces. For example, while humans see handwriting as a 2-dimensional problem, handwriting recognition algorithms see it as a 784-dimensional problem.⁷³ As the number of dimensions grow, the concept of distance changes. Points are geometrically farther away from each other in these high-dimensional spaces. For example, to cover 10% of the volume of a cube in 10 dimensions requires a cube having 80% of the length of the original. This “Curse of Dimensionality” allows these adversarial attacks and data poisoning attacks to visually appear as noise to humans, while still being effective against machines.⁷⁴

Understanding how AI and ML works illuminates the kinds of vulnerabilities these algorithms have. Though the above cases and the ones considered in this paper are simple, they demonstrate how humans could deceive AI on the battlefield. Research in the area of adversarial

examples and data poisoning attacks has military relevance as the US and its adversaries rush to deploy these systems.

Technical Background on Adversarial Examples and Data Poisoning Attacks

In *Can Machine Learning Be Secure*, Marco Barreno et al. (2006) proposed a framework for understanding the security of machine learning and AI. Their attack model defines three relevant properties for analyzing attacks on machine learning systems: influence, specificity, and security violation.⁷⁵ The influence of the attack can be causative or exploratory depending on whether the attacker alters the machine learning algorithm by controlling the training data or training process or exploits the algorithm through offline analysis or queries of the trained learning system.⁷⁶ This framework has been recently extended to include a third category, attacks that are exploitative. These so-called “inversion attacks” seek to recover sensitive information from the training set based on the behavior of the trained algorithm.⁷⁷ The specificity of an attack refers to whether the attack is targeted, in that it alters the output of the algorithm in a specified way, or indiscriminate, in that the attack evades proper classification.⁷⁸ Finally, the security violation property of an attack is whether the attack allows an attacker special privilege in evading proper classification from the system in the case of an attack on the integrity of the system, or the attack makes the algorithm unusable in an attack on the availability of the system. Research into these types of attacks has typically been studied independently and there is a need to both compare their relative strength as well as investigate combining their effects.

The most likely available attack surface of a machine learning algorithm is its input, and significant research exists into the viability of such attacks. Battista Biggio et al. in their paper *Evasion Attacks Against Machine Learning at Test Time* (2013), demonstrated a gradient descent

attack, making repeated small changes to the input along the path maximizing the overall error, resulting in an input that evades proper classification by a trained algorithm.⁷⁹ In the paper, *Intriguing properties of neural networks*, Christian Szegedy et al. (2014) demonstrated that discontinuities in the input-output mappings for neural networks allowed an attacker to force a network to misclassify an image with a visually imperceptible perturbation. This paper coined the term “adversarial examples” to describe these non-random attacks on neural networks.⁸⁰ Early explanations for adversarial examples conjectured that they took advantage of the non-linearity in machine learning algorithms, but in *Explaining and Harnessing Adversarial Examples* (2015), Ian Goodfellow et al. hypothesized that adversarial examples are the result of both the linearity of such algorithms and the high dimensionality of the data.⁸¹ Another hypothesis in Andrew Ilyas et al. *Adversarial Examples Are Not Bugs, They Are Features* (2019), is that adversarial examples are the result of spurious correlations in the training data, and that optimizing for accuracy emphasizes their effect.⁸²

The adversarial examples studies by Biggio, Szegedy, and Goodfellow required knowledge of the internals of the target algorithm or the data set used to train it. These “white box” attacks, while alarming, still required extensive knowledge on the attacking side for success. Nicolas Papernot et al. first demonstrated the ability to generate adversarial examples that function against “black box” deep neural networks, such that the attacker does not have access to the details of the machine learning algorithm, just its performance. Papernot, McDaniel, and Goodfellow demonstrated transferability in such black box attacks in their paper, *Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples* (2016), finding that adversarial examples generated against one machine learning architecture are often effective against other architectures.⁸³ The ability to create synthetic

training sets to perform a black box attack on a machine learning algorithm along with the transferability of such attacks proves their applicability given that an attacker can create adversarial examples without detailed knowledge.

Early research on defenses against adversarial examples focused on hiding the gradient values used by Biggio (2013) from queries to target algorithm, adding adversarial examples to training sets, or detecting them via a separate algorithm.⁸⁴ However, each of these defensive strategies has not been able to prevent attackers from creating effective adversarial examples. In *Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples*, Anish Athalye et al. (2018) demonstrate that defensive techniques focused on hiding or manipulating the gradient values of a machine learning algorithm are not effective in preventing an attacker from creating strong adversarial examples against the algorithm.⁸⁵ Adding adversarial examples to an algorithm's training set, called adversarial training, was suggested by Szegedy et al. (2014) in the original paper on the topic.⁸⁶ Aleksander Madry et al. formalized this training as a min-max optimization problem, solving for the optimal mix of adversarial examples in the training set by gradient descent.⁸⁷ Alexey Kurakin et al. in their paper, *Adversarial Machine Learning at Scale* (2016), demonstrated the effectiveness of this technique in defeating simple adversarial examples.⁸⁸ However, Florian Tramèr et al. in *Ensemble Adversarial Training: Attacks and Defenses* (2018), show that this technique can be defeated with more complex adversarial examples.⁸⁹ Another paper on this topic, *Robustness May Be at Odds with Accuracy* by Dimitris Tsipiras et al. (2019) shows that adversarial training lowers the overall accuracy of the algorithm, and that these adversarial trained algorithms are still vulnerable.⁹⁰ A third avenue for hardening machine learning algorithms is through trying to detect adversarial examples before they are input into the machine learning algorithm. In *Adversarial Examples*

Are Not Easily Detected: Bypassing Ten Detection Methods, Nicholas Carlini and David Wagner (2017) demonstrate that these detection methods are also ineffective.⁹¹ Finally, in *Adversarial Example Defenses: Ensembles of Weak Defenses are not Strong*, Warren He et al. (2017) demonstrate that not only are individual defensive techniques against adversarial examples ineffective, but that attackers can create effective adversarial examples against collections of defenses constructed from weak defensive techniques. This paper also demonstrated that evasion techniques against individual defenses show transferability across other defensive techniques and that the minimal perturbation needed to create an adversarial example is nearly as small as what is required to bypass the strongest detector.⁹²

Modifying training data presents another avenue for an attacker to subvert machine learning algorithms. In *Poisoning Attacks against Support Vector Machines*, Battista Biggio et al. (2012) demonstrate the ability for a single malicious data entry to cause machine learning algorithms, in this case SVMs, to miss-classify test samples indiscriminately, a low-specificity attack.⁹³ Huang Xiao et al. (2015) showed that similar data poisoning attacks were possible against linear regression techniques, such as Least Absolute Shrinkage and Selection Operator (LASSO) regression and ridge regression in *Is Feature Selection Secure against Training Data Poisoning?*⁹⁴ In *Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners*, Shike Mei and Xiaojin Zhu (2015) demonstrated data poisoning attacks against logistic regression. In each of these examples, the algorithm in question is linear and the attack was against data with only two categories of classification.⁹⁵ Luis Muñoz-González et al. in *Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization* (2017), demonstrated data poisoning both against a multi-classification problem and against non-linear algorithms, most importantly neural networks.⁹⁶ These papers demonstrate data-poisoning as an

effective technique in theory, but as a practical attack vector, the inputs were not constrained to be tagged by a human observer, a critical component of practical real world attacks.

Translating theoretical data-poisoning attacks into practical examples requires some method to add the poisoned data to the training set. One such method is making the data appear natural to a human observer. Initial attempts at such attacks focused on creating a so-called “backdoor” by adding a feature to a subset of the training data, then using that feature to mislead the trained algorithm on specific data points, thus conducting a targeted attack on the algorithm. In *Trojaning Attack on Neural Networks*, Yinqi Liu et al. (2017) showed how a model could be retrained with a specific trigger, creating a backdoor for an attacker. Retrained algorithms are fooled when they are presented with images containing the trigger when they would otherwise properly classify the input.⁹⁷ These attack vectors are considered “clean label” attacks, as a human observer will correctly classify the image and the attack works with the data properly labeled in the training set. In *Clean-Label Backdoor Attacks*, Alexander Turner et al. (2018) refined this technique, using a small black and white backdoor feature in the corner of human recognizable images.⁹⁸ Further research showed that these attacks are possible without the backdoor feature. In *Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks*, Ali Shafahi et al. (2018) create clean label examples leveraging “feature collision” within the overall data set in a targeted attack that caused the trained algorithm to misclassify specific inputs.⁹⁹ This feature collision used human-invisible features within the training set as their own sort of backdoor attack, causing the algorithm to misclassify inputs. These clean label attacks demonstrate an attacker’s ability to inject their poisoned data into the training pool in a way that would not be detected by a human observer.

Data-poisoning attacks demonstrate much of the same transferability as adversarial examples. In *Transferable Clean-Label Poisoning Attacks on Deep Neural Nets*, Chen Zhu et al. (2019) extended the techniques from Shafahi (2019) to an environment without attacker access to the underlying machine learning algorithm.¹⁰⁰ This attack uses a “convex polytope” to surround a portion of the data set with poisoned data, causing the attacked algorithm to misclassify data falling within the region bounded by this poisoned data. In *MetaPoison: Practical General-purpose Clean-label Data Poisoning*, W. Ronny Huang et al. (2020) developed a robust method to create tailored clean-label, black box, transferable data poisoning attacks. This technique frames the data-poisoning attack generation as a bi-level optimization problem, similar to methods used to create adversarial examples.¹⁰¹ This research shows that data-poisoning likely has the same advantages in black-box generation and transferability as adversary examples.

As in the case of adversarial examples, the discovery of practical data-poisoning attacks has driven research into defensive techniques against these attacks. Research predating the discovery of transferable data-poisoning focused on cleaning training sets. In *Casting out Demons: Sanitizing Training Data for Anomaly Sensors*, Gabriela F. Cretu et al. (2008) used a data-sanitization step to try to remove erroneously labeled or malicious data through examination of small subsets of training data for network intrusion detection systems.¹⁰² Directly addressing the threat of data poisoning attacks, Jacob Steinhardt et al. demonstrated how an oracle, or perfect outside database, with knowledge of the true statistics of the un-poisoned data can limit the effectiveness of such attacks against SVMs in *Certified Defenses for Data Poisoning Attacks* (2017).¹⁰³ In *Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering*, Bryant Chen et al. (2018) showed that backdoor attacks could be detected by looking for

clustering in the activation of nodes while training the neural network.¹⁰⁴ In defending against the targeted clean-label attacks outlined by Shafahi (2019) or the convex polytope attacks demonstrated by Zhu (2019), Peri et al. showed that using a K- Nearest Neighbor (kNN) algorithm to pre-filter training data removed poisoned data in *Deep k-NN Defense Against Clean-label Data Poisoning Attacks* (2020).¹⁰⁵

Assumed Threat Model for Analysis

Comparing adversarial examples to data poisoning attacks requires a threat model where the attacker can leverage both methods – otherwise comparison makes little sense. This model also must introduce some form of a trade-off between both attacks. The threat model also should demonstrate whether the attacker is performing a white box attack—where the attacker knows details of the defender’s machine learning algorithm—or a black box attack—where the attacker has no such knowledge.

