Editorial Committee: Gary Locklair, O. Joshua Locklair, Robert Wahl
Editor-in-chief: Robert Wahl
Assistant Editor: Valerie Locklair

Publisher: Concordia University Wisconsin, 12800 N. Lake Shore Drive, Mequon, WI 53097-2402

The Concordia University System (CUS) is comprised of 10 colleges and universities affiliated with The Lutheran Church – Missouri Synod. Located across the United States, the colleges and universities offer more than 160 undergraduate and 50 graduate programs. Additional information is available at: http://www.cus.edu

This new journal, The Concordia Technical Journal, will be published semi-annually. Our goal is to provide an outlet to showcase research undertaken within our schools while incorporating and celebrating our Christian worldviews.

Scholarly articles for inclusion may be sent to Dr. Robert Wahl, Concordia University Wisconsin, 12800 N. Lake Shore Drive, Mequon, WI 53097-2402, or techjournal@cuw.edu.

Books for review may be sent to Dr. Robert Wahl, Concordia University Wisconsin, 12800 N. Lake Shore Drive, Mequon, WI 53097-2402, or techjournal@cuw.edu.

Books received related to the purposes of the Concordia Technical Journal will be reviewed within six months of the reception.

Concordia Technical Journal
ISSN 2475-1529
**Table of Contents**

Editor’s Note .................................................. 5

Connecting Art and Science:
Computer Science as a Liberal Art ....................... 7

Developing a Test Infrastructure for
Measuring Cyber-Attacks and
Latency in Intrusion Detection Systems ............... 21

Giant Thinking... Minds? The Problem of
Strong Artificial Intelligence ......................... 35

Using ASICs to Train Neural Networks ................. 49

Machine Learning Methods for
Filtering Spam Email: A Literature Review .......... 55
Editor’s Note

Welcome to the inaugural issue of the Concordia Technical Journal. Our goal for this journal is to provide students, part-time faculty, adjuncts, and full-time faculty within the Concordia University System with a showcase for ongoing technical research in a peer-reviewed format. We look forward to interacting with you within these pages. If you have a research paper that you would like to submit for publication, it can be sent to techjournal@cuw.edu.

The publication, developed by the Computer Science department at Concordia University Wisconsin, is a peer-reviewed technical journal featuring, and limited to, research conducted by faculty and students of the 10 colleges and universities of the Concordia University System (CUS) of the Lutheran Church—Missouri Synod.

This new journal, The Concordia Technical Journal, will be published semi-annually. Our goal is to provide an outlet to showcase research undertaken within our schools while incorporating and celebrating our Christian worldviews. If you are interested in providing a peer-review of submitted papers, please contact us through the same email address.

Robert Wahl, Ph. D.
March, 2018
Connecting Art and Science: Computer Science as a Liberal Art
by
Gary H. Locklair

Computer science is a liberal art that connects art and science by bridging the boundaries between them.

Abstract
Computer science is an integrative, multi-disciplinary activity – it is a liberal art. An effective computer scientist must understand English and history (grammar), philosophy and mathematics (logic), and science and communication (rhetoric). Computer science is not about studying technology; it is about solving problems for others. We love and serve our neighbors as we solve their problems in our vocations. Successful problem solving requires a broad liberal arts education.

Computer science should be studied by everyone because it is a liberal art. As a discipline, computer science integrates art and science. As a liberal art, computer science provides a framework for learning and understanding not only technology but also many other subjects and disciplines, as well.

A liberal arts education provides freedom. (The term “liberal arts” is derived from the Latin libera, freedom.) The liberal arts are designed to educate the whole person for a life-long pursuit of truth. Computer science should be studied by everyone since all the technology that permeates society is based upon computer science. Understanding computer science as a discipline is necessary for a free citizen. Understanding computer science means one is neither a slave to technology nor at the mercy of the technological elite. Instead, a liberal arts knowledge of computer science allows a free citizen to harness technology in order to serve other people.
Introduction

“This is so weird … how does Amazon.com know I would be interested in this item?” The recent immigrant from Africa was amazed at the prospect of online shopping, and was curious about the computer science behind the experience.¹

The concise answer to the question is analytics. In this case, analytics involves big data analysis coupled with data mining techniques. Amazon’s recommendation system likely employs affinity analysis. “Affinity Analysis is a technique to identify the logical clustering of data and processes.”² “At its core, an affinity analysis is a data mining technique that uses association rule learning to identify the relationships between customers and the attributes related to them.”³

Analytics is the analysis (discovery), explanation (interpretation), and communication (dissemination) of information. Analytics is especially concerned with patterns and trends in historical data. “Analytics is an encompassing and multidimensional field that uses mathematics, statistics, predictive modeling and machine-learning techniques to find meaningful patterns and knowledge in recorded data.”⁴

Analytics is a great example of the liberal arts nature of computer science. Analytics encompasses foundational computer science concepts (eg, algorithms and methods of problem solving), mathematics (eg, statistics), and business (eg, operations research) to help inform effective decision making. Analytics can easily

---

¹ “Amazon currently uses item-to-item collaborative filtering, which scales to massive data sets and produces high-quality recommendations in real time. This type of filtering matches each of the user's purchased and rated items to similar items, then combines those similar items into a recommendation list for the user.” (http://rejoiner.com/resources/amazon-recommendations-secret-selling-online/)
² See http://it.toolbox.com/blogs/enterprise-solutions/affinity-analysis-14726
³ See https://blog.rjmetrics.com/2016/10/12/to-affinity-analysis-and-beyond/
⁴ see https://www.sas.com/en_us/insights/analytics/what-is-analytics.html#)
“reach” into psychology (behavior) and biology (machine learning), also. The analysis (discovery) is done using algorithms. Communication is done via graphics (meaningful charts and graphs), which is also a central aspect of computer science. Thus, computer science bridges and connects multiple disciplines in its pursuit of useful information.

**Arts and Sciences**

What’s the difference between art and science? According to computer scientist Dr. Don Knuth, “Science is what we understand well enough to explain to a computer. Art is everything else we do.”

Art and science are human activities that seem to have different purposes, goals, and processes. They appear to operate in different arenas and be practiced by different kinds of people. Because of this, some see a chasm between the arts and the sciences. Art and science are often viewed as separate spheres. Many will quip, ‘art expresses knowledge while science uncovers knowledge.’ Karen Hardison declared, “There are two paramount differences between art and science. The first is that art is subjective while science is objective. The second is that art expresses knowledge, most often in the form of subjective representation, while science is the system of acquiring knowledge. Art and science are therefore in fundamental character very dissimilar.”

However, art and science should not be separate. Dr. Don Knuth stated, “Science is knowledge which we understand so well that we can teach it to a computer; and if we don't fully understand

---


6 see https://www.enotes.com/homework-help/deference-between-art-science-172919
something, it is an art to deal with it.” He went on to explain, “In this sense, we should continually be striving to transform every art into a science: in the process, we advance the art.”

Arts and sciences should not be separate arenas. As an analogy consider the relationship between knowledge and wisdom. Knowledge consists of facts and principles. Wisdom is coming to a proper conclusion based upon that knowledge. Wisdom is the right application of what is known. It would be foolish to separate knowledge and wisdom. As the writer of Proverbs indicates “I, wisdom, dwell with prudence, and I find knowledge and discretion.” It would be equally foolish for an educated and free person to separate arts and sciences.

An educated person must understand both art and science. The purpose of the liberal arts is to integrate and connect knowledge and not fragment it. An educated person must have an extensive range of knowledge as a foundation before beginning to specialize in a particular field. This wide view is vital to understanding any specific discipline. As a college professor, I smile when a computer science student asks why English courses are required. In computer science, as in other disciplines, communication is the key to problem solving. If someone is unable to communicate precisely, problems cannot be solved. If he hasn’t mastered the “art” of language, the technologist will be a poor practitioner of his science.

**Liberal Arts**

The term “liberal arts” is derived from the Latin *libera*, freedom. A liberal arts education is considered necessary for a person to be free. In the Middle ages seven “arts” were a necessary

---

7 1974 Turing Award Lecture, Communications of the ACM 17 (12), (December 1974), pp. 667–673
8 Proverbs 8:12 (ESV)
9 see https://home.isi.org/purpose-liberal-arts
part of a free person’s education, the trivium (three parts) and quadrivium (four parts). The trivium was composed of grammar, logic, and rhetoric. The quadrivium consisted of astronomy, geometry, mathematics, and music. Liberal arts educate the whole person for a life-long pursuit of truth. A liberal arts education is grounded in the truth of God’s Word as revealed in Scripture and the person of Christ. A liberal arts education helps a free person understand his place in God’s creation.  

The liberal arts encompass both knowledge and a way of knowing. They consist of both a group of subjects and a way to learn about any other subject. The trivium can be knowledge about language, its structure (grammar), meaning (logic), and expression (rhetoric). In addition, the trivium can also serve as a framework to learn about any discipline. Computer science has a grammar (basic facts and concepts), logic (analyze problems and systems), and rhetoric (create new software tools). In order to program, a computer scientist must understand the syntax and semantics of a programming language (grammar). Before creating real systems, the computer scientist will learn to implement basic constructs in the language such as loops for repetition and decisions for selection (logic). Finally, the computer scientist will apply the knowledge and logical skills of programming to construct complex systems that will allow users to interact with a system to solve problems (rhetoric).

Thus, the liberal arts are a way to connect all spheres of learning. An effective computer scientist must understand many subjects in order to create systems. Creating an artificial intelligence application such as speech recognition requires understanding biology, communication, engineering, English, history, logic, mathematics, philosophy, psychology, and science.

10 See http://www.classicalwriting.com/blog/2012/05/18/a-liberal-arts-education-for-the-freeborn/
11 See http://public.callutheran.edu/~mccamb/veithand.htm
Producing educated citizens who are able to love and serve their neighbors well in their vocation is the goal of the liberal arts. They equip citizens to think critically and ultimately serve their neighbors as God’s masks in this world. While anyone can collect facts, it takes more than mere facts to truly serve others. A collection of disconnected knowledge will not help solve a problem. It is only by integrating the knowledge in a framework of wisdom that we put our inspiration into action and energize our vocations.

Vedika Khemani wrote, “Real-world problems rarely ever have textbook solutions. More than anything, the purpose of a college education is to learn how to think critically and what questions to ask. Liberal arts colleges aim to mold their students into well-rounded, well-informed global citizens with a wide skill set — whether it is through elective or voluntary courses that push specialized students to be broader, or general requirements that force every graduate to know at least something about certain subjects.”

The unifying principle of the liberal arts and computer science is a Christian worldview. Theology is the queen of the sciences which permeates and connects all of the liberal arts together in a unified whole. Far from being fragmented, the liberal arts are integrated under the umbrella of God’s Word. Christianity is astonishingly holistic in that it touches every aspect of a person’s being: body, mind (psyche), and spirit (soul).

**Why Study Computer Science?**

Computer science provides practitioners with vast job opportunities; there are hundreds of thousands of open computer science positions currently in the US. Computer scientists are in demand. But computer science is not just for geeks. Everyone needs to study and understand computer science principles. All

---

12 https://india.blogs.nytimes.com/2012/02/01/choice-on-india-ink-liberal/
technology is based upon computer science concepts. While it is possible to merely use technology without understanding the underlying concepts, it is impossible to be an effective problem-solver without such knowledge. Of course there are many employment opportunities for computer science professionals, and those positions pay well. At the same time someone who does not want to be a computer scientist can still benefit from studying computer science. The tools and processes used in computer science are helpful for everyone, regardless of their discipline.

