Project Evaluation

For an Intrusion Detection System using a Neural Network

Version 1.0

Submitted in partial fulfillment of the requirements of the degree of MSE

Blake Knedler

CIS 895 – MSE Project

Kansas State University

Table of Contents

1 Introduction

This document will be a Project Evaluation for the Intrusion Detection System (IDS). This document will describe the issues faced during the product development, the accuracy of the estimations used, some lessons learned from the project, and future work that could be done to enhance the final product.

2 Issues Faced

The IDS has a few requirements to be able to properly work on a host system. This section will explain the requirements for the IDS to work properly and how to start the system. The IDS may work with environments not described, but those environments are untested and are therefore not supported.

2.1 Lack of Useful Data

The biggest issue that I faced during the development of the IDS is the lack of useful data to train the Neural Network on. Originally I had found some data used by the KDD competition in 1999 documented by the Air Force. However, well into the second phase of the project I found that this data actually used an intelligent system to monitor the traffic along with using the header information about the packets received. My implementation of an IDS does not keep track of connections and packets received to use for future packets with the same host. This caused many issues with unreliable decisions by the neural network due to the inability to properly train. It would be extremely beneficial to gather new data and modify the training portion of the IDS to handle better training data. Unfortunately, I did not have time to explore new paths of better data or collecting my own data. I decided to proceed forward with the current implementation as more of a proof of concept that could be built upon.

2.2 Lack of Knowledge

An issue I faced was the learning curve I had to overcome on neural networks and networking. I had very little prior experience in these two areas of software. I did know this going into the project and still continued because I knew it would be a great learning experience for me. I did not realize the complexity of these two areas which caused me to produce less than someone who knew about these two areas prior to the start of this project.

2.3 Time Management

I had a difficult time with time management, especially as the project progressed. This is mainly due to the fact that I am a distance education student. I am currently working full time and only working on this project part time. At times, this is difficult to juggle working full time especially when I am needed to work overtime and make progress on my project. I think I would have been much more effective if

I could devote my full time to the project or been on campus to meet with the advisors face-to-face.

3 Accuracy of Estimations

3.1 SLOC and Total Time Estimate

	SLOC	Time (months)
Estimate	4000	6.59
Final	1172	6.33
Accuracy	29.30%	96.11%

Table 1. SLOC/Time Estimate

Shown in Figure 1 is my overall SLOC and Time estimates compared to my actuals. I was very accurate on my time estimate considering I knew I would finish at the end of the Spring 2016 semester. However, my SLOC estimates were very inaccurate. I believe the reason for my inaccurate SLOC estimates were due to the lack of knowledge of neural networks and networking. I had very little knowledge about these two areas starting the project. Even though the SLOC counts are low, the time was able to still be very accurate due to the complexity of this application and my initial lack of knowledge in these areas.

3.2 Time Breakdowns

Below are figures showing the breakdown of time spent on each phase of the project and on individual tasks within each phase.