For this analysis, the defender is constructing a machine learning algorithm to properly classify inputs by training a machine on properly tagged data. This input/initial data is aggregated from web-scraping or other open sources. This machine learning algorithm is then deployed in an operational environment to classify inputs from an external environment. For example, this could be an algorithm designed to identify and classify military vehicles, similar to a French proposal for machine learning.¹⁰⁶ In this proposal, the training data set is open-source images of military vehicles which are properly tagged, and the deployment use-case is deploying the algorithm to identify military vehicles on the battlefield.

The attacker for this analysis seeks to subvert the defender’s efforts to properly classify inputs. They can modify data in the defender’s training library—perturbing a properly tagged datum—or modify initial inputs to the defender’s trained algorithm. The attacker achieves

success if the defender's algorithm misclassifies a target element through either perturbing the input or changing the algorithm's ability to classify it. For the French proposal above, the attacker could seed the open-source data environment with properly tagged images of military vehicles that have been modified specifically to target the vehicle recognition algorithm or the attacker could modify the camouflage or silhouette of their deployed vehicles in order to deceive the trained system. For the purposes of this paper, the attacks will be white box attacks where the attacker knows the details of both the defender's training set and machine learning algorithm. These assumptions are not unreasonable, given that many machine learning companies publish enough details regarding their methodology that an attacker reasonably could be certain which algorithms they are attacking.¹⁰⁷ Additionally, the defender's use-case will allow the attacker to be able to determine what type of data is being used to train the algorithm. Assuming perfect knowledge will simplify the following analysis without losing generalizability.

Though adversarial examples and data-poisoning attacks share commonalities, little research has been done comparing these two techniques. Preliminary research into how different defensive techniques perform indicates that hardening an algorithm against one form of attack may make it more susceptible to others.¹⁰⁸ Combining data-poisoning and adversarial example approaches treats the entire life-cycle of the machine learning algorithm as an attack surface. As data-poisoning and adversarial examples use the same definitions of closeness, these measures could be used as a way to measure the trade-off between each method of attack, finding the best technique for a given Euclidean distance. If simultaneously solving two bi-level optimization problems is feasible, this method of comparison could find mixed approaches outperforming either technique alone.

Methodology

This paper analyzes the LDA from *Elements of Statistical Learning* in order to derive a single-point adversarial example for the classifier as well as to determine a method for generating a single-point data poisoning attack for the same classification method.¹⁰⁹ The same was attempted with an SVM using a linear kernel, via the method outlined in *Poisoning Attacks against Support Vector Machines*, by Battista Biggio et al. (2012).¹¹⁰

These methods were applied to simulated data sets drawn from 2-dimensional Gaussian distributions. These methods were chosen in order to allow both the LDA and SVM to be applicable to the ML data space. All analyses were completed using the Python coding language in the Jupyter Notebook web-based interactive environment.¹¹¹ Numerical calculations were processed using the Numpy package for Python, and the SciKit Learn SVM was used for all SVM-related research.¹¹² The Python code for this research is included as Appendix A.

Findings

For the LDA, single-point poisoning attacks were far larger than the corresponding adversarial examples. Over 10,000 trials covering a variety of distances between clusters, the average data-poisoning attack was 164.9 times larger than an adversarial example for the same target data, with a standard deviation of 29.9 for this ratio. This shows that data poisoning attacks against LDAs are far larger than adversarial examples.

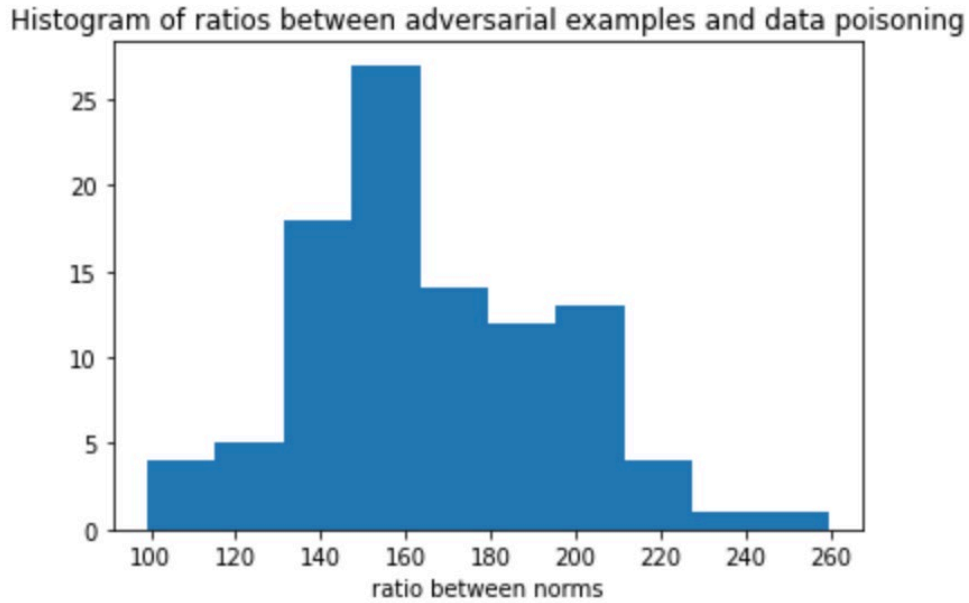


Figure 18 - Histogram of Ratios Between Adversarial Examples and Equivalent Data Poisoning Attacks

The variation in this ratio was not correlated to the distance between the two clusters, and was likely dependent on the random selection of a target data point in each test.

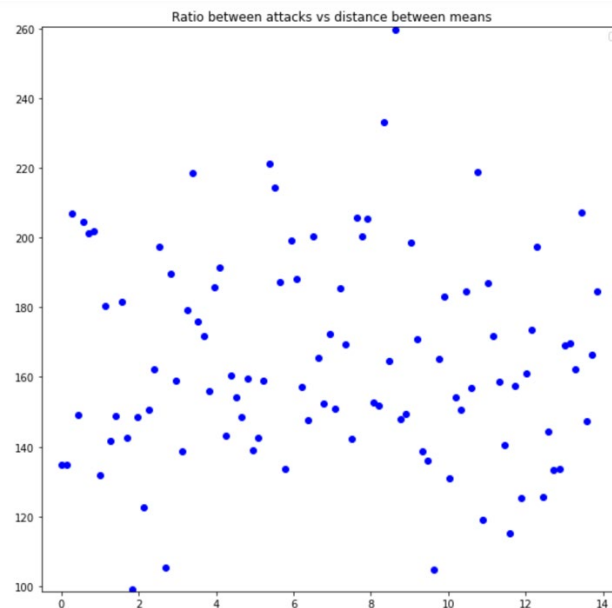


Figure 19 – Scatter Plot of Ratio Versus Distance Between Means of Simulated Data-Clusters

The code for this paper measured accuracy by comparing the poisoned and un-poisoned LDAs with data drawn from the same distribution as the training set. The accuracy dropped

across all poisoned LDAs. The average poisoned LDA was only 77.5% as accurate with a standard deviation of 2.3% after being attacked as it was prior to it, as shown by the ratio displayed below:

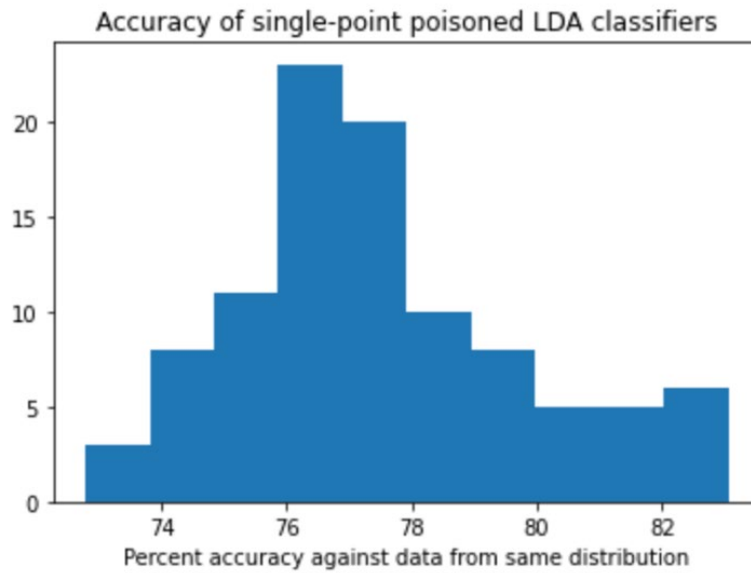


Figure 20 – Accuracy of Poisoned LDA Classifiers

This drop in accuracy had less variation than the ratios, but this drop in accuracy did not appear to vary as the distance between the clusters increased.

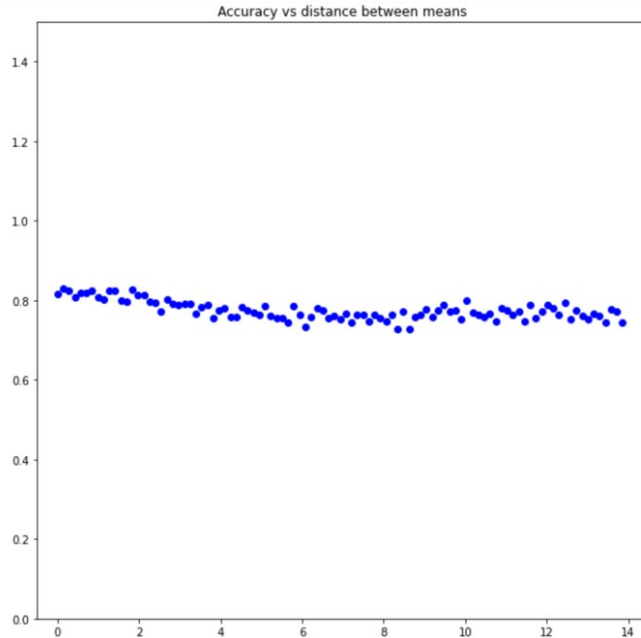


Figure 21 – Accuracy of Poisoned LDA Classifiers Compared with Distance Between Clusters

Comparing single-point data poisoning to adversarial attacks with SVMs proved more challenging. While the LDA attacks worked across any point with a target selected at random, the SVM version was not as accurate. However, some of the results where the code created a successful attack were informative regarding how data poisoning works against SVMs. For example, in the picture below the target’s location within the margin means that the attack could easily succeed (in this case the data poisoning was a point identical to the target with the opposite class tag):