Because of abstraction, technological tools can be used by anyone. Abstraction is the technique of hiding details and focusing on the important aspects. The science of abstraction is employed to hide the messy details of the technology from the user. Few people understand how their cell phone actually communicates with the cellular network because the science of computing has abstracted the process thus making it simple and accessible to anyone. However, if something goes wrong, an uninformed user is lost, unable to continue or resolve the issue.

Some argue, “Why should I understand my car just to drive it? I don’t have to know how the engine works, I just turn the key and go.” This is false, which is clearly seen when something goes wrong. In order to understand the problem, both knowledge and wisdom are required. If the oil light flashes on the dash and the user has no understanding of the concept, the engine will soon fail if not shut down. Understanding concepts is different than understanding the specifics of technology. The ABS (anti-lock braking system) found on a modern automobile is a complex collection of actuators, pumps, sensors, processors, and connecting equipment. No user needs to understand the physics involved in the hub sensor or the hydraulic principles of the brake fluid actuators. However, a user must understand the concept of ABS. Surveys show that many people don’t know how to stop a car in a panic braking situation. When the brakes are applied in a panic stop, the system monitors and prevents skidding. It does this by
rapidly applying and releasing the brakes (in the “old days” drivers had to “pump the brakes” to avoid skidding out of control). When active, the brake pedal on an ABS car will pulsate, kick back, and make noise. Many drivers are unprepared for this response and will release the brake when it happens. Of course, if the brake pedal is released the car will not stop. A driver does not need to understand the exact technology used in ABS, but he must understand the concepts of ABS in order to be an effective driver.

Steve Jobs remarked, “Computer science is … something everyone should know how to use, at least, and harness in their life. It’s not something that should be relegated to 5 percent of the population over in the corner. It's something that everybody should be exposed to and everyone should have mastery of to some extent, and that's how we viewed computation and these computation devices.”

It is hard to imagine a vocation that does not use computer systems, or could use technology to provide productivity and efficiency advantages. Because almost every job requires the use of computers and technology, employees must be proficient in their application. Consider a simple tool like a word processor. While one does not require a training program to use a word processor, without understanding the concepts of word processing a user is much less effective. Many users will continually type the same long word or phrase (such as “computer science”) into a word processor during data entry. A user who understands the concept of the “replace” command will enter abbreviations (such as “cs”) and then later issue a command to expand the abbreviations into full words using the ‘replace’ command. Understanding the concepts of word processing allows the user to be much more efficient.

---

13 http://www.npr.org/2011/10/06/141115121/steve-jobs-computer-science-is-a-liberal-art
Computer science is problem solving. Many would argue that the problems solved by computer scientists are some of the most complex and difficult imaginable. The same concepts and techniques used in the computer science problem-solving process can be applied to solving problems in other disciplines. Thus, a person who is educated in computer science is able to solve problems more effectively in his own field. Understanding computer science improves problem-solving skills and abilities.

Problem-solving involves deep analysis of problems and careful synthesis of solutions. Analysis and synthesis in computer science demand clear and logical thinking. At the heart of computer science are nine grand ideas. These are the concepts of algorithms, abstraction, automation, intelligence, interface, information, cognition, complexity, and creation. These ideas permeate the discipline and must be understood by practitioners and effective users of technology.

Abstraction, the ability to simplify complexity, is a key to effective problem-solving. Dr. Don Knuth refers to abstraction as the essence of the “computer science perspective.” In his book, Things a Computer Scientist Rarely Talks About, Knuth states, “One of the main characteristics of a computer science mentality is the ability to jump very quickly between levels of abstraction, between a low level and a high level, almost unconsciously.”

Computer science teaches you to think well. There is a connection between thinking well (and critically) and freedom, both of which are goals of a liberal arts education. At the same time, computer science is fun. Computer science is a creative discipline which develops and expands one’s imagination. Computer scientists create informational entities which assist others in solving problems. Computer scientists work with informational, rather than tangible, entities; therefore our creations

---

14 Donald Knuth, Things a Computer Scientist Rarely Talks About, Center for the Study of Language and Information, 2003
are liberated from the physical domain and our imaginations are freed from normal limits. This is fun since we are creating with “thought stuff!”

The skills and concepts of computer science are vital for everyone to understand. Not only do computer science concepts help everyone function effectively with technology, these skills and concepts are transferrable to other disciplines. The computer science problem-solving mindset is an effective framework for approaching problems in any field of endeavor. And since most fields employ technology, everyone can benefit from understanding the disciplined approach to problem-solving utilized in computer science.

Computer Science as a Liberal Art

Apple founder Steve Jobs declared, “In my perspective ... computer science is a liberal art.”

Computer science is the modern analog to astronomy in the quadrivium of the classical liberal arts education. After students learned the trivium, the quadrivium allowed them to apply knowledge in specific areas. The four arts of the quadrivium studied numbers in a specific context (arithmetic, geometry, music, and astronomy). Valerie Locklair stated, “The quadrivium carried with it an idea that foundation should precede specialization. One of the dangers of an increasingly technophilian culture is that we are inundated by new technology every day. We become enchanted by the end result—a flashy new application—and too quickly forget (or never understand in the first place) the concepts behind it.” The purpose of astronomy was to study numbers in space. Computer science deals with numbers, but more importantly it

---

15 http://www.npr.org/2011/10/06/141115121/steve-jobs-computer-science-is-a-liberal-art
16 personal correspondence with the author, December 2017
studies how to solve problems using numeric data. Knowing computer science helps you learn about anything!

There is a misconception that computer science is the study of computers. Actually, computer science is the study of problem solving. A computer scientist doesn’t study computers, but he does use computers as tools to understand problem solving. In a similar fashion an astronomer does not study telescopes. Instead, a telescope is used as a tool to understand the numeric precision of God’s created cosmos.

Computer scientist Dr. Brian Kernighan said that educated citizens need to understand “some of the tools of the trade that will make it possible for them to think intelligently about this technology for themselves.” Like Steve Jobs and Don Knuth, Brian Kernighan believes that computer science deserves a place in general education and should have a central role as a liberal art.

To understand the discipline of computer science requires a liberal arts approach. A student must master the grammar, logic, and rhetoric of computer science in order to fully comprehend it. Otherwise the student is merely focusing on the tool (computer) and not on the purpose of computer science. Technology changes constantly. The underlying concepts of computer science do not. The liberal arts form a framework for learning by progressing through the theory, practice, and application of any discipline.

Computer science isn’t a new discipline. I trace the historical roots back 400 years to the work of the Lutheran professor Wilhelm Schickard and to the Christian thinker Blaise Pascal. Since then, many brilliant people have helped bring the discipline into focus. These include George Boole, Charles Babbage, Lady Ada, Grace Hopper, and a host of others. All of these founders were liberal arts people. In order to lay the foundation for computer science these polymaths not only were

learned in many areas of the liberal arts, they were able to integrate their knowledge and focus their energies in new, creative ways. None of them were geeks or nerds. The individuals who created the discipline of computer science did not have degrees in computer science. They were liberal arts thinkers who could analyze the problem and synthesize a solution while integrating a number of seemingly disparate subjects. ⑩

Goal

The goal of a liberal arts education is to equip free citizens who are able to love and serve their neighbors well in their vocations. Computer scientists solve problems for other people. A computer system is a tool which enables people to solve problems effectively. Computer scientists do not solve their own problems; instead they reach out to their neighbors and solve problems using the tools of computer science. A computer scientist will expend much effort in analyzing and understanding the user’s problem (the problem domain) before designing a system to solve it. In the process, the computer scientist will work closely with the user and develop empathy and understanding of the user’s situation.

The liberal arts equip students to think critically by investigating relationships and dependencies between subject areas. Analyzing a problem requires understanding the breadth of the subject and how it interacts with different spheres of knowledge.

Computer science focuses on seeing and making connections between distinct areas in order to create algorithms which solve the specific problem. To critically investigate requires understanding all of the aspects and relationships involved. Any creative act is Trinitarian, requiring idea, implementation, and interaction. A computer scientist must analyze the user’s problem to formulate an idea of the algorithmic solution. In order to

implement the required system, the computer scientist must synthesize hardware and software into a viable product. The user must interact with the product and solve specific problems for the solution to be deemed successful.19

The liberal arts are designed to help students understand and deal with complexity. The computer science mindset of abstraction is a primary tool for peeling back layers of complexity, allowing one to focus on the central core and not be bogged down with details. Anyone can apply this computer science mindset to problems in any discipline and become a better problem-solver. Computer science is a mindset as much as it is a subject. It is a liberal arts mindset which also seeks to integrate and connect in order to analyze problems and synthesize solutions.

If “art expresses knowledge while science uncovers knowledge,” computer science does both! There are both science and engineering (“art”) in computer science. A computer scientist must understand the theoretical and conceptual aspects of algorithms, the science of computer science. In addition, a computer scientist will build and engineer systems, which is an artistic endeavor.

Conclusion

Understanding computer science is mandatory for a free citizen to be truly educated. All vocations can benefit from the use of technology which allows problems to be efficiently solved. The technology that we all use is based upon computer science. In addition, computer science is a way of learning and seeing, which reflects the liberal arts. Computer science is useful for any major as it is a mindset, a way of solving problems, that benefits everyone. Computer science reflects the liberal arts approach to learning, which has stood the test of time and proven its worth.

Abstract

Cyber-attacks, which are attempts to damage or destroy computer systems and networks, are a concern for both organizations and individuals. Since the first, rather simplistic attacks in the 1980s, cyber-attacks have been increasing in both frequency and complexity. Studies have determined that there is a financial impact of cyber-attacks on U.S. organizations (Luse, 2009; Yayla & Hu, 2010). While some organizations and individuals are prepared to detect and defend themselves from attacks, many are not. Proper defense against cyber-attacks is accomplished in three areas: software, hardware, and people.

Defense against cyber-attacks can be implemented in either hardware, software, or a combination of both. Focusing on hardware, one type of hardware defense mechanism is Intrusion Detection Systems (IDS). A challenge for IDS systems is the speed of detection (detection latency) as systems can become overloaded if more data needs to be processed than the system is capable of handling. Cyber-attackers are aware of this problem and have design attack scenarios that exploit this limitation. If the latency is significant, attackers may have enough time to damage the target system. Faster detection and increased sensitivity in intrusion detection systems are desirable.
Introduction

Many methods have been developed to detect, prevent, and minimize the damage caused by cyber-attacks (Thangavel, Thangaraj, & Saravanan, 2010). Among the hardware and software solutions, are firewalls, intrusion detection systems, and intrusion prevention systems. Intrusion detection systems (IDS) are software or hardware-based systems that are designed to detect and monitor intrusions as they occur, to prevent exploitation (Alaidaros, Mahmuddin, & Mazari, 2011). Basically, Intrusion Prevention Systems work to stop an attack outside of the organization and Intrusion Detection Systems work to detect and flag cyber-attacks on the systems themselves.

Two main classifications of cyber-attacks are targeted and non-targeted attacks. Targeted attacks are directed to impact a particular organization or a particular type/brand of equipment. Non-targeted attacks do not have a specific target but are generic in nature and are opportunistic. If your systems are running the operating system or software that is being attacked, it is at risk. As such, non-targeted attacks may result in damage to any organization that does not have the correct defenses in place if they happen to be using the desired environment.