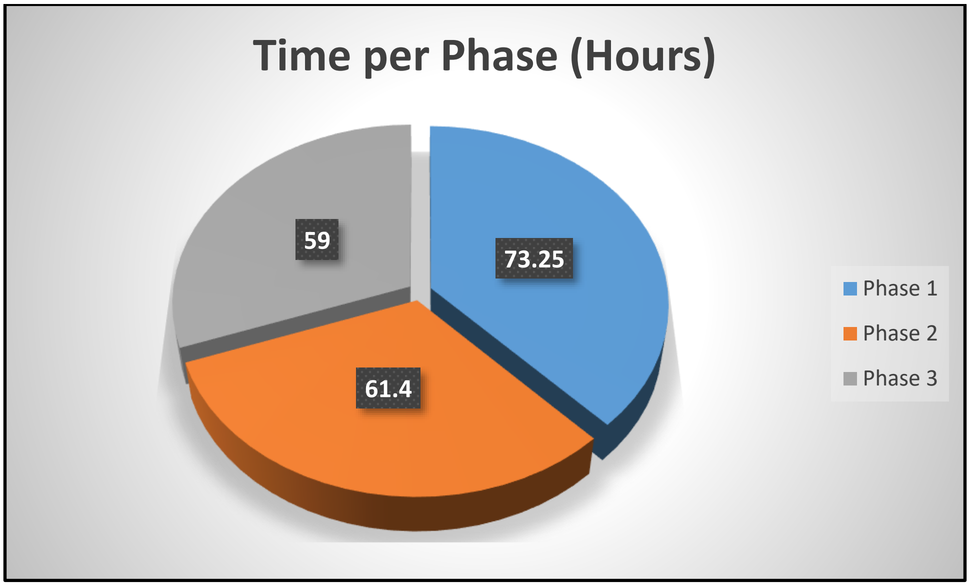


Figure 1. Overall Time Spent per Phase

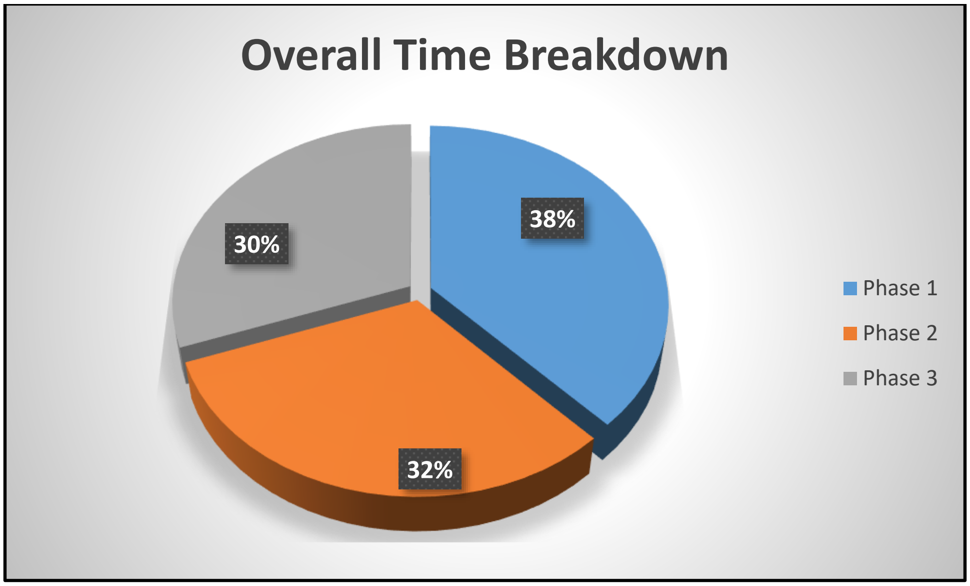


Figure 2. Overall Time Spent per Phase Percentage

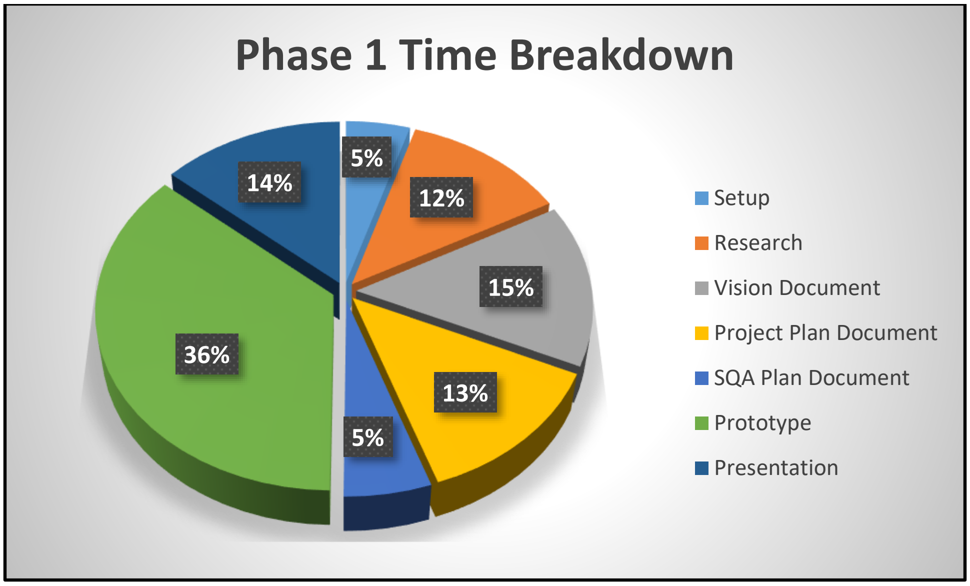