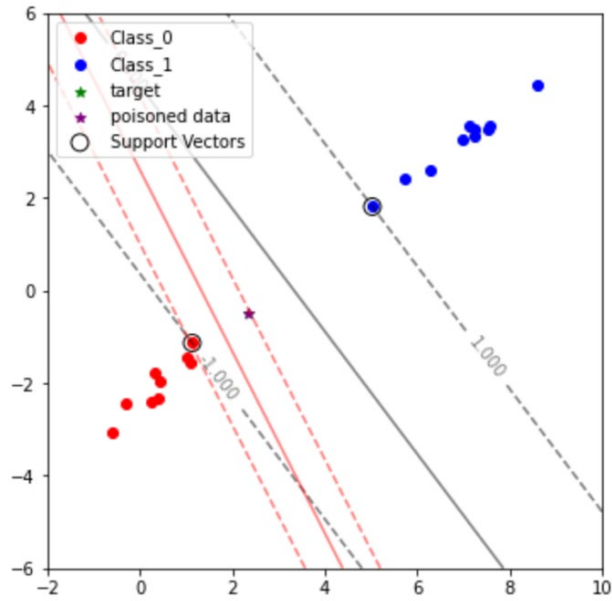


Figure 22 – Example of a Successful Adversarial Example Attack on an SVM

In other examples where the target was outside the SVM's margin region, the gradient descent algorithm (following the path of greatest change locally) managed to change the maximum separating hyperplane drastically, such as in the picture below:

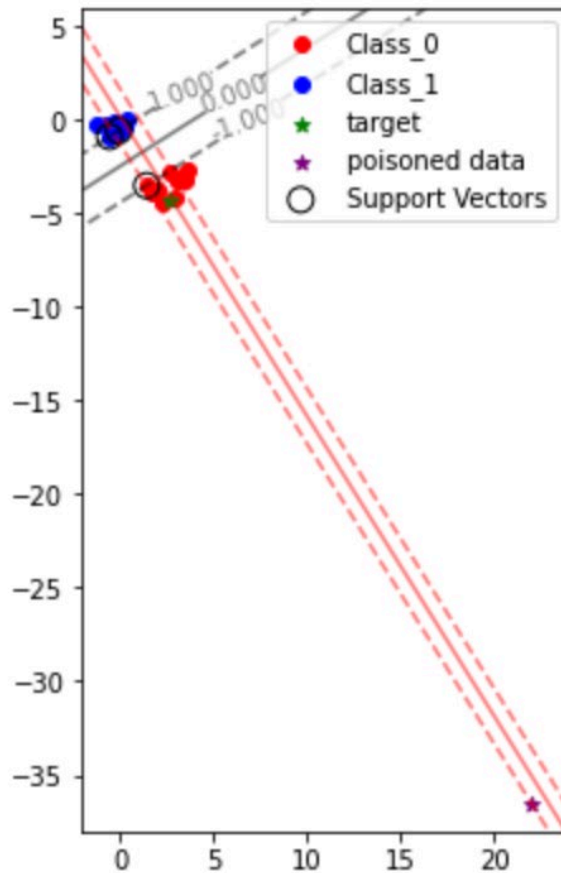


Figure 23 – Another Example of a Successful Adversarial Example Attack on an SVM

It is likely that a better comparison between attacks is possible with an improved algorithm, better coding, and a more elegant target selection method.

Though exact mathematical derivations of minimal attacks are possible on paper, they are less robust in application on a computer. Often these attacks had to be scaled by small amounts (1-2%) in order to adjust for underlying rounding and approximation done by the computer. In particular, there was a considerable disadvantage for the attacker with “grey box” knowledge of the distribution from which the simulated data was taken versus an attacker who had perfect access to actual data used to train the algorithm. Small approximations or simplifications in deriving the data poisoning attack also changed the results in ways that were counter-intuitive. For example, disregarding higher-order terms in the derivation led to much larger attacks than

necessary. The exact solution for these attacks was thrown off by small approximations, while an algorithm following a simple, iterative technique of following the slope of the loss function was successful.

Analysis and Future Research

Both the literature review and the simulations for this paper suggest that techniques like adversarial examples and data poisoning are viable against the future threat of AI. The DoD should investigate both defensive applications—such as robust AI and ML research—as well as offensive techniques—including blended attacks using both adversarial attacks and data poisoning in order to deceive adversary AI systems.

Though minimal attacks are interesting from a research perspective, the process of developing them indicates that they are more challenging to apply real-world applications. The approximations made during ML training suggest that a healthy margin likely is necessary to produce consistent effects. Also, military denial and deception usually consist of large perturbations: wearing face paint or covering a tank in foliage are very noticeable changes. AI research into the effectiveness of usual camouflage techniques against machine vision would be an interesting avenue of future research: AI could be effective in penetrating some denial and deception techniques or could be deceived simultaneously along with human vision. Adversarial examples, such as those demonstrated in *Adversarial Patch* by Tom B. Brown et al. (2017) could either be embedded into existing camouflage patterns or put on a patch deployed during combat in order to deceive AI systems.¹¹³ Similar techniques could embed text or cyberspace exploits into visually meaningless patterns as well, facilitating the introduction of malicious code from a physical object into a machine vision system.

Adversarial Examples Are Not Bugs, They Are Features, by Andrew (2019), is of particular interest from the perspective of training and designing AI algorithms.¹¹⁴ More research is needed on making algorithms focus on robust features—such as those that are used by humans during vehicle recognition—while suppressing spurious correlations. From an acquisitions perspective, this paper and Tsipiras et al. *Robustness May Be at Odds with Accuracy* (2019), indicate that vendors could develop vulnerable algorithms seeking to meet performance requirements or could promise performance that comes at the expense of robustness.¹¹⁵ Defense AI acquisitions programs should prioritize robustness in order to ensure that our future AI systems are not vulnerable to attack.

Professional Military Education (PME) should focus more on technical subjects across all PME programs. Military leaders need to understand both the incredible capabilities afforded by AI, as well as the severe limitations of the current state of the art. The U.S. military should make every effort to ensure data-literacy given the importance afforded to AI by our adversaries. Limiting technical education to a small number of PME students likely will exacerbate the problems identified by Col. J. Darren Duke in our limited approach to AI: “the U.S. military’s lag in applying this critical technological development is not due to a moral blind spot but rather a crippling type of illiteracy—data illiteracy.”¹¹⁶

The DoD and service components need to increase funding for AI experimentation alongside funding for new technologies. JAIC is ideally postured to coordinate experimentation at the department level and needs greater funding to do so. Service experimentation is also critical. AI will present an innovator’s dilemma, and institutional bias and resistance needs to be met with change-inducing experiences.¹¹⁷ Such experimentation should not only address how to

integrate AI enabled technology into our warfighting but must also address the challenges of fighting AI.

Conclusion

The AI elephants are coming. If we are lucky, we will also be riding them. However, the U.S. will not have 50 years to develop the right tactics to counter and leverage AI. The current pace of change in AI research and the ever-increasing levels of investment means that the state-of-the-art in AI research from five years ago is commonplace now. We need to start thinking about techniques for fighting AI while AI still underperforms, so that we can maintain our edge when AI represents the dominant battlefield technology.

AI challenges our conception of warfare. Autonomous systems make decisions about violence without supervision, ending a human monopoly on violence that has lasted for the entire history of warfare. Will a future conflict fought by machines even be considered a war? While this idea seems like science fiction, it encompasses conceptual challenges faced in space and cyberspace today.

This change does not mean an end to human participation in warfare. The US military must prepare its soldiers, sailors, airmen, and Marines for the challenge of fighting against machines. The impotence of tactics based on suppression will exacerbate the psychological impact of fighting a machine. However, we can develop new techniques, tactics, and procedures to leverage adversarial examples, data poisoning, and other yet-to-be-developed methods as substitutes for suppression. Just as the Romans had to learn to fight elephants, future Marines will have to learn to fight machines—and win.

Notes

-
- ¹Encyclopedia Britannica Online, s.v. “Pyrrhus,” accessed January 2021, <https://www.britannica.com/biography/Pyrrhus>
- ²Titus Livius, *The War With Hannibal*, ed. Betty Radice, trans. Aubrey de Selincourt, (Norwalk, C.T.: Easton Press, 1991), 650-651.
- ³Officer Candidates School, Officer Candidates School Order 1530.3Q Ch 1; Headquarters, Department of the Army, *ATP 3-21.8 INFANTRY PLATOON AND SQUAD*, (Washington, D.C.: Headquarters, Department of the Army, April 2016), J-2.
- ⁴Lt. Gen. HR McMaster, “On The Study Of War And Warfare,” *Modern War Institute*, accessed January 2021, <https://mwi.usma.edu/study-war-warfare/>,
- ⁵Carl von Clausewitz, *On War*, ed. Michael Howard and Peter Paret, trans. Michael Howard and Peter Paret (Princeton, NJ: Princeton University Press, 1976), 75.
- ⁶Bruce Weber, “Swift and Slashing, Computer Topples Kasparov,” *The New York Times*, May 12, 1997, accessed March 2021, <https://www.nytimes.com/1997/05/12/nyregion/swift-and-slashing-computer-topples-kasparov.html>; Clausewitz, *On War*, 605; Choe Sang-Hun and John Markoff, “Master of Go Board Game Is Walloped by Google Computer Program,” *The New York Times*, March 9, 2016, accessed March 2021, <https://www.nytimes.com/2016/03/10/world/asia/google-alphago-lee-se-dol.html>; Sang-Hun and Markoff, “Master of Go Board Game Is Walloped,” *The New York Times*.
- ⁷Alex Lee, “DeepMind has finally thrashed humans at StarCraft for real,” *Wired*, 30 October 2019, accessed March 2021, <https://www.wired.co.uk/article/deepmind-starcraft-alphastar>
- ⁸Dan Garisto, “Google AI beats top human players at strategy game StarCraft II,” *News, Nature*, 30 October 2019, accessed March 2021, <https://www.nature.com/articles/d41586-019-03298-6>
- ⁹Matt Bartlett, “The AI Arms Race in 2019,” *Towards Data Science*, Jan 28, 2019, accessed January 2021, <https://towardsdatascience.com/the-ai-arms-race-in-2019-fdca07a086a7>.
- ¹⁰Edith M. Lederer, “UN chief urges action to end Syria conflict, support rights,” *The Associated Press*, January 2020, accessed January 2021, <https://apnews.com/article/56e2368bc85fdc6517d30b5dae3f3c80>; Future of Life Institute, “An Open Letter Research Priorities For Robust And Beneficial Artificial Intelligence,” accessed January 2021, <https://futureoflife.org/ai-open-letter/>.
- ¹¹Headquarters US Marine Corps, MCDP-1: Warfighting (Washington, DC: US Marine Corps, June 30, 1991), 3.
- ¹²Sun Tzu, *The Art of War*, ed. and trans Lionel Giles (Hong Kong: Tuttle Publishing, 2008), 4.
- ¹³Deputy Secretary of Defense, Memorandum, Establishment of the Joint Artificial Intelligence Center, 27 June 2018.
- ¹⁴US Department of Defense, *Summary of the 2018 Department Of Defense Artificial Intelligence Strategy*, Washington, DC, 2018.
- ¹⁵Defense Advanced Research Projects Agency, “About US: Mission,” accessed March 2021, <https://www.darpa.mil/about-us/mission>.
- ¹⁶Defense Advanced Research Projects Agency, “ARPANET,” accessed March 2021, <https://www.darpa.mil/about-us/timeline/arpamet>; Defense Advanced Research Projects Agency, “TCP/IP,” accessed March 2021, <https://www.darpa.mil/about-us/timeline/tcp-ip>; Defense Advanced Research Projects Agency, “TCP/IP,” accessed March 2021, <https://www.darpa.mil/about-us/timeline/tcp-ip>.
- ¹⁷Defense Advanced Research Projects Agency, “Shakey the Robot,” accessed March 2021, <https://www.darpa.mil/about-us/timeline/shakey-the-robot>.
- ¹⁸Carnegie Mellon, “Shakey,” *The Robot Hall of Fame*, accessed March 2021, <http://www.robothalloffame.org/inductees/04inductees/shakey.html>.
- ¹⁹SRI International, “Shakey,” accessed March 2021, <http://www.ai.sri.com/shakey/>.
- ²⁰John Launchbury, “A DARPA Perspective on Artificial Intelligence,” Dec 23, 2015, PowerPoint presentation, accessed March 2021, <https://www.darpa.mil/attachments/AIFull.pdf>, 7.
- ²¹Launchbury, “A DARPA Perspective,” 5.
- ²²Launchbury, “A DARPA Perspective,” 11-23
- ²³Launchbury, “A DARPA Perspective,” 26.
- ²⁴Matt Turek, “Explainable Artificial Intelligence,” accessed March 2021, <https://www.darpa.mil/program/explainable-artificial-intelligence>
- ²⁵Bruce Draper, “Guaranteeing AI Robustness Against Deception (GARD),” accessed March 2021, <https://www.darpa.mil/program/guaranteeing-ai-robustness-against-deception>.