Information was lacking to show if damage resulting from cyber-attacks is higher as a result of current intrusion detection system latency than if intrusion detection systems with lower latency were available. In order to test cyber-attacks against IDS systems, a dedicated test environment needed to be developed. The goal of these tests was to extend research into the relationship between IDS detection latency and
the severity of cyber-attacks. This knowledge could lead to the development of intrusion detection systems with lower latencies if warranted.

The inherent value of data to organizations, governments, individuals, and the military has led to the need for improved methods of blocking cyber-attacks (Owens, et al., 2009). The need for protecting information is so important to organizations that security has become vital to protecting the economy and safety of society (Patel, Taghavi, Bakhtiyari, & Celestino, 2013).

Intrusion detection systems (IDS) are a method used to detect cyber-attacks and to limit the amount of damage that can be done by a cyber-attack. Intrusion detection systems may require a period of time between the start of an attack and the recognition that it is occurring. This is known as detection latency. The speed of detection (detection latency) is a key challenge in Intrusion Detection System (IDS) design (Mitchell & Chen, 2014). If detection latency is long enough, attackers may have enough time to damage the target system before detection occurs (Mitchell & Chen, 2014). Faster detection and better sensitivity in intrusion detection systems are desirable (Fadlullah, Nishiyama, & Kato, 2013) as increased sensitivity would lead to improved detection, which in turn can lead to safer organizations.

Knowledge of the potential impact of cyber-attacks is beneficial to organizations and governments for allocating security budgets and to ensure that the optimal security protection is purchased.
One Method for Designing a Test Environment

A goal of developing a test infrastructure is to ensure that all tested hardware and software are run in a consistent environment and that the tests can be easily duplicated by others. To help with the accuracy and repeatability of testing, all of the intrusion detection systems were tested on a single system running the same operating system, software, and workload throughout the testing process. By using a single test platform, latency timings were comparable across intrusion detection systems.

The test platform used for initial testing was an Apple Mac Pro with a 3.7 GHz Quad-Core Intel Xeon E5 processor and 16 Gb of memory. This test platform is running Mac OS X 10.10.3. VirtualBox from Oracle Corporation was used to run an emulation of the Security Onion Xubuntu distribution as a virtual machine on top of the Mac operating system. Security Onion is a Linux distribution for intrusion detection, network security monitoring, and log management. This OS runs on top of the base Mac OS X. A wired Ethernet connection was used for consistency and to minimize network volatility associated with wireless networking. Limiting a test system to a single use as a dedicated test platform is helpful for objective testing.

The Security Onion Linux distribution (www.securityonion.net) is designed for testing intrusion detection and for monitoring computer networks. It is an open source toolset and contains the following tools: Snort, Suricata, Bro, OSSEC, Squil, Squert, ELSA, NetworkMiner and other tools.
Once the hardware and operating system(s) have been selected, the next goal was to locate or capture network traffic to run through the systems. For ease of testing, the use of publically-available packet capture files is very beneficial. Using these tools and publically-available PCAP (packet capture) files. For the initial testing for my research, Snort, Suricata, Bro and OSSEC were used to detect intrusions in the PCAP files. Depending on the type of tests being conducted, additional SQL queries may need to be written.

This testing infrastructure was used to examine the impact of cyber-attacks organizations and to determine if longer latency periods result in larger impacts to these organizations than if intrusion detection systems with lower latency were available. The topic of cyber-attacks is of repeated interest in practitioner literature, the media, and scholarly sources as a result of the increasing number of attacks and the increasing damage.

Intrusion detection latency is rarely reported but is an important metric as part of the efficiency of intrusion detection systems. The detection of all attacks is important but not useful if excessive time is needed to report on intrusions (Mitchell & Chen, 2014). Reduced latency and limited resource consumption are considered desirable goals for detection engines (Zhang, Ho, & Naït-Abdesselam, 2010). This test infrastructure was developed to allow research into cyber-attacks and to increase understanding of the impact of intrusion detection system’s latency.

Several problems exist with current intrusion detection systems and since these problems are widely known, hackers have taken advantage of these weaknesses.
Challenges for Intrusion Detection Systems

Existing IDS systems can become overloaded with the sheer volume of network traffic that needs to be examined as network speeds increase and as the volume of traffic increases. This means that it becomes potentially impossible for an IDS system to examine each individual packet of network traffic at speeds of multiple Gigabits per second (Gbps) (Alaidaros, et al., 2011). An example of IDS systems becoming overloaded by traffic occurs in a denial of service (DoS) attack, which floods network equipment with enough traffic to overwhelm the system and block legitimate network traffic (Hahn & Layne-Farrar, 2006). One of the challenges in high-volume systems is that NIDSs may not be able to capture and examine every packet that is sent across a network (Alaidaros, et al., 2011). The approach is taken by hackers to either overload a system or to attempt to slip dangerous packets through.

An additional complication is that up to 80% of messages sent across networks are small in size (less that 200 bytes) so there can be a high volume of small messages which can lead to a system being overloaded (Ben Fredj, et al., 2010).

Need for Speed

The speed of detecting and responding to cyber-attacks is important. William Lynn noted that in cyber-attacks, a margin of milliseconds can make a difference (Farwell, 2012). Latency in detection can potentially mean that damage can be done before the attack is detected by the IDS system. The costs associated with cyber-crime are difficult to determine but a 2011 survey determined that 90% of U.S. businesses had experienced
at least one cyber security breach during the previous 12 months (Ponemon Institute) “and 41% of the businesses spent at least $500,000 to address the problem” (Anonymous, 2011, p. 8).

Organizational Preparation

Organizations have various levels of preparedness to prevent and deal with cyber-attacks and there are many varied reasons why organizations may not be as prepared as possible. Some of these reasons include a lack of knowledge, funding limitations, and complacency. Research indicates that it is sometimes difficult to get companies to invest in cyber security as it is a challenge to show a relationship between lost customers and lost business following a cyber-attack (Ryan, et al., 2012).

Complacency can set in if a period of time passes and an organization has not experienced attacks in recent history. This can lead to organizations cutting their cyber security budget since they have not seen a recent security breach. If an organization has not detected any attacks, they may still have experienced one though it remained undetected giving the attackers the opportunity to observe activity on the system (Towns, 2012).

Organizations may not place an adequate value on cyber-attack security and develop the correct infrastructure, as there is limited research into the effects that cyber-attacks have on the current and future market value for organizations (Crowther & Haimes, 2010).

Measuring and Testing Intrusion Detection Systems

Research has indicated a need for increased metrics on intrusion detection systems and specifically testing of the effects of latency in intrusion detection systems. The
challenges and costs of testing intrusion detection systems that produce reliable results, may have resulted in few tests of this nature being performed (Lippmann, et al., 2000). The process of measuring and testing intrusion detection systems is further complicated by their position within a network. Results vary depending on where the devices are physically located and this can affect the number of alarms detected. Their position in relation to other security tools has an effect on the overall performance (Elfeshawy, et al., 2012).

Obtaining Test Data

There are several ways of obtaining test data. The first is to perform “live” monitoring of traffic on an existing network. The challenge with this approach is that the testing can have a negative impact on the network. A second approach is to capture a stream of network traffic and save it for later analysis. Lastly, secondary data in the form of existing intrusion detection datasets (existing PCAP files) can be used. The PCAP files used for initial testing were obtained from publically available datasets gathered by NETRESEC AB (http://www.netresec.com/), an independent organization and website based in Sweden. NETRESEC focuses on network forensics and the analysis of network traffic. Packet capture files contain operational network traffic from network users (Ho, et al., 2012). PCAP files are recordings of network traffic that can be replayed against intrusion detection systems to test their responsiveness and effectiveness. An advantage of using this type of data is that the data reflects authentic network traffic and
authentic cyber-attacks, as opposed to using simulated data sets.

The PCAP files were then replayed using TCPREPLAY and each of the four selected intrusion detection systems were used to detect attacks. Figure 1 depicts an overview of the test environment and the testing steps used. The test hardware was running the Security Onion virtual machine on top of the existing Mac OS X operating system. The PCAP files were split into smaller files using the TCPDUMP command. TCPREPLAY is a UNIX command used to replay PCAP files. The TCPDUMP and TCPREPLAY commands are shown in Table 2.
Table 1. UNIX Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Sample Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCPDUMP</td>
<td>tcpdump -r snort.log.1425823273 -w splits/snort.log.1425823273a -C 82</td>
</tr>
<tr>
<td>TCPREPLAY</td>
<td>tcpreplay --topspeed --intf1=eth0 /home/security/pcaps/splits/snort.log.$1 &amp;</td>
</tr>
</tbody>
</table>

Figure 1. Testing Environment
Conclusion

This paper described the creation of an IDS test environment and the test methodology used to test network capture files. The importance of intrusion detection and the challenges in testing were detailed. No single solution exists to adequately deal with cyberattacks and this paper presents one way to perform testing. A multi-faceted approach appears to be the best solution for dealing with this problem. A combination of technology, policies, legislation, and training is needed to fend off cyber-attacks and Intrusion Detection Systems are a piece of this solution.
References


Giant Thinking… Minds?
The Problem of Strong Artificial Intelligence
by
Valerie A. Locklair

Abstract
Artificial Intelligence is an increasingly popular concept in our world today, and not just in the technological sector. Growing concerns over a robot apocalypse - either in the career force or in a catastrophic global event - haunt everyone from academics to job seekers to philosophers. Do we have anything to fear from so-called "strong AI"? Are we headed towards a civilization of giant thinking minds, or have we inadvertently followed the wrong logical path? By examining artificial intelligence through the lens of theistic dualism, we can prepare ourselves for the technological advances of tomorrow - and perhaps come to some surprising conclusions about what it means to be human today.

Introduction
The subject of this book is a type of machine that comes closer to being a brain that thinks than any machine ever did before 1940. These new machines are called sometimes mechanical brains and sometimes sequence-controlled calculators and sometimes by other names. Essentially, though, they are machines that can handle information with great skill and great speed. And that power is very similar to the power of a brain.20

It has been over sixty-five years since Edmund Callis Berkeley popularized the concept of computers as “Giant Brains or Machines that Think.” The newfound and rapidly expanding computing powers of the mid twentieth century have only grown exponentially over the past six and half decades. Expert systems, wearable electronics, and chess playing computers, once relegated to the B-level blockbuster crowd, now make headlines the world

over and spark debates over the future of artificial intelligence. Is the power of these iconic machines “very similar” to the power of our own brains, as Berkeley believed? Can machines think, and does the answer to this question matter to the current discussions in the philosophy of mind?

Different aspects of philosophy have attempted to answer this question. This paper will present a dualistic, theistic philosophical response and highlight the incompatibilities that this worldview presents with so-called strong artificial intelligence.

Defining Terms

Before we delve headlong into the controversy over thinking machines, it would be helpful to define a few key terms. “Artificial intelligence,” though coined in relatively recent history, is a term whose definition is a point of contention among computer scientists and philosophers.21 The definition we will use for this paper is one of the less controversial explanations, credited to M.L. Minsky: “artificial intelligence is the science of making machines do things that would require intelligence if done by men.”22 This is typically carried out by means of computer programs, or software application packages designed to fulfill a specific task. This is accomplished by means of programmers creating code, or instructions written in an intermediary language, that can then be translated into the language of binary (1s and 0s) that machines are capable of manipulating. The binary nature of this machine language is such that a machine operates on the basis of symbol manipulation, or of translating instructions into “on-off” cues. Computers, far from being mere number crunchers, are general purpose. As Margaret Boden says, “Computers can crunch numbers if specifically programmed so to do, but this is not their essential computational function.”23 She uses this concept of a

---

21 The term was coined by John McCarthy in a 1955 proposal for a Dartmouth Summer Research Project. The text of the proposal is available online at <http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html>


23 Ibid., 16.
computer as a general purpose machine to drive her definition of artificial intelligence:

In sum, artificial intelligence is the use of programs as tools in the study of intelligent processes, tools that help in the discovery of thinking-procedures and epistemological structures employed by intelligent creatures.\(^{24}\)

This does not seem like a very controversial statement, and indeed, were we to stop there with that definition we would bypass the majority of controversy over strong artificial intelligence. However, we are interested in the adjective modifying our term. What, exactly, does *strong* artificial intelligence mean?