Figure 3. Phase 1 Time Breakdown

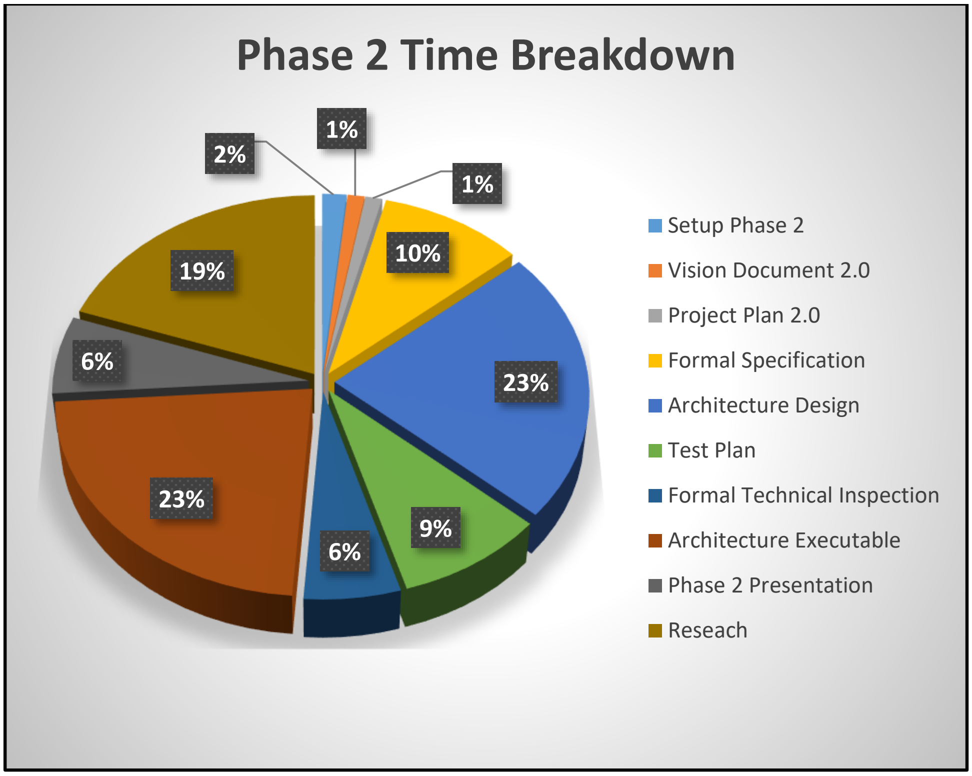


Figure 4. Phase 2 Time Breakdown

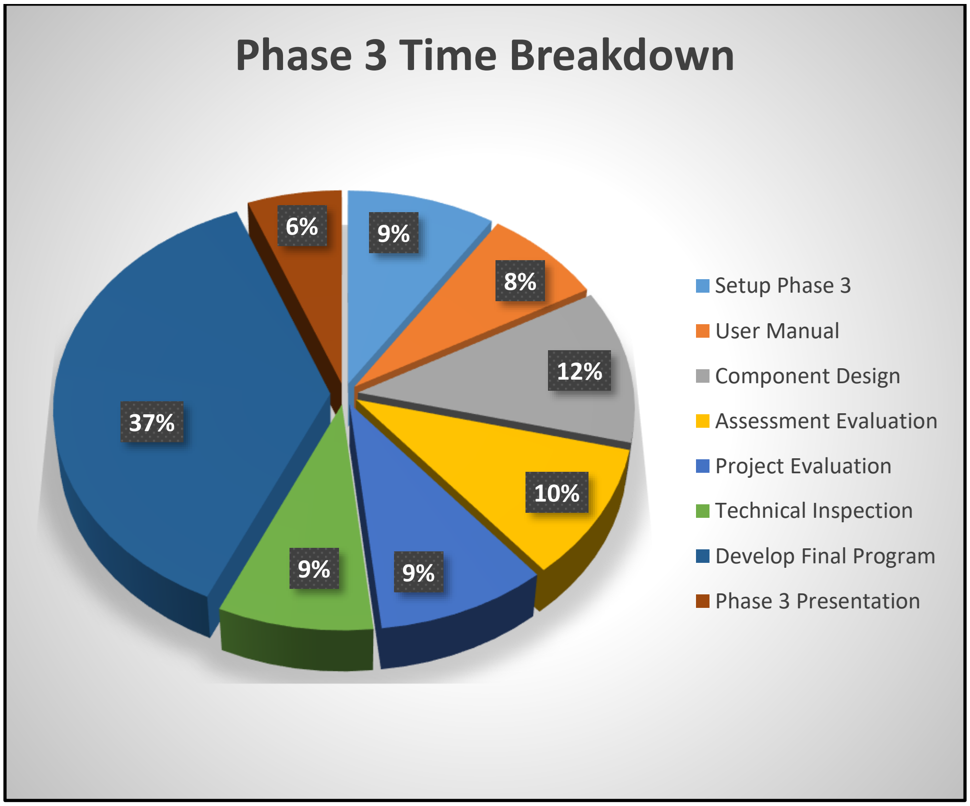


Figure 5. Phase 3 Time Breakdown

3.3 Task Estimation