-
- ²⁶Robert O. Work, “Establishment of an Algorithmic Warfare Cross-Functional Team (Project Maven),” Department of Defense Memorandum, 26 April 2017, accessed January 2021, https://www.govexec.com/media/gbc/docs/pdfs_edit/establishment_of_the_awcft_project_maven.pdf.
- ²⁷Amado Cordova, Lindsay D. Millard, Lance Menthe, Robert A. Guffey, Carl Rhodes, “Motion Imagery Processing and Exploitation,” (Washington, D.C: Rand, 2013), 3.
- ²⁸Stew Magnuson, “Military ‘Swimming in Sensors and Drowning in Data,’” *National Defense*, 1 January 2020, accessed March 2021, <https://www.nationaldefensemagazine.org/articles/2009/12/31/2010january-military-swimming-in-sensors-and-drowning-in-data>.
- ²⁹John Keller, “Army to Brief Industry on Artificial Intelligence and Machine Learning Intelligence Data Processing,” *Military Aerospace*, 4 October, 2017, accessed March 2021, <https://www.militaryaerospace.com/computers/article/16726219/army-to-brief-industry-on-artificial-intelligence-and-machine-learning-for-intelligence-data-processing>.
- ³⁰Daisuke Wakabayashi and Scott Shane, “Google Will Not Renew Pentagon Contract That Upset Employees,” *The New York Times*, 1 June, 2018, <https://www.nytimes.com/2018/06/01/technology/google-pentagon-project-maven.html>.
- ³¹Deputy Secretary of Defense, Memorandum, Establishment of the Joint Artificial Intelligence Center, 27 June 2018.
- ³²Sydney J. Freedberg Jr., “Joint Artificial Intelligence Center Created Under DoD CIO,” *Breaking Defense*, 29 June 2018, accessed March 2021, <https://breakingdefense.com/2018/06/joint-artificial-intelligence-center-created-under-dod-cio/>.
- ³³US Department of Defense, Overview, Joint Artificial Intelligence Center Website, accessed January 2021. <https://dodcio.defense.gov/About-DoD-CIO/Organization/jaic/>.
- ³⁴Terri Moon Cronk, “DOD Unveils Its Artificial Intelligence Strategy,” *Defense.gov*, 12 February 2019, accessed March 2021, <https://www.defense.gov/Explore/News/Article/Article/1755942/dod-unveils-its-artificial-intelligence-strategy/>.
- ³⁵Defense Advanced Research Projects Agency, “AlphaDogfight Trials Foreshadow Future of Human-Machine Symbiosis,” 26 August, 2020, <https://www.darpa.mil/news-events/2020-08-26>.
- ³⁶Albert L., “AI Pilot Beats Human in AlphaDogfight Trials Finale,” *Overt Defense*, 20 August, 2020, accessed March 2021, <https://www.overtdefense.com/2020/08/20/a-i-pilot-beats-human-in-alphadogfight-trials-finale/>.
- ³⁷Clayton M. Christensen, *The Innovator’s Dilemma*, (Cambridge, M.A.: Harvard University Press, 2000), 228.
- ³⁸David B. Larter, “Unclear on unmanned: The US Navy’s plans for robot ships are on the rocks,” *Defense News*, 10 January, 2021, accessed March 2021, <https://www.defensenews.com/digital-show-dailies/surface-navy-association/2021/01/10/unclear-on-unmanned-the-us-navys-plans-for-robot-ships-are-on-the-rocks/>.
- ³⁹Tyler Rogoway, “The Alarming Case of the USAF’s Mysteriously Missing Unmanned Combat Air Vehicles,” *The Drive*, 9 June, 2016, accessed March 2021, <https://www.thedrive.com/the-war-zone/3889/the-alarming-case-of-the-usafs-mysteriously-missing-unmanned-combat-air-vehicles>.
- ⁴⁰J. Darren Duke, “Illiteracy, Not Morality, Is Holding Back Military Integration of Artificial Intelligence,” *The National Interest*, 15 February, 2021, <https://nationalinterest.org/feature/illiteracy-not-morality-holding-back-military-integration-artificial-intelligence-178261>.
- ⁴¹Elsa Kania and Rogier Creemers, “Xi Jinping Calls for ‘Healthy Development’ of AI (Translation),” *New America*, 5 November, 2018, accessed March 2021, <https://www.newamerica.org/cybersecurity-initiative/digichina/blog/xi-jinping-calls-for-healthy-development-of-ai-translation/>.
- ⁴²
- ⁴³Elsa Kania and Rogier Creemers, “Xi Jinping Calls for ‘Healthy Development’ of AI (Translation).”
- ⁴⁴Gregory C. Allen, *Understanding China’s AI Strategy: Clues to Chinese Strategic Thinking on Artificial Intelligence and National Security*, (Washington, D.C.: Center for a new American Security, February 2019), 8.
- ⁴⁵Christensen, *The Innovator’s Dilemma*, 228.
- ⁴⁶Graham Webster, Rogier Creemers, Paul Triolo, and Elsa Kania, “Full Translation: China’s ‘New Generation Artificial Intelligence Development Plan’ (2017),” 1 August, 2017, accessed March 2021, <https://www.newamerica.org/cybersecurity-initiative/digichina/blog/full-translation-chinas-new-generation-artificial-intelligence-development-plan-2017/>.
- ⁴⁷Webster, Creemers, Triolo, Kania, “China’s ‘New Generation Artificial Intelligence Development Plan.’”
- ⁴⁸China Institute for Science and Technology Policy at Tsinghua University, *China AI Development Report 2018*, (Beijing, China: China Institute for Science and Technology Policy), 50.
- ⁴⁹Allen, *Understanding China’s AI Strategy*, 7.
- ⁵⁰Webster, Creemers, Triolo, Kania, “China’s ‘New Generation Artificial Intelligence Development Plan.’”