Strong artificial intelligence takes the definition of standard AI one step further, as John Searle explains, because it says “the brain is just a digital computer and the mind is just a computer program,” and that

On this view, any physical system whatever that had the right program with the right inputs and outputs would have a mind in exactly the same sense that you and I have minds… Any system whatever that is capable of manipulating physical symbols in the right way is capable of intelligence in the same literal sense as human intelligence of beings.\(^{25}\)

Note again how the manipulation of symbols comes up in this definition, too, but with the crucial addition that this is a sufficient characteristic to justify labeling an entity as “intelligent.” On this view, technology will advance – or, as some believe, has already advanced – to the level of equivalence with the human mind. John

\(^{24}\) Ibid., 17.

\(^{25}\) John Searle, *Minds, Brains and Science* (Cambridge: Harvard University Press, 1984), 28, 29. Strict materialism is also sometimes referred to as “physicalism.” For the sake of consistency, this paper will use the term “materialism.”
McCarthy, for example, who himself coined the term “artificial intelligence,” purported that “machines as simple as thermostats can be said to have beliefs.”

How does one arrive at this claim, and is it a sound stance to take? Berkeley, Searle, and Boden highlight the central issue of the controversy over strong artificial intelligence that this paper will discuss: are these “mechanical brains” called computers capable of thought? Do we really live in an age of automated, thinking minds?

The Evolutionary Mind behind the Materialist Curtain

Although it may be difficult to see on the surface, the concept of strong artificial intelligence or “AI” is supported by a few key areas in the philosophy of mind. Most notably, strong AI is compatible with strict materialism, which in turn influences the concept of evolutionary psychology. We will deal with these two issues briefly to highlight their connection to the giant, thinking machines in question.

Strict materialism is a form of naturalism, which, to quote neuroscientist Mario Beauregard, is “the philosophy that nature is all that exists and everything has a natural cause.” Naturalism is exclusively monistic; that is, it says that there is only one type of substance that exists – whatever type it may be – and that everything can be explained in terms purely relating to that classification. To take it one step further, materialism insists that “everything that exists is physical or material in nature,” and “maintains that the single, all-embracing temporal system contains nothing but the entities recognized by the most mature physics.”

How did these physical entities come into being, and how does this relate to intelligence? For strong AI advocates like Ray Kurzweil, the model of evolution is an irrefutable axiom:

26 Ibid., 31.


Here’s another critical question for understanding the twenty-first century: *Can an intelligence create another intelligence more intelligent than itself?*

Let’s first consider the intelligent process that created us: evolution.

Evolution is a master programmer. It has been prolific, designing millions of species of breathtaking diversity and ingenuity. And that’s just here on Earth.\(^{29}\)

For Kurzweil, evolution is the perfect example of an intelligence creating something more intelligent than itself. He praises evolution’s creative powers, but he also highlights one of its fatal flaws: its incorrigible slowness. If we take time to completion into consideration, Kurzweil ranks evolution’s IQ “only slightly greater than zero,” or just above entropy on the intelligence totem pole: “evolution is thereby only a quantum smarter than completely unintelligent behavior.”\(^{30}\) Yet evolution created human brains and therefore, it is assumed, intelligence, and it is this human intelligence that is “far more intelligent than its creator,” as Kurzweil summarizes,

And so, too, will the intelligence we are creating come to exceed the intelligence of its creator… The human species creating intelligent technology is another example of evolution’s progress building on itself. Evolution created human intelligence. Now human intelligence is designing intelligent machines at a far faster pace. Yet another example will be when our intelligent technology takes control of the creation of yet more intelligent technology than itself.\(^{31}\)

How, exactly, did we become endowed with this evolution-given intelligence? The answer for materialists lies in evolutionary psychology.


\(^{30}\) Ibid., 44.

\(^{31}\) Ibid., 47.
As its name implies, evolutionary psychology states that the human (and ape) brain comprises many functional mechanisms, called psychological adaptations or evolved psychological mechanisms, that evolved by natural selection to benefit the survival and reproduction of the organism.  

Everything about us as humans – our likes, dislikes, so-called beliefs, desires, and intelligence – is the product of our creator, evolution. At one time these characteristics aided us in survival, and through the painfully slow process from approximately 190,000 years ago to the present, our traits have been modifying and, presumably, our intelligence increasing. The brain, supposedly the seat for the intelligent mind, has developed slowly over hundreds of thousands of years, and now the smarter, stronger machine will take over for us. As software pioneer Bill Gates said,

I don’t think there’s anything unique about human intelligence. All the neurons in the brain that make up perceptions and emotions operate in a binary fashion.

If evolution could create intelligence over a long period of time, couldn’t the resulting intelligence create an even more intelligent being given less time, especially if the projected being is a machine that already processes data the same way that the human brain does? Not everyone agrees with this assessment, as we shall see, and one of the strongest criticisms of strong artificial intelligence comes from the realm of dualism.

---


33 Ibid. The 190,000 follows from the theory that “Mitochondrial Eve,” the mother of every living human being, lived between 190,000 and 130,000 years ago.

34 Ibid., 23.
Two is Better than One: The Dualistic Argument against Strong AI

Dualism, as opposed to monism, is “a philosophy that accepts the coexistence of fundamentally different entities (e.g., matter and mind)."\(^{35}\) That is to say, dualism recognizes that there are at least two separate types of entities that make up a human being, typically seen as brain and mind or soul. There are many scientific arguments for dualism, but for the sake of space we shall only examine one that neatly highlights the issue of strong AI: subjectivity.

If the mental can be shown to have different properties than merely physical, then materialism would seem to be an insufficient explanation for the human mind and intelligence. Dualistic philosophers see the inherent subjectivity of mental occurrences as evidence of just such a property difference. For example, “thoughts seem to be about things in the world,” and it is possible to “think about the bowl of radishes sitting on my dining room table, and this thought is about the radishes.”\(^{36}\) Thoughts have intentionality, that is, they are “about other things in the world,” unlike the radish which does not seem to represent or be about anything else.\(^{37}\) My thought about the radishes, likewise, will not be exactly the same as your thought about the odd vegetables or the awe of an Egyptian sky at dusk. My qualia, or the “something it is like” to experience a particularly appealing radish or the fierce beauty of a Middle Eastern sunset, is subjective to myself and my own experience – there does not, for example, appear to be “something it is like” for a camera to look at the same thing and have that same experience.\(^{38}\)

John Searle, professor emeritus of philosophy at the University of California, Berkeley, addresses subjectivity and

\(^{35}\) Ibid., 344.

\(^{36}\) Barbara Montero, *On the Philosophy of Mind* (Belmont: Wadsworth, 2009), 34.

\(^{37}\) Ibid.

\(^{38}\) Ibid.
intentionality and applies them to the world of artificial intelligence when he says,

There is more to having a mind than having formal or syntactical processes… Even if my thoughts occur to me in strings of symbols, there must be more to the thought than the abstract strings, because strings by themselves can’t have any meaning. If my thoughts are to be about anything, then the strings must have a meaning which makes the thoughts about those things. In a word, the mind has more than a syntax, it has a semantics.\(^\text{39}\)

Why is intentionality – what Searle refers to as semantic meaning – incompatible with strong artificial intelligence? If we remember our earlier definition of strong artificial intelligence as manipulating symbols in the correct way to produce intelligence equivalent to that of a human, we can see an issue forming. According to Searle, computer operations must be

specified purely formally; that is, we specify the steps in the operation of the computer in terms of abstract symbols – sequences of zeroes and ones printed on a tape, for example… the symbols have no meaning; they have no semantic content; they are not about anything. They have to be specified purely in terms of their formal or syntactical structure… this feature of programs, that they are defined purely formally or syntactically, is fatal to the view that mental processes and program processes are identical.\(^\text{40}\)

As Searle points out in his famous Chinese room thought-experiment, even if a computer can be thought to be performing a task in an intelligent manner – in essence passing the infamous Turing test – it would not mean that it is intelligent or that it has


\(^{40}\) Ibid., 30, 31.
any of the subjective understanding necessary for a dualistic view of mind and intelligence.41

Rise of the Living Machines?
Could humans ever create artificial intelligence that would surpass man’s greatest mental feat? From a strictly materialistic view, there is no reason to disbelieve that in ten, twenty, or thirty years we will be surrounded by Commander Datas (à la Star Trek: The Next Generation) that will come fully equipped with emotion chips and supra-human brain capacity. If, on the other hand, thoughts, intelligence, and the semantics of the human mind cannot be reduced to purely physical causes, we appear to be at an impasse.

The great allure of strong artificial intelligence research appears to be creating man in his own image, only stronger, smarter, a super-powered force to benevolently carry on our gene pool and secure immortality.42 What exactly does it mean to create something more intelligent than ourselves if, as the evidence suggests, our intelligence is not reducible to the purely material? What if the evidence suggests that evolution, far from being Kurzweil’s irrefutable axiom, could perhaps be false altogether?43

41 Ibid., 31-32. Searle’s Chinese room experiment in which symbol manipulation of Chinese characters by someone who does not comprehend the language is an excellent example of what it means to understand and have a subjective mental state about something. The Turing test is an experiment to see if an AI machine can trick a human user into believing that the machine is in fact a thinking, living human via intelligent interactions – c.f., David Barker-Plummer’s explanation in "Turing Machines, " The Stanford Encyclopedia of Philosophy (Summer 2013 Edition), Edward N. Zalta (ed.).

42 In a by no means atypical example of modern AI optimism, a June 2013 headline in the Daily Mail newspaper proclaimed “We'll be uploading our entire MINDS to computers by 2045 and our bodies will be replaced by machines within 90 years, Google expert claims.” The Google expert to which the headline refers is none other than strong AI advocate Ray Kurzweil.

43 For the sake of space this fascinating argument will not appear in this paper. For additional resources on contemporary research into the evidence against evolution, see the Creation Research Society Quarterly Journal.
What if it was not the intelligent, mythical all-father of evolution that created us, but a Being infinitely higher than ourselves?

“Tyger, tyger, burning bright,” writes the poet William Blake,

In the forests of the night;
What immortal hand or eye,
Could frame thy fearful symmetry?

