Artificial Intelligence: The Most Dangerous Tool

Executive Summary

With billions of dollars poured into the research and development of artificial intelligence (AI), its usefulness and impact on societies is under intense scrutiny. AI is not exactly a new field. From as early as 1950, famous computer scientist Alan Turing developed a Turing test to examine whether a subject is a machine or a human. Although some argue that the test is impractical and flawed, it marked as a major advancement in recent computer science history. Fast forward to today, in 2016, where highly capable computers are cheap and readily available, this harbors a favorable environment for the growth of AI as they rely heavily on the computer's' computational power. AIs find its application is innumerable areas including automobiles in the form of self-driving cars, military applications, medical applications, and many others just to name a few. So far, they have shown tremendous computational power and most importantly, unmatched decision making capability to the extent that it is clear AI will be deployed to even more fields and become more prevalent to directly impact people's lives. However, due to its its software based nature, it raises grim concerns. It is well known that software bugs occur all the time, and notably, in this paper, three independent cases are evaluated, and their relevance to AI is also illustrated. From the Turing Test and Mars Global Surveyor to the SSL Heartbleed, all three exhibit typical software faults that may be relevant to potential AI disasters. Due to its rapid development, many have expressed concerns and doubts for AI's impeccable intelligence that is unmatched by humans. Notable people such as Elon Musk and Stephen Hawking have made efforts to combat this concern. In this paper, we have developed strategies and made recommendations in order to avoid the AI disaster that may pose global catastrophic risk to humanity.

Our research finding indicate that while the field of Artificial Intelligence is by no means a novel field, the definition of what encompasses an AI varies. At the beginning of this report, we generate our own view of Artificial Intelligence, in which we find that AIs share many common attributes with historical programming disasters, including: sensitivity to input data, prone to errors due to oversights or negligence in programming, and used immensely in our current society (albeit in more primitive forms). In our initial survey of AI definition and presence in our society, we find the definition of Artificial Intelligence to be contested. In particular, the subject of whether the inadequacies of the Turing Test, a set of protocols and behaviour analysis techniques to determine if an AI can pass as *human*. Three camps exist in this debate, some who agree with the test and others who don't, the third camp regards the test as frivolous and unnecessary as AI technology matures. However what is prevalent in all these discussions is the threat that an advanced AI technology poses to the human race should it have no restrictions. An AI can lead to devastating result, by outsmarting humans, or simply because of a malfunction. Such an event, coined *technological singularity*, is a subject of debate in both literature and pop culture with no immediate means of prevention. An Artificial Intelligence can appear from any code base currently developing in AI technologies, for example Google's search

engine or a music suggestion engine already employ technologies closely related to AI development. Further research into historical programming disasters has shown us that although we cannot pinpoint exactly from where the first advanced AI will appear from, there are certain cultural, financial and social modifications we can make to minimize the chance of such a technological singularity. Through NASA's Mars Global Surveyor (MGS) historical disaster, we found the code review process to be lacking in terms of strictness and completeness, and all errors related to the disaster to have stemmed from human negligence. Additionally, the redundant systems were clearly not tested a priori, and the team did not expect the initial systems to fail. This led to a situation in which a panicked team without proper procedure in case of an emergency further added to the catastrophe by changing parameters without a thorough understanding of the problem. In our second disaster, research found that an exploit in an implementation of the TLS protocol lead to the possible leaking of sensitive information. We found that OpenSSL, the popular TLS protocol implementation that was exploited was neglected financially by large corporations that utilized the code, was seriously understaffed and underwent no external quality checks. It seems that human negligence and inadequacies remains the main factor in many programming disasters, therefore the solution should not be on theoretical concepts, but rather implementation issues that may arise due to human negligence. Based on these findings, we find the following to be effective in preventative measures against future AI disasters:

- *Support for AI Projects* - This mainly stems from the lack of manpower due to financial constraints in contemporary programming projects. Perhaps critical pieces of code such as OpenSSL, and definitely future AIs should get the proper financial backing they deserve.
- *Emphasis on Coding Analyzing and Test Frameworks* - humans cannot check every facet of code to ensure correctness, the issue is then designing programs that are capable of doing so. As AIs advance issues of how to test their correctness arise. We presently cannot give a strict definition of an AI in terms of behaviour or programming, how then should we test it? More research is needed in this area as AI's continue to develop.
- *Open Source Projects* - The more eyes on a project, the more feedback it gets and it lessens the chance of an oversight. AI projects should foster an inclusive community, instead of holding knowledge in the hands of a few. Open Source Projects create a cooperative environment, and could quicken rapid development of AIs, as indexed by Open Innovation projects of the past.

AI is something that is happening, and we can fight it, or we can follow it and make sure that is going on the right pass, because rather the risk, AI will truly change lives, to the better, and that is why some countermeasures and prevention have to be put in place in order to make sure an AI does not derive off his path. A few measures would be to build an AI on existing and tested libraries, thinking ahead and preparing preventive sculpted rules, and finally working collaboratively and transparently with others on an AI Open Source project that is available to everyone.

One thing for sure is, while working on this project we realized that you don't have to be an engineer to be concerned about AI. AI has the power to change lives for better or worse, and once we started working on this project, it was short enough that we realized that the concept of AI and its impact is embedded all around us, and almost everyone has his own idea about AI that he generated one way or the other.