-
- ⁵¹Tsinghua University, *China AI Development Report 2018*, 46-60.
- ⁵²US Department of State, Memorandum, "What is MCF One Pager," accessed March 2021, <https://www.state.gov/wp-content/uploads/2020/05/What-is-MCF-One-Pager.pdf>.
- ⁵³ Elsa B. Kania and Lorand Laskai, *Myths and Realities of China's Military-Civil Fusion Strategy*, (Washington, D.C.: Center for a New American Security, 2021), 4. <https://www.cnas.org/publications/reports/myths-and-realities-of-chinas-military-civil-fusion-strategy>; Webster, Creemers, Triolo, Kania, "China's 'New Generation Artificial Intelligence Development Plan.'" ⁵⁴Allen, *Understanding China's AI Strategy*, 5.
- ⁵⁵ <https://www.cnas.org/publications/reports/understanding-chinas-ai-strategy>
- ⁵⁶ Jiang Lianju, *Lectures on the Science of Space Operations* [kongjian zuozhan xue jiaocheng], (Beijing, China: Military Science Press, Jan 2013), 160-161.
- ⁵⁷Allen, *Understanding China's AI Strategy*, 5.
- ⁵⁸Allen, *Understanding China's AI Strategy*, 8; Elsa B. Kania, *Battlefield Singularity: Artificial Intelligence, Military Revolution, and China's Future Military Power*, (Washington, D.C.: Center for a new American Security, November 2017), 5.
- ⁵⁹Radina Gigova, "Who Vladimir Putin thinks will rule the world," *CNN*, 2 September, 2017, accessed March 2021, <https://edition.cnn.com/2017/09/01/world/putin-artificial-intelligence-will-rule-world/index.html>.
- ⁶⁰Center for Naval Analysis, "Spotlight: Russian Military Establishment Discusses AI," *Artificial Intelligence in Russia*, Issue 21, February 26, 2021, accessed March 2021, https://www.cna.org/CNA_files/PDF/DOP-2021-U-029296-Final.pdf.
- ⁶¹"The Use of Robots and the Widespread use of High Precision Weapons Will Become the Main Features of the wars of the Future: Chief of the General Staff of the Russian Army," (Translated) *Interfax AVN*, 24 March, 2018, accessed March 2021, <https://www.militarynews.ru/story.asp?rid=1&nid=476975&lang=RU>.
- ⁶²Samuel Bendett, "In AI, Russia Is Hustling to Catch Up," *Defense One*, 4 April, 2018, accessed March 2021, <https://www.defenseone.com/ideas/2018/04/russia-races-forward-ai-development/147178/>.
- ⁶³Samuel Bendett, "Autonomous Robotic Systems in the Russian Ground Forces," *Mad Scientist Laboratory* (US Army TRADOC Blog), 11 February, 2019, accessed March 2021, <https://madsciblog.tradoc.army.mil/120-autonomous-robotic-systems-in-the-russian-ground-forces/>.
- ⁶⁴Thomas Gibbons-Neff, "ISIS drones are attacking U.S. troops and disrupting airstrikes in Raqqa, officials say," *The Washington Post*, 14 June, 2017, accessed March 2021, <https://www.washingtonpost.com/news/checkpoint/wp/2017/06/14/isis-drones-are-attacking-u-s-troops-and-disrupting-airstrikes-in-raqqa-officials-say/>.
- ⁶⁵Droneshield promotional material, <https://www.droneshield.com/isis-use-drone-to-drop-grenade-on-tank#>.
- ⁶⁶Iman Ghosh, "AIoT: When Artificial Intelligence Meets the Internet of Things," *Visual Capitalist*, 12 August 2020, accessed March 2021, <https://www.visualcapitalist.com/aiot-when-ai-meets-iot-technology/>.
- ⁶⁷US Congress, *Artificial Intelligence and National Security*, (Washington, D.C.: Congressional Research Service, 10 November 2020), 1.
- ⁶⁸Defense Innovation Board, White Paper, "AI Principles: Recommendations on the Ethical Use of Artificial Intelligence by the Department of Defense –Supporting Document," November 2019, 8-10.
- ⁶⁹Launchbury, "A DARPA Perspective," 8.
- ⁷⁰Ali Shafahi et al. "Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks"; W. Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, Tom Goldstein, "Transferable Clean-Label Poisoning Attacks on Deep Neural Nets," *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97 (2019).
- ⁷¹Battista Biggio, Blaine Nelson, and Pavel Laskov. "Poisoning attacks against support vector machines," *International Conference on Machine Learning* (2012), 1467–1474
- ⁷²Abdul Rahim Ahmad, M. Khalia, C. Viard-Gaudin and E. Poisson, "Online handwriting recognition using support vector machine," 2004 IEEE Region 10 Conference TENCON 2004., Chiang Mai, Thailand, 2004, 311-314 Vol. 1; Shujun Huang, Nianguang Cai, Pedro Penzuti Pacheco, Shavira Narrandes, Yang Wang, Wayne Xu "Applications of Support Vector Machine (SVM) Learning in Cancer Genomics," *Cancer Genomics Proteomics*. 2018 Jan-Feb, Vol. 15, Issue 1, 41-51.
- ⁷³Yann LeCun, Corinna Cortes, and Christopher J.C. Burges, "The MNIST Database," accessed March 2021, <http://yann.lecun.com/exdb/mnist/>.
- ⁷⁴Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, (Heidelberg, New York:Springer, 13 January 2017), 23.

-
- ⁷⁵Marco Barreno et al. “Can machine learning be secure?” *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security (2006)*
- ⁷⁶Marco Barreno et al. “Can machine learning be secure?”
- ⁷⁷Matt Fredrickson, Somesh Jha, and Thomas Ristenpart, “Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures,” *Proceedings of the 22nd SIGSAC Conference on Computer and Communications Security*, October 2015, 1322-1333.
- ⁷⁸Marco Barreno et al. “Can machine learning be secure?”
- ⁷⁹Battista Biggio et al. “Evasion attacks against machine learning at test time,” *Joint European conference on machine learning and knowledge discovery in databases (2013)*, 387–402.
- ⁸⁰Christian Szegedy et al. “Intriguing properties of neural networks,” *International Conference on Learning Representations (2014)*.
- ⁸¹Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy, “Explaining and Harnessing Adversarial Examples.” *CoRR abstract (2015)*.
- ⁸²Andrew Ilyas et al. “Adversarial Examples Are Not Bugs, They Are Features,” *Advances in Neural Information Processing Systems*, Vol. 32 (NeurIPS 2019).
- ⁸³Nicolas Papernot, Patrick McDaniel and Ian J. Goodfellow, “Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples,” *ArXiv (2016)*.
- ⁸⁴Battista Biggio, Blaine Nelson, and Pavel Laskov. “Poisoning attacks against support vector machines,” *International Conference on Machine Learning (2012)*, 1467–1474.
- ⁸⁵Anish Athalye, Nicholas Carlini, and David Wagner, “Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples,” *Proceedings of the 35th International Conference on Machine Learning*, PMLR 80 (2018), 274-283.
- ⁸⁶Szegedy et al. “Intriguing properties of neural networks.”
- ⁸⁷Aleksander Madry et al. “Towards Deep Learning Models Resistant to Adversarial Attacks,” *ICLR 2018 Conference Blind Submission (2018)*.
- ⁸⁸Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio, “Adversarial Machine Learning at Scale,” *International Conference on Learning Representations 2017 Conference Submission (Nov 2016)*.
- ⁸⁹Florian Tramèr et al. “Ensemble Adversarial Training: Attacks and Defenses,” *Eighth International Conference on Learning Representations 2018 Conference Blind Submission (February 2018)*.
- ⁹⁰Dimitris Tsipras et al. “Robustness May Be at Odds with Accuracy,” *International Conference on Learning Representations 2019 Conference Blind Submission (2018)*.
- ⁹¹Carlini, Nicholas and David Wagner, “Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods,” *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (November 2017)*, 3–14.
- ⁹²Warren He et al. “Adversarial Example Defenses: Ensembles of Weak Defenses are not Strong,” *USENIX Workshop on Offensive Technologies (2017)*.
- ⁹³Battista Biggio et al. “Poisoning attacks against support vector machines.”
- ⁹⁴Huang Xiao et al. “Is feature selection secure against training data poisoning?” *International Conference on Machine Learning (2015)*.
- ⁹⁵Shike Mei, and Xiaojin Zhu, “Using machine teaching to identify optimal training-set attacks on machine learners,” *Association for the Advancement of Artificial Intelligence (2015)*.
- ⁹⁶Luis Muñoz-González et al. “Towards poisoning of deep learning algorithms with back-gradient optimization,” *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (2017)* 27–38.
- ⁹⁷Yingqi Liu et al. “Trojaning Attack on Neural Networks,” *Department of Computer Science Technical Reports, (2017)*, <https://docs.lib.purdue.edu/cstech/1781>.
- ⁹⁸Alexander Turner, Dimitris Tsipras, and Aleksander Madry, “Clean-Label Backdoor Attacks,” *Ninth International Conference on Learning Representations 2019 Conference Blind Submission (December 2018)*.
- ⁹⁹Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suci, Christoph Studer, Tudor Dumitras, Tom Goldstein, “Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks,” *Presented at the NIPS 2018 conference (April 2018)*.
- ¹⁰⁰Chen Zhu et al, “Transferable Clean-Label Poisoning Attacks on Deep Neural Nets,” *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97 (2019)
- ¹⁰¹W. Ronny Huang et al. “MetaPoison: Practical General-purpose Clean-label Data Poisoning,” *ArXiv abs/2004.00225 (2020)*.
- ¹⁰²Gabriela F. Cretu et al. “Casting out Demons: Sanitizing Training Data for Anomaly Sensors,” *2008 IEEE Symposium on Security and Privacy (2008)*, 81-95.

-
- ¹⁰³Jacob Steinhardt, Pangwei Koh, and Percy Liang, “Certified defenses for data poisoning attacks,” *Proceedings of the 31st International Conference on Neural Information Processing Systems* (December 2017), 3520–3532.
- ¹⁰⁴Bryant Chen et al. “Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering,” *ArXiv abs/1811.03728* (2019).
- ¹⁰⁵Ali Shafahi et al. “Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks”; W. Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, Tom Goldstein, “Transferable Clean-Label Poisoning Attacks on Deep Neural Nets,” *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97 (2019); Nehar Peri et al. “Deep k-NN Defense Against Clean-Label Data Poisoning Attacks,” *arXiv pre-publication*, arXiv–1909 (2019).
- ¹⁰⁶Colton Jones, “French Army will use AI to identify enemy combat vehicles,” *Defense Blog*, <https://defence-blog.com/news/army/french-army-will-use-ai-to-identify-enemy-combat-vehicles.html>.
- ¹⁰⁷DeepMind, “Alphago, the Story So Far,” <https://deepmind.com/research/case-studies/alphago-the-story-so-far>.
- ¹⁰⁸Nathan VanHoudnos, “Train, but Verify: Towards Practical AI Robustness,” Carnegie Mellon University Research Review 2020 PowerPoint presentation.
- ¹⁰⁹Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 109.
- ¹¹⁰Battista Biggio, Blaine Nelson, and Pavel Laskov. “Poisoning attacks against support vector machines.”
- ¹¹¹Guido van Rossum, Python tutorial, Technical Report CS-R9526, *Centrum voor Wiskunde en Informatica*, Amsterdam, May 1995; Thomas Kluyver, et al. “Jupyter Notebooks – a publishing format for reproducible computational workflows,” IOS Press, 2016, pp. 87-90.
- ¹¹²Charles R. Harris, et al. “Array programming with NumPy,” *Nature* 585, 357–362 (2020); Fabian Pedregosa, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, 12 (2011) 2825-2830, 12.
- ¹¹³Tom B. Brown et al. “Adversarial Patch,” *Conference and Workshop on Neural Information Processing Systems (NIPS) Workshop* (2017). <https://arxiv.org/abs/1712.09665>
- ¹¹⁴Andrew Ilyas et al. “Adversarial Examples Are Not Bugs, They Are Features.”
- ¹¹⁵Tsipras et al. “Robustness May Be at Odds with Accuracy.”
- ¹¹⁶J. Darren Duke, “Illiteracy, Not Morality, Is Holding Back Military Integration of Artificial Intelligence.”
- ¹¹⁷Chrstensen, *The Innovator’s Dilemma*, 228.