What the hammer? what the chain,
In what furnace was thy brain?
What the anvil? what dread grasp,
Dare its deadly terrors clasp?44

Before we can speak of creating machines superior to our own holistic human experience, we must understand who we are and what it would take to create ourselves. Are we material? Are we made of purely physical “star stuff,” or are we created by the One who formed the stars?45 Would we need something even more fearfully intelligent to create us? As the saying goes, how can we expect the gods, those questions and thoughts just outside of our feeble reach, to “meet us face to face till we have faces?”46

The evidence for dualism is striking for many reasons, not the least of which for the fact that it is largely ignored in contemporary debates about strong artificial intelligence. Theism, far from being outdated psychobabble, has much to say on the key issue confronting strong AI today: what is intelligence, and can we create it and jettison evolution to version 2.0? Perhaps thinking minds like Kurzweil’s, if he truly believes that “intelligence is the ability to use optimally limited resources—including time—to achieve [goals],” should begin to use that precious time and to

44 William Blake, “The Tyger,” 1, 4.


listen not for the first signs of computer consciousness, but for the truth about humanity and its future.\textsuperscript{47} Perhaps the answer to the fate of our universe does not need to be “intelligently consider[ed] when the time is right” and executed via mind-uploading to a cosmic network, but rather has already been settled by the ultimate Intelligence, who in infinite wisdom chose to limit our sub-creative abilities and endow us with thoughts that can consider a created universe and other minds as the echoing evidence of irrefutable Intelligence.\textsuperscript{48}

\textsuperscript{47} Kurzweil, \textit{The Age of Spiritual Machines}, 73.

\textsuperscript{48} Ibid., 258.
References


Using ASICs to Train Neural Networks
by
Timothy R. Wildauer

Abstract
Artificial Neural Networks (ANNs) are being used to solve a diverse range of problems, from determining election results (Borisyuk et al., 2005) to facilitating speech recognition (Hinton et al., 2012). A vast amount of the computing power required to run ANNs comes from training the network. Millions of training examples are run thousands of times. Depending on the size of the network, this can take days to months. The same set of calculations are used in each training example. Graphics Processing Units (GPUs) are typically used for these calculations. Application Specific Integrated Circuits (ASICs) are used to solve a sequence of calculations in a single clock cycle. ASICs could be used in place of GPUs to speed up the calculations necessary to train a network. However, depending on the size and connectedness of a network, this substitution may not be cost effective.

Introduction
Shown below is a simplified fully connected neural network, showing two input nodes, two hidden nodes, and two output nodes. The weights between them are labeled $w_1$ through $w_8$.

Figure 1: A fully connected network with one hidden layer. Each layer has two nodes.
To calculate the activation of the hidden or output nodes, the output of the nodes in the previous layer is multiplied by the weight from that node to the node in question. All of the weighted outputs are summed, and the result is passed into an activation function. In this paper we will use the Sigmoid Function:

\[ f(x) = \frac{1}{1 - e^{-x}} \]

This function maps negative inputs to zero, and gradually increases to one for positive inputs. The output of this function is the activation of each node.

The effectiveness of a network is based on cost function. The cost function is the sum of the squared errors of each of the output nodes when the output is compared with the expected output:

\[ C(X) = \sum_n \frac{1}{2} (o_n - \delta_n)^2 \]

**Backpropagation**

To train the network, we need to find the combination of weights which minimize the cost function \( C(X) \). To do this, we need to find how the cost function changes with respect to each weight. For example, we can find the change in the error with respect to the change in \( w_5 \). This is done with a partial derivative which can be expanded to read

\[
\frac{\partial E_{\text{total}}}{\partial w_5} = \frac{\partial E_{\text{total}}}{\partial \text{output}_{o_1}} \times \frac{\partial \text{output}_{o_1}}{\partial \text{input}_{o_1}} \times \frac{\partial \text{input}_{o_1}}{\partial w_5}
\]

The partial derivative of the error with respect to \( o_1 \) is the partial derivative of the cost function with respect to \( o_1 \), which is \(-(o_1 - \delta_1)\). The partial derivative of \( o_1 \) with respect to the input of \( o_1 \) is the partial derivative of the sigmoid function, which is \( \text{output}_{o_1} (1 - \text{output}_{o_1}) \). The partial derivative of \( \text{input}_{o_1} \) with respect to \( w_5 \) simplifies to \( \text{output}_{h_1} \). Thus,

\[
\frac{\partial E_{\text{total}}}{\partial w_5} = -(o_1 - \delta_1) \text{output}_{o_1} (1 - \text{output}_{o_1}) \text{output}_{h_1}
\]
If we simplify
\[
\frac{\partial E_{total}}{\partial output_{o1}} \times \frac{\partial output_{o1}}{\partial input_{o1}} = \delta_{o1},
\]
then
\[
\frac{\partial E_{total}}{\partial w_5} = \delta_{o1} output_{h1}.
\]

This can be generalized to say
\[
\delta_x = \sum_n \frac{\partial E_{total}}{\partial w_n} \times output_x (1 - output_x)
\]
where \( w_n \) is a weight that goes from node \( x \) to any node in the next layer. Furthermore,
\[
\frac{\partial E_{total}}{w_n} = \delta_{x2} output_{x1}
\]
where \( w_n \) is the weight from node \( x_1 \) to node \( x_2 \).

Once the change in \( w_n \) is determined, that weight is adjusted,
\[
w_{n_{new}} = w_{n_{old}} - \alpha \frac{\partial E_{total}}{w_n},
\]
where \( \alpha \) is a preset learning rate. This adjustment is made for all the weights in the network. Afterwards, the network is used to determine the output using the new weights.
ASICs

Training an ANN is where a large amount of computer time is spent. After computing there are three different operations performed. Implementing an ASIC could reduce this to one computation, speeding up this portion of the training by a factor of 3. However, as the size of the network grows, the calculations involved in computing the sum quickly outweigh the computation time of the multiplication. The circuitry in this proposed ASIC would closely resemble the advances shown in Google’s Tensor Processing Unit for continuous summations.

To test how much quicker any firmware or ASIC is at training a network, this new training method would have to be fully implemented and optimized in both hardware and software. A network would need to be trained once with each method, and the duration of training time would need to be logged. The time taken to train the network using both methods would be compared. Depending on the implementation, the speed improvement would vary based on the size of the network.

We can make a rough estimate of improvement by counting the number of calculations involved in computing $\delta_x$. This number would be approximately $n+3$, with $n$ being the number of connections between node $x$ and the next layer. The number of calculations using the new method would be at most $n + 1$. This means that the new method using ASICs would train each layer of weights at $\frac{n+1}{n+3}$% the speed of using a GPU, assuming an ASIC with the same number of cores and processing speed. However, the ASIC or firmware could keep a running total of as those $\frac{\partial E_{total}}{w_n}$ be designed to values are being computed, potentially cutting the training time in half, or by some other constant multiplier. Any training time speedup would depend entirely on how it is implemented.
Additional Notes

Any attempt to train a network depends entirely on the activation function. The activation function is chosen in large part to make the partial derivative easily computable. However, it might be the case that other activation functions are able to train a network with fewer trials. The commonly used ReLU function is easily computed and has a simple derivative, but has potential drawbacks in how complex a network is required to be in order to overcome the function’s limitations. These functions might also be prohibitive because of the complexity of their derivative. Implementing ASICs to compute the partial derivatives would make these functions cost effective, speeding up the training process even more. However, a new ASIC would have to be developed and produced for each activation function that is used. Implementing this with digital circuits in firmware would result in slower processing speeds than physical circuits, but the cost of development would be significantly lower.
Citations

Abstract

Unsolicited commercial email (commonly referred to as “spam”) is a serious problem facing internet users. Spam results in financial losses, leads to reduced productivity, and often serves as a delivery mechanism for viruses and malware. While different methods have been employed to combat spam, many solutions require a great deal of time and effort to maintain. Extensive research has been conducted on the suitability of machine learning techniques to filter spam, as they require less manual intervention. This paper examines a number of seminal and recent scholarly articles related to the use of machine learning to identify spam. Sources are identified as seminal or recent and arranged chronologically within these categories. The literature suggests that future research on this topic should center on combining different machine learning techniques to create effective spam filters. There is also a need to study filters under real-world conditions and to further explore the ways in which email messages are analyzed and represented within each technique.

Machine Learning Methods for Filtering Spam Email

The proliferation of spam email messages has become a serious problem facing internet users today (Awad & Elseuofi, 2011; Santos, Laorden, Sanz, & Bringas, 2011; Torabi, Nadimi-Shahraki, & Nabiollahi, 2015). Spam is typically defined as “unsolicited commercial email” (Sakkis et al., 2003, p. 50). According to Blanzieri and Bryl (2008), spam “is used to advertise different kinds of goods and services” (p. 65).

The problem has gained more attention in recent years due to the widespread use of smart phones, tablets, and other mobile devices to access email (Pérez-Díaz, Ruano-Ordás, Fdez-Riverola, & Méndez, 2016). Sakkis et al. (2003) stated that in 1998 roughly 10% of messages sent to business networks were spam. Recent estimates of the amount of spam vary, but Torabi et al. (2015)
stated that about 65% of all email messages were classified as spam.

Financial losses due to spam are significant. Awad and Elseuofi (2011) gave a figure of $355 million per year as the total cost to email users. When lost productivity is factored in, the cost climbs into the billions of dollars (Santos et al., 2011). Spam email is also used as a vehicle for the delivery of malicious software (viruses, malware, Trojans, etc.) and as a means to conduct phishing scams (attempts to trick users into divulging personally identifying information such as passwords) (Santos et al., 2011). Spam may also lead to violations of privacy rights and the perpetuation of illegal activity (Blanzieri & Bryl, 2008, p. 63-64). The prevention of spam is certainly a worthy goal deserving of research and study.

There have been attempts to address spam via legislation. One example is the CAN-SPAM act of 2003, which was an attempt by the United States government to place limits on the sending of spam email (Yu, 2011). However, research has shown that these efforts have been unsuccessful in addressing the problem, possibly due to the difficulty of enforcing such laws (Yu, 2011). Therefore, it is necessary to develop other methods for combating spam messages.

One of the most common approaches is the use of a so-called spam filter, which analyzes incoming email messages and marks those identified as spam (Drucker, Wu & Vapnik, 1999; Guzella & Caminhas, 2009; Pérez-Díaz et al., 2016). Such filters may either block messages identified as spam or they may indicate that a message is spam via some other means (such as color coding, sorting, or placement in a separate folder) and allow the user to make the final determination (Drucker et al., 1999).

One of the primary difficulties in the fight against spam is the fact that the senders of such messages (“spammers”) actively change their methods in an attempt to avoid detection (Fawcett, 2003). They modify the subject line of their messages and use punctuation marks to break up words, among other methods. Spammers also employ various techniques to falsify message header information and hide the origin of their messages (Blanzieri & Bryl, 2008). Another significant concern is the potential for a
spam filtering system to block non-spam (legitimate) messages (Drucker et al., 1999; Santos et al., 2011).

Machine learning techniques hold much promise in the area of spam prevention due to their suitability for text classification problems (Sakkis et al., 2003; Santos et al., 2011; Wu, 2009). They are also much more efficient than requiring administrators to manually create and maintain rules to detect spam messages (Torabi et al., 2015). Because of these advantages, a significant amount of research has been undertaken to identify the most appropriate machine learning approaches for identifying spam email.

This paper will provide a review of the literature on the subject of applying machine learning techniques to the filtering of spam email. It will be structured as follows. First, it will discuss a number of seminal articles which laid the foundation for subsequent research and summarize their contributions to scholarly knowledge on this topic. Second, it will examine recent papers, analyze their approach to this topic, and evaluate their connection to the seminal works. Third, it will compare the seminal and recent literature to show how scholarly knowledge on this topic has grown and developed. Finally, it will provide recommendations for future research based on the existing literature.

**Seminal References**

One of the foundational works on this topic is the paper by Drucker et al. (1999). This work can be classified as a research paper since the authors collected, analyzed, and interpreted data. Their research was carried out using a total of 3,617 email messages collected from employees of AT&T for this purpose (Drucker et al., 1999, p. 1052). The study was carried out in a controlled environment, as the authors used only sample messages that had previously been classified as spam or legitimate. The authors provided several tables of results (Drucker et al., 1999, pp. 1052-1053) and provided an analysis of their findings in their conclusion (Drucker et al., 1999, pp. 1053-1054). The primary author had significant credibility to address this topic, as he was a professor of electrical engineering and had consulted on the topic of machine learning for a number of major corporations (including AT&T, Lucent Technologies, and Bell Laboratories).
The purpose of this research was to determine if the Support Vector Machines (SVM) machine learning technique was useful for identifying and classifying spam email (Drucker et al., 1999, p. 1048). SVM is a classifier which uses a number of pre-labeled training examples to classify new examples as belonging to one of two different classes. (Drucker et al., 1999, p. 1050). Spam filtering presents significant challenges because the misclassification of legitimate email messages as spam is a highly undesirable outcome. The authors proposed a system where all email messages were sorted according to the likelihood that they were legitimate and delivered to the user. Messages considered likely to be spam would appear at the bottom of the list (Drucker et al., 1999, p. 1048).

Drucker et al. took an objectivist view in their research. They measured the performance of their filtering method (and others) via quantitative means. Three of the primary variables used to compare different methods were the error rate (the percentage of messages classified incorrectly), the miss rate (the number of legitimate messages classified incorrectly divided by the total number of legitimate messages examined) and the false alarm rate (the number of spam messages classified incorrectly divided by the total number of spam messages examined). As the authors stated, “given two classifiers with the same error rate, the one with lower false alarm and miss rates is the better one” (Drucker et al., 1999, p. 1049).

The authors found that SVM was one of the two best machine learning techniques for classifying their sample data (Drucker et al., 1999, p. 1053). SVM has an advantage over other methods in terms of training time – it takes less time for the method to “learn” to classify data items (Drucker et al., 1999, p. 1054).

One of the key limitations identified by Drucker et al. was that their results were “based on data sets collected from different individuals” (p. 1054). There is no guarantee that the sample data used was representative of the type and amount of spam messages received by other users. Since the authors used the same data to evaluate all of the methods they tested, SVM could provide the best performance based on that particular data, but “real world” performance may vary. Therefore, in at least some usage scenarios,
the authors determined that SVM is a useful technique for classifying spam email.

There are many different varieties of spam. In their seminal research article, Sakkis et al. (2003) considered filtering methods for spam sent to mailing lists. According to the authors, the paper “presents a thorough empirical evaluation of memory-based learning in the context of a novel cost-sensitive application, that of filtering unsolicited commercial email messages” (Sakkis et al., 2003, p. 49). The article was intended as a research paper, as the authors gathered data and interpreted the results. Their credibility was established by their association with the Athens University of Economics and Business as well as the National Centre for Scientific Research in Greece.

The purpose of the paper was to determine whether a memory-based machine learning technique would be an appropriate choice for filtering spam messages sent to mailing lists. These lists are often targeted by spammers in an attempt to reach as many individuals as possible (Sakkis et al., 2003, p. 51). Many lists are moderated – there is a human being who reviews messages submitted to the list before they are sent to the list subscribers. This is not an ideal situation, as “lists with intense traffic […] can overwhelm their moderators, and in many cases there may be nobody willing to act as a moderator” (Sakkis et al., 2003, p. 51). It is clear that an automated filter would be preferable.

The paper reflected an objectivist perspective. The authors believed the best filtering method could be determined based on a quantitative analysis of their data. They conducted their research using a database of 2,893 messages gathered from an actual mailing list (Sakkis et al., 2003, p. 52). The key variables were recall (the percentage of spam email blocked by the technique) and precision (the percentage of blocked messages that are actually spam) (Sakkis et al., 2003, p. 58). Additionally, the authors considered the weighted accuracy (the ratio of messages correctly classified divided by the total number of messages) and the weighted error rate (the ratio of messages incorrectly classified divided by the total number of messages) to account for the fact that blocking a legitimate message is a much more serious error than allowing a spam message to be delivered. Weighting was accomplished by allowing each legitimate message to be counted
as multiple messages, represented as $\lambda$ (Sakkis et al., 2003, p. 58).

Finally, the authors measured total cost ratio (the weighted error measured in a scenario where no spam messages are blocked divided by the weighted error of the filtering technique being examined) to provide a baseline performance measurement (Sakkis et al., 2003, p. 58).

A memory-based technique stores a set of training examples and classifies new examples “by estimating their similarity to the stored examples” (Sakkis et al., 2003, p. 55). The particular method examined by the authors was $k$-nearest neighbour. This method identifies a specified number (k) of training examples that are most similar to the item under review. The item is assigned to the most common class of these examples (Sakkis et al., 2003, p. 55).

A limitation of this study was that messages sent to a mailing list tend to be related to the same topic. Therefore, it can be easier to identify spam by looking for messages that are different. As Sakkis et al. stated, “one cannot, therefore, safely generalize conclusions drawn from experimenting with mailing lists to anti-spam filters for personal mailboxes” (p. 51). It is important to keep this study within the proper context and realize that it applies to one specific type of spam.

The authors determined their memory-based technique provided better performance than an existing filter based on the Naïve Bayes method, “particularly when the misclassification cost for non-spam messages is high” (Sakkis et al., 2001, p. 1). It is clear the authors achieved their purpose for the study, as they determined that memory-based techniques can be used to successfully filter mailing list spam.

Another foundational work was the theory paper by Fawcett (2003). The author did not collect new data for this paper. Instead, he relied on existing research articles and publically available data sources which he listed in an appendix (Fawcett, 2003, p. 146). The author’s primary argument was that “real-world in vivo spam filtering is a rich and challenging problem” (Fawcett, 2003 p. 140). His purpose was to list and describe the various research problems associated with real world spam filtering and provide suggestions as to how these research challenges might be met (Fawcett, 2003, p. 140). The author’s doctorate in machine
learning and previous work experience with influential technology companies such as Hewlett Packard made him well qualified on this topic.

The author describes four primary phenomena in this paper. The first is “the proportion of spam to legitimate email is uneven” (Fawcett, 2003, p. 140). In other words, different individuals receive different amounts of spam messages at different times. This makes it difficult to assess the performance of different filtering methods under real world conditions.

Another challenge facing researchers is the fact that the severity of different types of classification errors can vary. A false positive error (incorrectly classifying a legitimate message as spam) is considered to be significantly worse than a false negative error (incorrectly classifying spam as a legitimate message) (Fawcett, 2003, p. 143). Researchers must remember that “only the end user will know the consequences of filtering mistakes and be able to estimate error tradeoffs” (Fawcett, 2003, p. 143). Correct error weighting can only be accomplished by consulting the users.

The third challenge facing researchers is “the content of spam changes over time” (Fawcett, 2003, p. 143). The nature of spam changes in unpredictable ways as it is used to advertise different goods and services and to propagate different types of scams. The creation of a longitudinal spam database (reflecting changes in spam content over time) would be valuable for future researchers (Fawcett, 2003, p. 144).

The final challenge described by the author concerns “intelligent adaptive adversaries” (Fawcett, 2003, p. 144). Spammers are constantly changing and refining their methods for creating spam in order to defeat filtering methods. In most other text classification problems, the text is not composed with the intent of avoiding classification (Fawcett, 2003, p. 145). This makes spam classification a more difficult prospect.

This paper exhibits a strong objectivist philosophy. The author believed it was possible to determine the nature of an email method through scientific investigation. His goal was to help researchers view the problem as more than just “an isolated text classification task” (Fawcett, 2003, p. 140) and encourage them to consider some of the difficulties involved with spam filtering.
The author’s primary implication was research on spam filtering must take real world factors into consideration. In particular, he argued that “researchers wishing to pursue this domain should begin collecting longitudinal data in a controlled manner” (Fawcett, 2003, p. 145). He believed such databases should be made publically available to aid with general research in this area (Fawcett, 2003, p. 146). By laying out the various challenges related to real world spam filtering and providing suggestions on how future researchers might cope with them, the author achieved his goal.

As new machine learning techniques are developed, researchers look for ways to apply them to the problem of spam email. Wu (2009) wrote a seminal article in which he proposed a hybrid method for spam filtering which combined rules and neural networks. The reference to “experimental results” (Wu, 2009, p. 4321) early in the paper is an indication that it is a research article. This view is further supported by the fact that the author gathered data and described his research methods.

The author’s primary goal was to show the proposed filtering technique addressed shortcomings of previous methods used to combat spam (Wu, 2009, p. 4321). These methods often involved blocking email addresses known to send spam and the creation of rules to filter messages that showed characteristics of spam. Because spammers are constantly changing their methods to evade spam filters, much effort was required to maintain these rules and address blacklists. A machine learning approach (which requires much less manual intervention) would be preferable (Wu, 2009, p. 4321). The author’s credibility was established by his position as a member of the Department of Electrical Engineering at the National University of Kaohsiung, Taiwan.

This article also employs an objectivist approach. The author believed the nature of an email message could be determined by careful examination using quantitative techniques (Wu, 2009). He identified a number of key variables. These include mean squared error (the difference between the neural network’s desired output and the actual output), accuracy (the ratio of messages correctly classified to the total number of messages), and total cost ratio (the ratio of messages correctly classified to messages incorrectly classified) (Wu, 2009, p. 4327).
A neural network consists of several layers of nodes which receive input and generate output (Wu, 2009, p. 4325). Feature values for the email being evaluated are fed into the first layer (the input layer). An activation function is used to generate an output value and pass that value on to nodes in the second layer. This process may be repeated through a number of intermediate (or hidden) layers. Finally, there is an output layer consisting of a single node. This layer generates the final output, which is used to classify the email as spam or legitimate. The number of layers in a neural network may vary; Wu used a network consisting of one input layer, two hidden layers, and an output layer (p. 4326). It is necessary to train a neural network by giving it examples of spam and legitimate messages. If the network gives incorrect output for a training example, weights between the nodes are adjusted in order to generate correct output.

The sample for the study consisted of a total of 120,207 email message which the author collected for his research (Wu, 2009, p. 4327). These messages were manually classified as either spam or legitimate so the network’s output could be verified. The author found that publically available email databases did not contain syslog information (which can only be gathered from email servers) and thus were inadequate for testing his proposed filtering method (Wu, 2009, p. 4327).

One of the key limitations the author identified was the fact that only certain features (portions) of email addresses are examined in this method (Wu, 2009, p. 4329). There may be other features not considered in this study that could provide for more accurate identification of spam messages. Another limitation is neural networks are dependent on having an adequate number of training examples, and the time needed for the network to train (or “converge”) can be unpredictable. There are many ways to structure a neural network, and the author suggested future research could examine different architectures. (Wu, 2009, p. 4329). Finally, the author noted the best approach to spam detection and filtering would likely consist of a combination of several different approaches due to the strengths and weaknesses inherent in each (Wu, 2009, p. 4329).

The author determined that the neural network was able to identify spam email with a high degree of accuracy and, in fact,
outperformed many established filtering methods (Wu, 2009, p. 4329). A key advantage of the neural network over other methods is the ease with which it can be adapted to new spamming methods – it is only necessary to provide new training examples. Because this method examined information from email server logs in addition to the messages themselves, it was less easily bypassed by changing the text of the messages (Wu, 2009, p. 4329). Based on these findings, the author clearly achieved his stated purpose.