Appendix A

Below is the code that was used to simulate adversarial example and data poisoning attacks against

LDAs:

```
import numpy as np
import matplotlib.pyplot as plt
import copy
import statistics
from matplotlib.colors import ListedColormap
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import FunctionTransformer
from sklearn.covariance import EmpiricalCovariance
from sklearn.svm import SVC
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA

#This function generates a random covariance matrix
#and ensures that none of the eigenvalues are small
#in order to prevent the inverse from exploding
def Generate_Covariance(dimension, thresh):
    eigenvalues = np.zeros((dimension))
    while np.amin(eigenvalues) < thresh:
        A = np.random.rand(dimension,dimension)
        covariance_matrix = np.dot(A, A.transpose())
        eigenvalues = np.linalg.eigvals(covariance_matrix)
    return covariance_matrix

def Get_statistics(A_class, Target_class, A_class_size, Target_class_size):
    #This function computes the mean and covariance for each of the classes
    #print("Computing statistics for attack generation")
    cvf = EmpiricalCovariance()
    cov_class_A = cvf.fit(A_class)
    covariance_A = cov_class_A.covariance_
    mean_class_A = cov_class_A.location_
    #print("The observed mean for the class of the target data is ", mean_class_A)
    cov_target_class = cvf.fit(Target_class)
    mean_target_class = cov_target_class.location_
    #print("The observed mean for the target class is ", mean_target_class)
    covariance_target = cov_target_class.covariance_
    #Then, the function computes the overall covariance across the groups
    #using the formula from p 109 of Elements of Statistical Learning
    k_one = np.divide(A_class_size - 1, A_class_size + Target_class_size -2)
    k_two = np.divide(Target_class_size - 1, A_class_size + Target_class_size -2)
    covariance = np.multiply(k_one, covariance_A) + np.multiply(k_two, covariance_target)
    #print("The observed covariance is ", covariance)
    return mean_class_A, mean_target_class, covariance

#This function serves as the LDA from page 109 for Statistical Learning
#This function serves as the LDA from page 109 for Statistical Learning
```

```

def LDA_from_statistics(x_evaluate, mean_0, mean_1, covariance, size_0, size_1):
    covariance_inverse = np.linalg.inv(covariance)
    alpha = np.add(mean_0, mean_1)
    beta = np.subtract(mean_1, mean_0)
    threshold = np.multiply(.5, np.dot(alpha, np.dot(covariance_inverse, beta))) - np.log(np.divide(size_1,
size_0))
    value = np.dot(x_evaluate, np.dot(covariance_inverse, beta))
    evaluation = 0
    if value >= threshold:
        evaluation = 1
    return evaluation

```

#This function helps to plot the decision boundary for the LDA from
#statistics function

```

def LDA_Line_Values(mean_0, mean_1, covariance, size_0, size_1):
    covariance_inverse = np.linalg.inv(covariance)
    alpha = np.add(mean_0, mean_1)
    beta = np.subtract(mean_1, mean_0)
    threshold = np.multiply(.5, np.dot(alpha, np.dot(covariance_inverse, beta))) - np.log(np.divide(size_1,
size_0))
    vector = np.dot(covariance_inverse, beta)
    slope = np.divide(vector[0], vector[1])
    intercept = np.divide(threshold, vector[1])
    return slope, intercept

```

#This function generates an adversarial perturbation against an LDA

#It takes a target datum and statistics about the dataset to compute the attack

```

def LDA_Adversarial_Attack_Generator(a_target, mean_a, mean_b, covariance, Target_class_size,
A_class_size):
    beta = np.subtract(mean_b, mean_a)
    norm_beta_squared = np.dot(beta, beta)
    gradient_adversarial = np.multiply(np.divide(1, norm_beta_squared), np.dot(covariance, beta))
    constant = np.multiply(.5, np.dot(np.add(mean_b, mean_a), np.dot(np.linalg.inv(covariance), beta))) -
np.dot(a_target, np.dot(np.linalg.inv(covariance), beta)) - np.log(np.divide(Target_class_size,
A_class_size))
    zeta_adversarial_example = np.multiply(constant, gradient_adversarial)
    return zeta_adversarial_example

```

#This function gets the gradient of the margin for a poisoned LDA given

#a data-poisoning perturbation of the training set zeta, the target data

#and the current covariance and target set size

```

def LDA_Gradient(zeta, z, covariance, a_size):
    gradient = np.matmul(np.linalg.inv(covariance), zeta),
    gradient = np.divide(gradient, np.multiply(2, np.multiply(a_size, a_size)))
    gradient = np.divide(z, a_size) - gradient
    return gradient

```

#This function generates a single-point data poisoning attack using the gradient function above

```

def LDA_Data_Poisoning_Attack_Generator_Gradient(a_target, class_A, class_B, size_A, size_B, step):
    mean_A, mean_B, covariance_i = Get_statistics(class_A, class_B, size_A, size_B)
    test = LDA_from_statistics(a_target, mean_A, mean_B, covariance_i, size_A, size_B)

```

```

z = np.subtract(a_target, mean_B).transpose()
while test != 1:
    class_A[0] = class_A[0] + z.reshape(2)
    mean_A, mean_B, covariance_i = Get_statistics(class_A, class_B, size_A, size_B)
    test = LDA_from_statistics(a_target, mean_A, mean_B, covariance_i, size_A, size_B)
    z = np.subtract(a_target, mean_B).transpose()
return class_A[0]

#This function generates both attacks and compares them.
def LDA_Attack_Tester(mu_0, mu_1, sigma, size):

    #select machine learning algorithms
    clf_lda = LDA()
    clf_svm = SVC(kernel = 'linear')

    #generate data
    Class_0, Class_1, X, y = generate_data(mu_0, mu_1, sigma, sigma, size, size)
    Class_T0, Class_T1, Xt, yt = generate_data(mu_0, mu_1, sigma, sigma, size, size)
    #get the statistics for the LDA and the attack generator
    mean_class_0, mean_class_1, sigma_observed = Get_statistics(Class_0, Class_1, size, size)

    clf_lda.fit(X, y)
    clf_svm.fit(X, y)
    svm_vector = clf_svm.coef_[0]
    svm_intercept = clf_svm.intercept_[0]

    #select target data
    #select a target from Class_0 to attack/posion to Class_1
    t = 1
    u = 1
    v = 1
    aa = 1
    bb = 1
    while v != 0:
        target = np.random.multivariate_normal(mu_0, sigma, 1)
        t = LDA_from_statistics(target, mu_0, mu_1, sigma, size, size)
        u = LDA_from_statistics(target, mean_class_0, mean_class_1, sigma_observed, size, size)
        aa = clf_lda.predict(target)
        bb = clf_svm.predict(target)
        v = t + u + aa + bb
    print("The target data is ", target)

    step = 1
    Class_0_attacked_observed = copy.deepcopy(Class_0)
    z_ae_observed_stats = LDA_Adversarial_Attack_Generator(target, mean_class_0, mean_class_1,
sigma_observed, size, size)
    z_p_observed_stats = LDA_Data_Poisoning_Attack_Generator_Gradient(target,
Class_0_attacked_observed, Class_1, size, size, step)
    z_ae_svm = SVM_Adversarial_Attack_Generator(target, svm_vector, svm_intercept)

    #Demonstrate attacks

```

```
u = LDA_from_statistics(target + z_ae_observed_stats, mean_class_0, mean_class_1, sigma_observed,
size, size)
```

```
Class_0_attacked_observed = copy.deepcopy(Class_0)
Class_0_attacked_observed[0] = np.add(Class_0_attacked_observed[0], z_p_observed_stats)
mean_attacked_0o, mean_attacked_1o, sigma_attacked_o = Get_statistics(Class_0_attacked_observed,
Class_1, size, size)
q = LDA_from_statistics(target, mean_attacked_0o, mean_attacked_1o, sigma_attacked_o, size, size)
```

```
z_o = np.multiply(z_ae_observed_stats, 1)
if u == 0:
    delta = .01
    i = 1
    while u != 1:
        z_o = np.multiply(z_ae_observed_stats, 1 + delta)
        u = LDA_from_statistics(target + z_o, mean_class_0, mean_class_1, sigma_observed, size, size)
        delta += .01
        i += 1
```

```
else:
    delta = 0
    i = 0
    while u == 1:
        z_o = np.multiply(z_ae_observed_stats, 1 - delta)
        u = LDA_from_statistics(target + z_o, mean_class_0, mean_class_1, sigma_observed, size, size)
        delta += .01
        i += 1
```

```
z_p = z_p_observed_stats
Class_Po = copy.deepcopy(Class_0)
if q == 0:
    delta = .01
    i = 1
    while q != 1 and i != 100:
        z_p = np.multiply(z_p_observed_stats, 1 + delta)
        Class_Po[0] = np.add(Class_0[0], z_p)
        mu_0o, mu_1o, sigma_o = Get_statistics(Class_Po, Class_1, size, size)
        q = LDA_from_statistics(target, mu_0o, mu_1o, sigma_o, size, size)
        delta += .01
        i += 1
```

```
else:
    delta = 0
    i = 0
    while q == 1:
        z_p = np.multiply(z_p_observed_stats, 1 - delta)
        Class_Po[0] = np.add(Class_0[0], z_p)
        mu_0o, mu_1o, sigma_o = Get_statistics(Class_Po, Class_1, size, size)
        q = LDA_from_statistics(target, mu_0o, mu_1o, sigma_o, size, size)
        delta += .01
        i += 1
```

```
z_s = z_ae_svm
```

```
score_LDA = 0
```

```

score_poisoned = 0
Class_0pt = copy.deepcopy(Class_T0)
Class_0pt[0] = Class_T0[0] + z_p
mean_tp0, mean_tp1, sigma_tp0 = Get_statistics(Class_0pt, Class_T1, size, size)
for i in range(size + size):
    if LDA_from_statistics(Xt[i], mean_class_0, mean_class_1, sigma_observed, size, size) == yt[i]:
        score_LDA += 1
    if LDA_from_statistics(Xt[i], mean_tp0, mean_tp1, sigma_tp0, size, size) == yt[i]:
        score_poisoned += 1

return target, z_o, z_p, z_s, score_LDA, score_poisoned, Class_0, Class_1

#This function runs a series of tests (runs)
#for a given relationship between the centers of each cluster
def Attack_Size_Comparison(j, runs, center_1, center_2, dimension, thresh, data_size):
    ratio = 0
    score_un_poisoned = 0
    score_poisoned = 0
    i = 0
    thresh = .01
    while i < runs:
        print("RUN ", j, "/", i+1)
        spread = Generate_Covariance(dimension, thresh)
        mean_0 = np.ravel(np.random.multivariate_normal(center_1, spread, 1))
        mean_1 = np.ravel(np.random.multivariate_normal(center_2, spread, 1))
        cov_matrix = Generate_Covariance(dimension, thresh)
        target, z_o, z_p, z_s, score_LDA, score_p, Class_0, Class_1 = LDA_Tester(mean_0, mean_1,
cov_matrix, data_size)
        ratio += np.divide(np.linalg.norm(z_p), np.linalg.norm(z_o))
        print("ratio is ", ratio)
        score_un_poisoned += score_LDA
        print("score_un_poisoned is ", score_un_poisoned)
        score_poisoned += score_p
        print("score_poisoned is ", score_poisoned)
        i += 1
    print("Runs ", j, " Concluded")
    ratio = np.divide(ratio, i)
    score_un_poisoned = np.divide(score_un_poisoned, i)
    score_poisoned = np.divide(score_poisoned, i)
    return ratio, score_un_poisoned, score_poisoned

simulations = 100
runs = 100
dimension = 2
thresh = .01
data_size = 1000
center_1 = (1, 1)
center_2 = (1, 1)
step = (.1, .1)
j = 1