**Summary of Seminal References**

The foundational research articles demonstrate that machine learning techniques are useful for identifying spam email (Drucker et al., 1999; Sakkis et al., 2001; Wu, 2009). Machine learning techniques allow filters to adapt to the changing nature of spam with less human intervention than other methods. A recurring theme in the literature is there is not one machine learning technique that can be said to be superior in all respects. Sakkis et al. (2003) noted “our main interest is in combining memory-based, probabilistic, and other induced classifiers” (p. 71). Wu (2009) stated “it may not be possible to detect, totally and always exactly, all spams using a single technique” (p. 4329). It will be important for future researchers in this field to consider a wide variety of approaches and to consider how different approaches may be combined in order to create the most effective spam filters.

Throughout the seminal references there is an objectivist approach (Drucker et al., 1999; Sakkis et al., 2001; Fawcett, 2003; Wu, 2009). The authors consistently compared filtering techniques on the basis of variables such as error rate, accuracy, total cost ratio, and other quantitative measurements. Underlying this research was an assumption that all messages could be classified as either spam or legitimate according to different characteristics. It could be said that the true nature of a message was an objective truth waiting to be discovered via scientific inquiry.

Early research on this topic often required authors to gather their own data (Drucker et al., 1999). Eventually researchers recognized the value of publically available databases of email messages (Sakkis et al., 2001). These databases simplified the research process by reducing the amount of time required to gather
They also enabled researchers to more accurately compare different filtering methods against a consistent data set. However, as all the sources note, there is no guarantee that any collection of email messages is representative of real world conditions as the amount and type of email received varies widely from user to user. Fawcett (2003) in particular argued that researchers need to be aware of and account for real world conditions in their research.

Recent References

In 2008, Blanzieri and Bryl conducted a review of current literature related to machine learning and spam filtering. Their stated purpose was to “give an overview of the state of the art of machine learning applications for spam filtering, and of the ways of evaluation and comparison of different filtering methods” (p. 63). Since they did not gather any data or conduct any experimental evaluation of their own, but relied on existing research papers, this paper can be classified as a literature review.

The authors began by providing a brief overview of spam email and some of the problems it causes, including financial impact, loss of privacy, and the spread of illegal activity (Blanzieri & Bryl, 2008, pp. 63-64). Then they provided an in-depth discussion of the characteristics of spam. In addition to advertising products, spam is often used in an attempt to trick individuals into divulging personally identifying information (Blanzieri & Bryl, 2008, p. 65).

Next, the authors provided an introduction to spam filtering which they defined as the “automatic classification of messages into spam and legitimate email” (Blanzieri & Bryl, 2008, p. 67). There are a number of different machine learning techniques used to perform this classification. The authors first grouped techniques into five categories: bag-of-words model, language-based filters, filters based on non-content features, collaborative spam filtering, and hybrid approaches. When discussing each technique, they grouped articles in largely chronological order, beginning with seminal papers (Blanzieri & Bryl, 2008, pp. 71-75). In the following section, the authors discussed a number of quantitative measurements used to evaluate spam filtering methods along with a listing of 16 studies dedicated to comparing different methods (Blanzieri & Bryl, 2008, pp. 75-83). The relative performance of
each method can be determined based on how accurately they classify a set of messages in a database (or corpus).

One limitation of all spam filtering methods is the fact that spammers are reactive in nature and constantly change their messages to try to avoid filters (Blanzieri & Bryl, 2008, p. 83). This makes it practically impossible to determine how well a filter will perform when confronted by this reactivity. It is important that researchers continue to explore new filtering techniques to address this problem. Another factor to consider is that spam filtering techniques are only effective if server administrators employ them (Blanzieri & Bryl, 2008, p. 84). There is not one best way to filter spam; therefore the best approach is often to employ a variety of techniques.

The authors determined that the Naïve Bayes method provided “high speed and simplicity with sufficiently high accuracy,” and thus it was one of the most commonly used filtering methods (Blanzieri & Bryl, 2008, p. 83). This method “rests on the so-called naïve independence assumption, namely that all the features are statistically independent” (Blanzieri & Bryl, 2008, p. 72). From this conclusion, it is clear that they accomplished their stated purpose.

Guzella and Caminhas (2009) provided a more focused overview of several machine learning approaches to spam filtering. The authors’ purpose was to evaluate the adequacy of these different approaches by examining “only the most distinguishing characteristics of each algorithm” (Guzella & Caminhas, 2009, p. 10206). In addition, the authors did not gather new data but relied on previous research. This shows that the article can be classified as theoretical.

The authors described the steps followed by a typical spam filter in order to classify email messages. These steps include tokenization, the extraction of words from the message body; lemmatization, the reduction of words to their root forms; stop-word removal, the elimination of words that occur in many messages; and representation, the conversion of the processed message to the format required by a specific machine learning technique (Guzella & Caminhas, 2009, p. 10207). Additionally, machine learning techniques require training (the examination of
many sample messages, both spam and legitimate) prior to implementation (Guzella & Caminhas, 2009, p. 10206).

The authors described a number of different machine learning techniques. These included Naïve Bayes, Support Vector Machines, Artificial Neural Networks, Logistic Regression, Lazy Learning, Artificial Immune Systems, and Boosting (Guzella & Caminhas, 2009, pp. 10211-10216). They did not provide all the specifics of each method but referred to existing literature to provide details. Additionally, they described a number of hybrid filters which employed multiple machine learning techniques (Guzella & Caminhas, 2009, p. 10216).

As an alternative to previous text-based approaches, the authors described techniques based on image analysis (Guzella & Caminhas, 2009, p. 10216). These techniques view the email message as an image and attempt to identify spam based on the layout of the document rather than on its contents. Since many spam emails contain images as well as text, these techniques may also be used to examine images attached to email messages (Guzella & Caminhas, 2009, p. 10216).

The author’s primary conclusion was that “more reliable systems can be obtained by combining several algorithms with different characteristics” (Guzella & Caminhas, 2009, p. 10221). Each technique described in the paper has individual strengths and weaknesses. By combining different techniques, researchers may be able to account for these differences to produce a robust filter.

A key limitation that the authors identified in the literature was the fact that many researchers conducted their experiments on static data (Guzella & Caminhas, 2009, p. 10220). They argued that “in order to have an adequate assessment of the performance of filters, it is necessary to adopt more realistic evaluation settings” (p. 10220). This echoed Fawcett’s (2003) call for researchers to collect longitudinal data and account for real life scenarios. The ever-changing nature of spam implies that any filtering technique will need to be updated on a regular basis to account for the different methods spammers employ. By providing an overview of current techniques as well as recommendations for future researchers, the authors achieved their stated purpose.

Awad and Elseuofi (2011) also examined a number of machine learning techniques often used for spam filtering. Their
analysis used a single database of email messages in order to compare the accuracy and performance of each method. This ensured that accurate performance comparisons could be drawn between the different methods. In their introduction, the authors made reference to an “experiment implementation” (Awad and Elseuofi, 2011, p. 174), showing that their work can be categorized as a research paper. This is confirmed by their discussion of methods and results later in the article (Awad and Elseuofi, 2011, pp. 180-183).

The stated purpose of this article was to give a description of several different machine learning techniques used for spam filtering and analyze their performance (Awad and Elseuofi, 2011, p. 173). These methods included Naïve Bayes, k-nearest neighbour, Artificial Neural Networks, Support Vector Machines, and Artificial Immune Systems (Awad and Elseuofi, 2011, pp. 175-180). Other methods (which the authors refer to as knowledge engineering) may be used to filter spam, but they are less desirable than machine learning as they require rules to be manually created and updated. Machine learning allows a spam filter to build its own rules by examining sample email messages (Awad and Elseuofi, 2011, p. 173). The primary author is a member of the Math and Computer Science Department of Port Said University, Egypt.

The email database used by the authors (SpamAssassin) contained 6,000 unique messages. They employed quantitative techniques to conduct their research. Important variables included spam precision (spam messages correctly classified divided by messages classified as spam), spam recall (spam messages correctly classified divided by the total number of messages), and accuracy (messages correctly classified divided by the total number of messages) (Awad & Elseuofi, 2011, pp. 181-182). Their use of quantitative variables demonstrates an objectivist approach, where different methods can be evaluated using empirical means.

The authors did not identify any specific limitations in their paper. Fawcett’s (2003) comments regarding the importance of real world testing might be applied to this article. It is difficult to determine the true performance of any machine learning technique when measured against a single, static database of messages.

The results indicated that Naïve Bayes had the highest precision, recall, and accuracy among the six methods evaluated
The authors suggested that future research should focus on creating hybrid systems using two or more techniques, as this could provide a way to address the drawbacks of individual methods. In particular, they found that artificial immune systems performed well in terms of precision and might prove useful in such systems (Awad & Elseuofi, 2011, pp. 183). The authors achieved their stated purpose to compare and evaluate different machine learning methods, although more discussion of limitations may have been useful.

Santos et al. (2011) studied the application of one particular machine learning method, the enhanced Topic-based Vector Space Model (eTVSM), to spam filtering. The purpose of their paper was “to show that the proposed method can detect the internal semantics of spam messages” (Santos et al., 2011, p. 437). The article contains a section titled “Experiments and results” (Santos et al., 2011, p. 440), which demonstrates that it is a research article.

According to the authors, many spam filtering systems use the Vector Space Model (VSM) to represent the text of an email message. A key weakness of this method is its inability to deal with meaning and the relationships between words; for example, synonyms (Santos et al., 2011, p. 438). They proposed the use of eTVSM, which models not only the content of the message but also the relationships between concepts through the use of an ontology graph (Santos et al., 2011, p. 439).

The authors used quantitative variables to compare the performance of different filtering methods (Santos et al., 2011, p. 441). Key measurements included accuracy, the number of correctly classified messages divided by the total number of messages; true positive ratio, the number of correctly classified spam messages divided by the number of messages misclassified as legitimate plus the correctly classified spam messages; false positive ratio, the number of legitimate messages incorrectly classified as spam divided by the number of legitimate messages correctly classified plus the legitimate messages incorrectly classified as spam; and the Area Under the ROC Curve, which was determined by graphing the rate of true positives against the rate of false positives. Additionally, they measured training time, how long it took a method to “learn” the training examples, and testing time, how long it took a method to analyze the test data (Santos et
This demonstrates that the authors took an objectivist approach; they believed that careful study and measurement would reveal the best machine learning method for this task.

The article identified several limitations related to the authors’ research. The first is that eTVSM was only used to represent synonyms; other types of relationships between words were not considered (Santos et al., 2011, p. 442). The authors suggested that future research might examine other linguistic concepts such as hyponymy, the relationship between a general class and a specific member of that class (Santos et al., 2011, p. 442). This model also has difficulty handling polysemenes, words with multiple meanings (Santos et al., 2011, p. 442).

Santos et al. used a publically available database of 2,893 email messages to conduct their experiment (p. 440). They used eTVSM to represent email messages for filters based on Bayesian networks, decision trees, k-nearest neighbour, and support vector machines (Santos et al., 2011, p. 441). The result indicated that the use of eTVSM allowed these filters to achieve higher levels of accuracy and also reduced the number of legitimate messages that were incorrectly identified as spam (Santos et al., 2011, p. 441). Thus, they succeeded in showing that this representation model was suitable for use in spam detection filters.

Many researchers take a quantitative approach to the issue of spam email. By contrast, Yu (2011) used a qualitative approach to analyze a regulatory attempt to limit the sending of spam email. This study was intended to determine whether the CAN-SPAM Act of 2003 succeeded in reducing the volume of spam messages (Yu, 2011, p. 716). The author collected data, discussed experimental methods, and reported results; therefore, this is an example of a research paper. The credibility of the paper is established by the author’s position on the faculty of The College at Brockport, State University of New York.