```

```

ratio_between_attacks = list()
score_for_poisoned_classifier = list()
score_for_LDA = list()

while j < simulations:
    center_2 = np.add(center_2, step)
    print("center 2 is ", center_2)
    ratio, score_un_poisoned, score_poisoned = Attack_Size_Comparison(j, runs, center_1, center_2,
dimension, thresh, data_size)
    ratio_between_attacks.append(ratio)
    score_for_poisoned_classifier.append(score_poisoned)
    score_for_LDA.append(score_un_poisoned)
    j += 1

```

The code below was used to investigate the difference between adversarial example and data poisoning attacks for the SciKit SVM:

```

import numpy as np
import matplotlib.pyplot as plt
import copy
from matplotlib.colors import ListedColormap
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import FunctionTransformer
from sklearn.covariance import EmpiricalCovariance
from sklearn.svm import SVC

#This function generates a random covariance matrix
#and ensures that none of the eigenvalues are small
#in order to prevent the inverse from exploding
def Generate_Covariance(dimension, thresh):
    eigenvalues = np.zeros((dimension))
    while np.amin(eigenvalues) < thresh:
        A = np.random.rand(dimension,dimension)
        covariance_matrix = np.dot(A, A.transpose())
        eigenvalues = np.linalg.eigvals(covariance_matrix)
    return covariance_matrix

#This function generates random data
#This function samples using a multivariate normal distribution
#mean_1, cov_1 are the mean and covariance for class 0
#mean_2, cov_2 are the mean and covariance for class 1
#size_1, size_2 are the number of data points in class 0, 1
def generate_data(mean_1, mean_2, cov_1, cov_2, size_1, size_2):
    #X is the generated data set
    class_zero = np.random.multivariate_normal(mean_1, cov_1, size_1)
    class_one = np.random.multivariate_normal(mean_2, cov_2, size_2)
    #A = np.r_[class_zero, class_one + np.array([1, 1])]
    A = np.r_[class_zero, class_one]
    #y is the vector of class variables for X

```

```

b = np.hstack((np.zeros(size_1), np.ones(size_2)))
return class_zero, class_one, A, b

def Get_statistics(A_class, Target_class, A_class_size, Target_class_size):
    #This function computes the mean and covariance for each of the classes
    #print("Computing statistics for attack generation")
    cvf = EmpiricalCovariance()
    cov_class_A = cvf.fit(A_class)
    covariance_A = cov_class_A.covariance_
    mean_class_A = cov_class_A.location_
    #print("The observed mean for the class of the target data is ", mean_class_A)
    cov_target_class = cvf.fit(Target_class)
    mean_target_class = cov_target_class.location_
    #print("The observed mean for the target class is ", mean_target_class)
    covariance_target = cov_target_class.covariance_
    #Then, the function computes the overall covariance across the groups
    #using the formula from p 109 of Elements of Statistical Learning
    k_one = np.divide(A_class_size - 1, A_class_size + Target_class_size - 2)
    k_two = np.divide(Target_class_size - 1, A_class_size + Target_class_size - 2)
    covariance = np.multiply(k_one, covariance_A) + np.multiply(k_two, covariance_target)
    #print("The observed covariance is ", covariance)
    return mean_class_A, mean_target_class, covariance

#This function generates an adversarial perturbation against an SVM
#It takes a target datum and the values w and b for the equation  $y = wx + b$ 
#of the maximally separating hyperplane of the SVM
def SVM_Adversarial_Attack_Generator(target, w, b):
    #First, this function calculates the distance between the target and the hyperplane
    distance = np.divide(np.absolute(np.add(np.dot(target, w), b)), np.linalg.norm(w))
    #then it creates a vector along that distance to generate the adversarial example
    zeta_adversarial_example = np.multiply(np.divide(w, np.linalg.norm(w)), distance)
    return zeta_adversarial_example

#This function calculates the gradient for a poisoning attack
#against a SVM
#Based on the paper Poisoning Attacks against Support Vector Machines
#by Biggio
def SVM_Poisoning_Attack_Gradient(c, s_indices, a, y, data_size):
    j = 0
    for i in range(np.shape(s_indices)[0]-1):
        if c[0][i]<1 and c[0][i] > -1:
            j += 1
    v_s = np.zeros((1,j))
    q_s = np.zeros((j, 2))
    i = 0
    for i in range(j):
        v_s[0][i] = y[s_indices[i]]
        q_s[i] = A[s_indices[i]]*v_s[0][i]
    q_ss = np.dot(q_s, q_s.transpose())
    gamma = np.dot(v_s, np.dot(np.linalg.inv(q_ss), v_s.transpose()))
    v = np.dot(np.linalg.inv(q_ss), v_s.transpose()).reshape(1,j)

```

```

q_t = np.multiply(np.ones((1,1)), target)
q_ks = np.dot(q_t, q_s.transpose())
m_k = np.multiply(gamma, np.linalg.inv(q_ss))
m_k = m_k - np.dot(v.transpose(), v)
m_k = np.dot(q_ks, m_k)
m_k = m_k - v
m_k = -np.divide(m_k, gamma)
length_y = len(y)-1
length_A = len(A)-1
q_c = np.dot(np.ones(1).reshape(1,1),
             A[length_A].reshape(dimension,1).transpose()).transpose()
q_sc = np.dot(q_s, q_c).reshape(j,1)
q_k = np.dot(-np.ones((1,1)), target)
q_kc = np.dot(q_k, q_c)
gradient = np.dot(m_k, q_sc)
gradient = np.add(gradient, q_kc)
gradient = np.multiply(gradient, c[0][j])
return gradient

```

#This function generates a data poisoning attack against a SVM

#Based on the paper Poisoning Attacks against Support Vector Machines

#by Biggio

def SVM_Data_Poisoning_Attack_Generator(target, target_p, Class_0, Class_1, A, b, data_size, step):

```

    clf_svm = SVC(kernel = 'linear')
    x_c = target_p
    A = np.r_[Class_0, Class_1, x_c]
    b = np.hstack((np.zeros(data_size), np.ones(data_size + 1)))
    clf_svm.fit(A, b)
    x_c_list = list()
    x_c_list.append(x_c)
    while clf_svm.predict(target)[0] != 1:
        print(clf_svm.predict(target))
        a = clf_svm.support_vectors_
        s_indices = clf_svm.support_
        c = clf_svm.dual_coef_
        y = copy.deepcopy(b)
        y = np.multiply(y, 2)
        y = np.subtract(y, 1)
        alphas = clf_svm.dual_coef_
        grad = SVM_Poisoning_Attack_Gradient(c, s_indices, a, y, data_size)
        grad = np.multiply(grad, np.multiply(step, x_c))
        grad = np.divide(grad, np.linalg.norm(grad))
        print("grad is ", grad)
        x_c += np.multiply(step, grad)
        if np.any(x_c_list == x_c):
            step = np.multiply(step, 1.5)
        x_c_list.append(x_c)
        print("x_c is ", x_c)
        A[2*data_size] = x_c
        clf_svm.fit(A, b)
        print("x_c is ", A[2*data_size])

```



```

print(clf_svm.predict(target))
return x_c

def SVM_Attack_Tester(mu_0, mu_1, sigma, size):

    #select machine learning algorithms
    clf_lda = LDA()
    clf_svm = SVC(kernel = 'linear')

    #generate data
    Class_0, Class_1, X, y = generate_data(mu_0, mu_1, sigma, sigma, size, size)
    #get the statistics for the LDA and the attack generator
    mean_class_0, mean_class_1, sigma_observed = Get_statistics(Class_0, Class_1, size, size)

    clf_lda.fit(X, y)
    clf_svm.fit(X, y)
    svm_vector = clf_svm.coef_[0]
    svm_intercept = clf_svm.intercept_[0]

    #select target data
    #select a target from Class_0 to attack/posion to Class_1
    t = 1
    u = 1
    v = 1
    aa = 1
    bb = 1
    dd = 1
    while v != 0:
        target = np.random.multivariate_normal(mu_0, sigma, 1)
        t = LDA_from_statistics(target, mu_0, mu_1, sigma, size, size)
        u = LDA_from_statistics(target, mean_class_0, mean_class_1, sigma_observed, size, size)
        aa = clf_lda.predict(target)
        bb = clf_svm.predict(target)
        v = t + u + aa + bb
    while dd != 0:
        target_p = target = np.random.multivariate_normal(mu_0, sigma, 1)
        dd = clf_svm.predict(target)

    print("The target data is ", target)
    return target, target_p, Class_0, Class_1, X, y

i = 0
runs = 10
center_1 = (1, 1)
center_2 = (3, 3)
dimension = 2
thresh = .01
spread = [[4, 0],[0, 4]]
data_size = 10
dimension = 2
thresh = .01

```

```

mean_0 = np.ravel(np.random.multivariate_normal(center_1, spread, 1))
mean_1 = np.ravel(np.random.multivariate_normal(center_2, spread, 1))
cov_matrix = Generate_Covariance(dimension, thresh)
print("stats for the run ", mean_0, mean_1, cov_matrix)

target, target_p, Class_0, Class_1, A, b = SVM_Attack_Tester(mean_0, mean_1, cov_matrix, data_size)
clf_svm = SVC(kernel = 'linear')
clf_svm.fit(A, b)
target_b = copy.deepcopy(target)
w = clf_svm.coef_[0]
b = clf_svm.intercept_[0]
x_ae = SVM_Adversarial_Attack_Generator(target, w, b)

step = 2
x_c = SVM_Data_Poisoning_Attack_Generator(target, target_p, Class_0, Class_1, A, b, data_size, step)

poison = x_c - Class_0[0]

print("The Adversarial Attack is ", x_ae, "and the poisoning attack is ", x_c)

#demonstrate the attack

A = np.r_[Class_0, Class_1]
b = np.hstack((np.zeros(data_size), np.ones(data_size)))
clf_svm.fit(A, b)
t = clf_svm.predict(target_b)

C = np.r_[Class_0, Class_1, x_c]
d = np.hstack((np.zeros(data_size), np.ones(data_size + 1)))
clf_svm.fit(C, d)
u = clf_svm.predict(target_b)

print(t, u)

```

Bibliography

- Ahmad, Abdul Rahim, M. Khalia, C. Viard-Gaudin and E. Poisson, "Online handwriting recognition using support vector machine." *2004 IEEE Region 10 Conference TENCN 2004.*, Chiang Mai, Thailand, 2004, 311-314 Vol. 1.
- Allen, Gregory C.. *Understanding China's AI Strategy: Clues to Chinese Strategic Thinking on Artificial Intelligence and National Security*. Washington, D.C.: Center for a new American Security, February 2019.
- Athalye, Anish, Nicholas Carlini, and David Wagner. "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples." *Proceedings of the 35th International Conference on Machine Learning*, PMLR 80 (2018), 274-283.
- Barreno, Marco, Blaine Nelson, Russell Sears, Anthony Joseph, J. D. Tygar. "Can machine learning be secure?" *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security* (2006).
- Biggio, Battista, Blaine Nelson, and Pavel Laskov. "Poisoning attacks against support vector machines." *International Conference on Machine Learning* (2012), 1467–1474.
- Biggio, Battista, Iginio Corona, Davide Maiorca, Blaine Nelson, Nedim Srndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. "Evasion attacks against machine learning at test time." *Joint European conference on machine learning and knowledge discovery in databases* (2013), 387–402.

Brown, Tom, Dandelion Mane, Aurko Roy, Martin Abadi, Justin Gilmer. “Adversarial Patch.” *Conference and Workshop on Neural Information Processing Systems (NIPS) Workshop* (2017). <https://arxiv.org/abs/1712.09665>

Carlini, Nicholas and David Wagner. “Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods.” *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security* (November 2017), 3–14.

Center for Naval Analysis. “Spotlight: Russian Military Establishment Discusses AI.” *Artificial Intelligence in Russia*, Issue 21, February 26, 2021. accessed March 2021, https://www.cna.org/CNA_files/PDF/DOP-2021-U-029296-Final.pdf.

Chen, Bryant, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, Biplav Srivastava. “Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering.” *ArXiv abs/1811.03728* (2019).

Chen Zhu, W. Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, Tom Goldstein “Transferable Clean-Label Poisoning Attacks on Deep Neural Nets.” *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97 (2019), 7614-7623.

China Institute for Science and Technology Policy at Tsinghua University. *China AI*

Development Report 2018. Beijing, China: China Institute for Science and Technology Policy, 2018.

Christensen, Clayton M. *The Innovator's Dilemma*. Cambridge, M.A.: Harvard University Press, 2000.

Cordova, Amado, Lindsay D. Millard, Lance Menthe, Robert A. Guffey, Carl Rhodes. *Motion Imagery Processing and Exploitation*. Washington, D.C: Rand, 2013.

Clausewitz, Carl von. *On War*. Edited by Michael Howard and Peter Paret. Translated by Michael Howard and Peter Paret. Princeton, NJ: Princeton University Press, 1984.

Cretu-Ciocarlie, Gabriela F., Angelos Stavrou, Michael E. Locasto, Salvatore Stolfo, Angelos D. Keromytis. "Casting out Demons: Sanitizing Training Data for Anomaly Sensors." *2008 IEEE Symposium on Security and Privacy* (2008), 81-95.

Dritsoula, Lemonia, Patrick Loiseau, and John Musacchio. "A Game-Theoretic Analysis of Adversarial Classification." *IEEE Transactions on Information Forensics and Security* (Volume: 12, Issue: 12, Dec. 2017). <https://ieeexplore.ieee.org/document/7954621>

Fredrickson, Matt, Somesh Jha, and Thomas Ristenpart. "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures." *Proceedings of the 22nd SIGSAC Conference on Computer and Communications Security*, October 2015, 1322-1333.

Goodfellow, Ian J., Jean Pouget-Abadie, M. Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville and Yoshua Bengio. "Generative Adversarial Nets." *NIPS* (2014).

Goodfellow, Ian J., Jonathon Shlens and Christian Szegedy. "Explaining and Harnessing Adversarial Examples." *CoRR abs/1412.6572* (2015).

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Heidelberg, New York. Springer. 13 January 2017.

He, Warren, James Wei, Xinyun Chen, Nicholas Carlini, Dawn Song. "Adversarial Example Defenses: Ensembles of Weak Defenses are not Strong." *USENIX Workshop on Offensive Technologies* (2017).

Headquarters, Department of the Army. ATP 3-21.8 INFANTRY PLATOON AND SQUAD. Washington, D.C.: Headquarters, Department of the Army, April 2016.

Headquarters US Marine Corps. Warfighting. MCDP 1. Washington, DC: Headquarters US Marine Corps, June 30, 1991.

Huang, Shujun, Nianguang Cai, Pedro Penzuti Pacheco, Shavira Narrandes, Yang Wang, Wayne

Xu. "Applications of Support Vector Machine (SVM) Learning in Cancer Genomics." *Cancer Genomics Proteomics*. 2018 Jan-Feb, Vol. 15, Issue 1, 41-51.

Huang, W. Ronny, Jonas Geiping, Liam Fowl, Gavin Taylor, Tom Goldstein. "MetaPoison: Practical General-purpose Clean-label Data Poisoning." *ArXiv abs/2004.00225* (2020).

Hunter, John. D. "Matplotlib: A 2D Graphics Environment." *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90-95, 2007.

Ilyas, Andrew, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, Aleksander Madry. "Adversarial Examples Are Not Bugs, They Are Features," *Advances in Neural Information Processing Systems*, Vol. 32 (NeurIPS 2019).

Kania, Elsa B. "*AI Weapons*" *In China's Military Innovation*. Washington, DC: Brookings, 2020. <https://www.brookings.edu/research/ai-weapons-in-chinas-military-innovation>.

Kania, Elsa B. *Battlefield Singularity: Artificial Intelligence, Military Revolution, and China's Future Military Power*. Washington, D.C.: Center for a new American Security, November 2017.

Kania, Elsa B. and Lorand Laskai. *Myths and Realities of China's Military-Civil Fusion Strategy*. (Washington, D.C.: Center for a New American Security, 2021).

Kluyver, Thomas, Benjamin Ragan-Kelley , Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and the Jupyter development team. “Jupyter Notebooks – a publishing format for reproducible computational workflows,” *IOS Press*, 2016, pp. 87-90.

Kurakin, Alexey, Ian J. Goodfellow, and Samy Bengio. “Adversarial Machine Learning at Scale.” *International Conference on Learning Representations 2017 Conference Submission* (Nov 2016). <https://arxiv.org/abs/1611.01236>

Lianju, Jiang. *Lectures on the Science of Space Operations* [kongjian zuozhan xue jiaocheng]. Beijing, China: Military Science Press, Jan 2013.

Liu, Yingqi, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, Xiangyu Zang. "Trojaning Attack on Neural Networks" *Department of Computer Science Technical Reports.Paper* (2017), <https://docs.lib.purdue.edu/cstech/1781>.

Livius, Titus. *The War With Hannibal*. Edited by Betty Radice. Translated by Aubrey de Selincourt Norwalk, C.T.: Easton Press, 1991.

Madry, Aleksander, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu. “Towards Deep Learning Models Resistant to Adversarial Attacks.” *ICLR 2018 Conference Blind Submission* (2018). <https://arxiv.org/abs/1706.06083>

Mei, Shike and Xiaojin Zhu. “Using machine teaching to identify optimal training-set attacks on machine learners.” *Association for the Advancement of Artificial Intelligence* (2015).

Muñoz-González, Luis, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. “Towards poisoning of deep learning algorithms with back-gradient optimization.” *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security* (2017) 27–38.

Nguyen, Anh, Jason Yosinski, and Jeff Clune. “Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images.” *2015 IEEE Conference on Computer Vision and Pattern Recognition* (June 2015).
<https://ieeexplore.ieee.org/document/7298640>

Papernot, Nicolas, P. McDaniel and Ian J. Goodfellow. “Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples.” *ArXiv abs/1605.07277* (2016).

Papernot, Nicholas, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Berkay Celik, and Ananthram Swami. “Practical black- box attacks against machine learning.” *Proceedings of the 2017 Association for Computing Machinery on Asia conference on computer and communications security* (2017), 506–519.

Pedregosa, Fabian. “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, 12 (2011) 2825-2830, 12.

Peri, Neehar, Neal Gupta, W. Ronny Huang, Liam Fowl, Chen Zhu, Soheil Feizi, Tom Goldstein, John P. Dickerson. “Deep k-NN Defense Against Clean-Label Data Poisoning Attacks.” *arXiv pp. arXiv-1909* (2019)

Shafahi, Ali, W. Ronny Huang, Mahyar Najibi, Octavian Suci, Christoph Studer, Tudor Dumitras, Tom Goldstein. “Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks.” *Presented at the NIPS 2018 conference* (April 2018)
<https://arxiv.org/abs/1804.00792>

Steinhardt, Jacob, Pangwei Koh, and Percy Liang. “Certified defenses for data poisoning attacks” *Proceedings of the 31st International Conference on Neural Information Processing Systems* (December 2017), 3520–3532.

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, Rob Fergus. “Intriguing properties of neural networks.” *International Conference on Learning Representations* (2014).

Tramèr, Florian, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, Patrick

McDaniel. “Ensemble Adversarial Training: Attacks and Defenses.” *Eighth International Conference on Learning Representations 2018 Conference Blind Submission*, (February 2018).

Tsipras, Dimitris, Santurkar, Shibani, Engstrom, Logan, Turner, Alexander and Madry, Aleksander. “Robustness May Be at Odds with Accuracy.” *International Conference on Learning Representations 2019 Conference Blind Submission* (2018).
<https://arxiv.org/abs/1805.12152>

Turner, Alexander, Dimitris Tsipras, and Aleksander Madry, “Clean-Label Backdoor Attacks.” *Ninth International Conference on Learning Representations 2019 Conference Blind Submission* (December 2018).

Tzu, Sun. *The Art of War*. Edited and translated by Lionel Giles. Hong Kong: Tuttle Publishing, 2008.

US Department of Defense. *Summary of the 2018 Department Of Defense Artificial Intelligence Strategy*. Washington, DC, 2018. <https://media.defense.gov/2019/Feb/12/2002088963/-1/-1/1/SUMMARY-OF-DOD-AI-STRATEGY.PDF>

van Rossum, Guido. “Python tutorial, Technical Report CS-R9526,” *Centrum voor Wiskunde en Informatica*, Amsterdam, May 1995.

Xiao, Huang, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, Fabio Roli. “Is feature selection secure against training data poisoning?” *International Conference on Machine Learning* (2015).

Yuan, Xiaoyong, Pan He, Qile Zhu, Rajendra Rana Bhat, Xiaolin Li. “Adversarial Examples: Attacks and Defenses for Deep Learning.” *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Neural Networks and Learning Systems* (Volume: 30, Issue: 9, Sept. 2019). <https://ieeexplore.ieee.org/document/8611298>

Zhu, Chen, W. Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, Tom Goldstein. “Transferable Clean-Label Poisoning Attacks on Deep Neural Nets.” *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97 (2019)