The CAN-SPAM act was legislation passed by the United States government in an attempt to reduce the number of spam messages sent (Yu, 2011, p. 716). The goal was to increase the cost of transmitting spam by imposing restrictions on how it could be sent. Some of the restrictions included “using authentic header information, no deceptive subject lines, identifying the message as
an advertisement, providing the real physical location of the business, offering an opt-out choice, and honoring opt-out requests within 10 business days” (Yu, 2011, p. 716). The sending of commercial email messages that did not comply with the act would be considered a criminal offense and could result in fines or jail time. However, previous literature had suggested that the Act had little or no impact on the spam issue (Yu, 2011, p. 716).

To conduct this research, the author gathered 3,983 spam messages from five Gmail accounts (Yu, 2011, p. 719). These accounts were created with the intention of gathering as much spam as possible. This was accomplished by using them for a variety of online registration forums and posting them on websites (Yu, 2011, p.719). The spam messages that were received were then subjected to qualitative review by the author. They were first categorized as either scam/fraud, advertising/promotion, or unclear. They were then examined to determine if they met the five requirements of the CAN-SPAM act: providing an opt-out choice, being non-deceptive in nature, containing true information on the sender of the message, listing a physical business address, and providing accurate host system information (Yu, 2011, p.720).

The author found that only 108 of the spam messages studied appeared to meet the requirements of the CAN-SPAM act (Yu, 2011, p. 726). There is some debate as to whether a P.O. Box meets the requirements for a physical address; if it does not, then only five messages were in compliance with the Act (Yu, 2011, p. 726). In particular, only 268 messages offered an opt-out choice, and only 183 provided a physical address (Yu, 2011, p. 726). These findings suggest that spammers are generally ignoring the requirements of CAN-SPAM.

A key limitation identified by the author was that the messages collected for this research came from only five separate email accounts (Yu, 2011, p. 729). Due to the small sample size, this data cannot be said to be representative of a general population. Additionally, the email accounts used to gather data were intentionally exposed to as much spam as possible, and thus they may not be representative of the amount and type of spam received by a typical email user. Despite these limitations, the findings were consistent with previous research on this topic (Yu, 2011, p. 729).
This paper suggests that legislative attempts to regulate spam are not effective, due in part to the difficulty associated with enforcing such legislation (Yu, 2011, p. 728). Therefore, researchers should continue to explore alternative methods for dealing with the problem (including filtering). The author suggested that future research should explore the true impact of spam beyond simply the time required to delete the message; for example, health risks associated with buying drugs online (Yu, 2011, p. 729). Thus, the author achieved his goal and provided a direction for future research.

Torabi et al. (2015) examined support vector machine theory and built on the foundational work of Drucker et al. (1999). Their goal was to describe how SVM may be applied to the problem of spam filtering, as well as the criteria used to evaluate its suitability for this task (Torabi et al., 2015, p. 11). They described theoretical issues surrounding SVM but did not gather data or perform experiments. Therefore, this article can be classified as a theory paper.

According to the authors, spam filtering techniques could be divided into two groups, end-user (or client side) and server side. End-user techniques are used on individual client machines, while server side techniques attempt to block spam at the email server level. (Torabi et al., 2015, p. 12). Among server side techniques, machine learning-based filters are preferable to other methods because they require less effort to maintain (Torabi et al., 2015, p. 15).

The SVM method represents messages as points on a two dimensional plane. It then attempts to separate the messages into different classes by drawing boundary lines between them. This method is well-suited to classification problems and thus appears to be a good candidate for a spam filtering system (Torabi et al., 2015, pp. 18-19).

The authors initially examined 10 scholarly articles dealing with SVM for spam email filtering, beginning with the seminal paper by Drucker et al. (1999) and continuing through 2014. They found that the literature showed SVM had higher memory and time requirements for large datasets than other methods (such as Naïve Bayes), and thus SVM was considered less useful for filtering spam (Torabi et al., 2015, p. 22).
Torabi et al. believed that the performance problem could be addressed by reducing the number of features (individual elements) that were analyzed for each email message passing through a filter (Torabi et al., 2015, p. 22). In addition, they examined a further eighteen scholarly papers published between 2005 and 2014 where a separate technique was used to identify key features in email messages, and these features were then analyzed using SVM. This approach was found to provide better performance, and in many cases it was also more accurate than initial implementations of SVM (Torabi et al., 2015, pp. 24-25). The articles reviewed by the authors reflected an objectivist approach, evaluating filtering techniques based on quantitative measurements such as accuracy, error rate, and speed.

In this paper, the authors identified a key limitation of SVM – it performs poorly when evaluating many features for each email message. They also suggested that the limitation could be addressed by reducing the number of features considered for each message, and their suggestion was supported by a significant amount of literature (Torabi et al., 2015, p. 25). This information is valuable for future researchers in this field, as it provides insight into how the SVM model can best be applied to spam filtering. Torabi et al. clearly achieved their purpose and made a useful contribution to the body of scholarly knowledge.

Rather than investigate a new machine learning technique, Pérez-Díaz et al. (2016) sought to increase the accuracy of several existing techniques through the application of rough set theory. Their goal was to determine if this method would increase the accuracy of spam filters while maintaining satisfactory performance (Pérez-Díaz et al., 2016, p. 3). They collected data for their experiment, and the article contains a section titled “Model Benchmarking” (Pérez-Díaz et al., 2016, p. 5) which discusses their results, indicating that this is a research paper. Their credibility was supported by their positions on the faculty of the Higher Technical School of Computer Engineering at the University of Vigo, Spain.

The authors stated that only one type of machine learning technique, Naïve Bayes, is commonly used in spam filters. Other techniques do not provide a usable combination of speed and accuracy (Pérez-Díaz et al., 2016, pp. 1-2). Rough set theory has
the ability to “discover redundancy and dependencies between features” (Pérez-Díaz et al., 2016, p. 2), and the authors believed it would be useful in identifying the features most relevant to the accurate classification of an email message. If this technique were shown to increase classification accuracy, it could allow for additional machine learning techniques to be incorporated into spam filters.

Pérez-Díaz et al. tested their proposed technique on a publically available database of 9,332 messages (p. 6). They employed quantitative measurements, such as the percentage of messages that were correctly classified, incorrectly classified as spam (a false negative), and incorrectly classified as legitimate (a false positive) (Pérez-Díaz et al., 2016, pp. 5-6). This indicates that they took an objectivist approach.

One limitation identified by the authors is that the methods employed by rough set theory have difficulty handling the inherent uncertainty of text classification problems (Pérez-Díaz et al., 2016, p. 7). They suggested that future research might focus on refining these methods to better deal with uncertainty. This provides an opportunity for future researchers to contribute to the scholarly literature on this topic.

The author’s results demonstrate that they met their goal. They found that the use of rough set theory reduced the number of false positive errors generated by the Naïve Bayes, Flexible Bayes, AdaBoost, and Support Vector Machines filtering techniques (Pérez-Díaz et al., 2016, p. 7). The performance impact was described as negligible, due in part to the fact that rough set processing was only used for messages that were initially classified as spam (Pérez-Díaz et al., 2016, p. 7).

Summary of Recent References
A theme that emerges from the literature is the indication that while the Naïve Bayes classifier is generally regarded as a preferable technique (Awad & Elseuofi, 2011; Blanzieri & Bryl, 2008; Pérez-Díaz et al., 2016), it could be possible to create superior spam filters through the use of several machine learning techniques (Awad & Elseuofi, 2011; Blanzieri & Bryl, 2008; Guzella & Caminhas, 2009; Pérez-Díaz et al., 2016; Wu, 2009). In particular, Pérez-Díaz et al. (2016) showed that Naïve Bayes could
be improved by using it in conjunction with rough set theory. Examination of different combinations of machine learning techniques holds great promise for future research on this topic.

Despite Fawcett’s (2003) call for an increased awareness of real-world factors related to spam filtering, Guzella & Caminhas (2009) noted that many researchers still conducted their experiments on static data which did not reflect the way in which spam email changes over time. While accounting for real-world factors is admittedly difficult, an opportunity exists for the creation of longitudinal databases that could demonstrate how machine learning techniques adapt to changing spam patterns. This may prove easier to address than attempting to gather spam samples that are truly representative of the population of internet users at large, for example.

A third research possibility is to investigate different methods of representing email messages for machine learning techniques (Pérez-Díaz et al., 2016; Torabi et al., 2015; Wu, 2009). Before a message can be classified by a filter, the different features of the message must be extracted and translated into a form that a particular machine learning technique can handle (Guzella & Caminhas, 2009). As Pérez-Díaz et al. (2016) demonstrated, changing the way messages are represented has the potential to improve filter accuracy. Future researchers might continue to explore this aspect of spam filtering.

Comparison of Seminal and Recent References

Much of the research conducted on spam filtering reflects an objectivist approach. Quantitative methods are often used to evaluate and compare the suitability of different machine learning techniques for the problem of email classification. This is evident in Drucker et al. (1999), Sakkis et al. (2003), Santos et al. (2011), and Torabi et al. (2015), among others. Qualitative methods have been employed in spam research, often when the focus is on some other aspect of the problem besides filtering. For example, Yu (2011) used a qualitative approach to evaluate the effectiveness of regulating spam via legislative means. There is a consistent view that the nature of spam can be discovered by structured scientific inquiry, and it can be classified accordingly.
Earlier references tended to evaluate filters based upon a single machine learning technique (Drucker et al., 1999; Sakkis et al., 2003). As the body of knowledge on this topic developed, researchers became aware of the promise found in combining multiple methods (Wu, 2009). A significant number of recent papers incorporate multiple machine learning techniques in their filtering methods (Awad & Elseuofi, 2011; Guzella & Caminhas, 2009; Santos et al., 2011; Torabi et al., 2015). This shows one way in which research on this topic has grown and matured.

There is a great level of consistency between different sources with respect to methods. The accuracy of filtering methods is of particular interest, with practically all research articles containing related measurements. Early research often required authors to gather their own database of email messages (e.g. Drucker et al., 1999), but as time went on the importance of having a standard message set for testing became apparent (see especially Sakkis et al., 2003). Testing different techniques against the same messages allows for valid comparisons to be drawn regarding accuracy and speed.

Conclusion

This paper has provided an overview of the application of machine learning techniques to the filtering of spam email. The task of identifying the best method for classifying email presents many challenges due to the difficulty of carrying out research under real-world conditions, the severe consequences of false positive errors, and the fact that spam is designed to avoid classification (Fawcett, 2003). However, existing literature has determined that the Naïve Bayes classifier is generally considered to be well-suited to this challenge (Awad & Elseuofi, 2011; Guzella & Caminhas, 2009; Pérez-Díaz et al., 2016). Many researchers, including Pérez-Díaz et al. (2016) and Wu (2009), have suggested that combining different methods is the key to developing the most effective spam filters. Other topics for further research include an increased emphasis on real-world testing (Guzella & Caminhas, 2009; Fawcett, 2003) and evaluating the way in which messages are represented for different techniques (Pérez-Díaz et al., 2016).
Examining different combinations of machine learning techniques and evaluating their combined effectiveness in combating spam is a promising direction for future research. As computing power increases, more machine learning techniques that were not viable for such tasks even a decade ago may now be practical. As the volume of spam email sent shows no signs of decreasing, Artificial Intelligence may prove to be a key tool for mitigating its impact.
References