The following figures compare the individual task estimations to actuals, overall estimates to actuals by phase, and the accuracy of the estimates per phase.

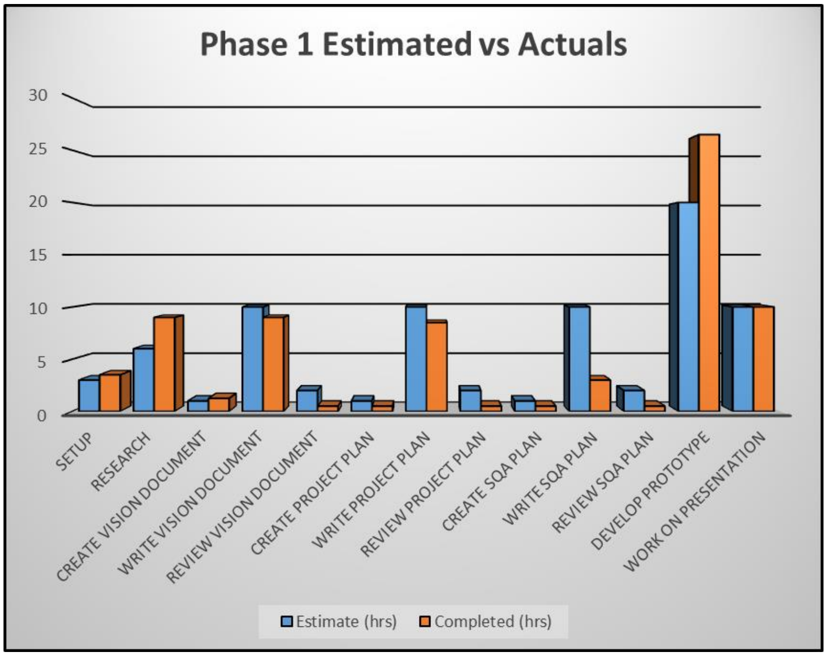


Figure 6. Phase 1 Task Estimates vs Actuals

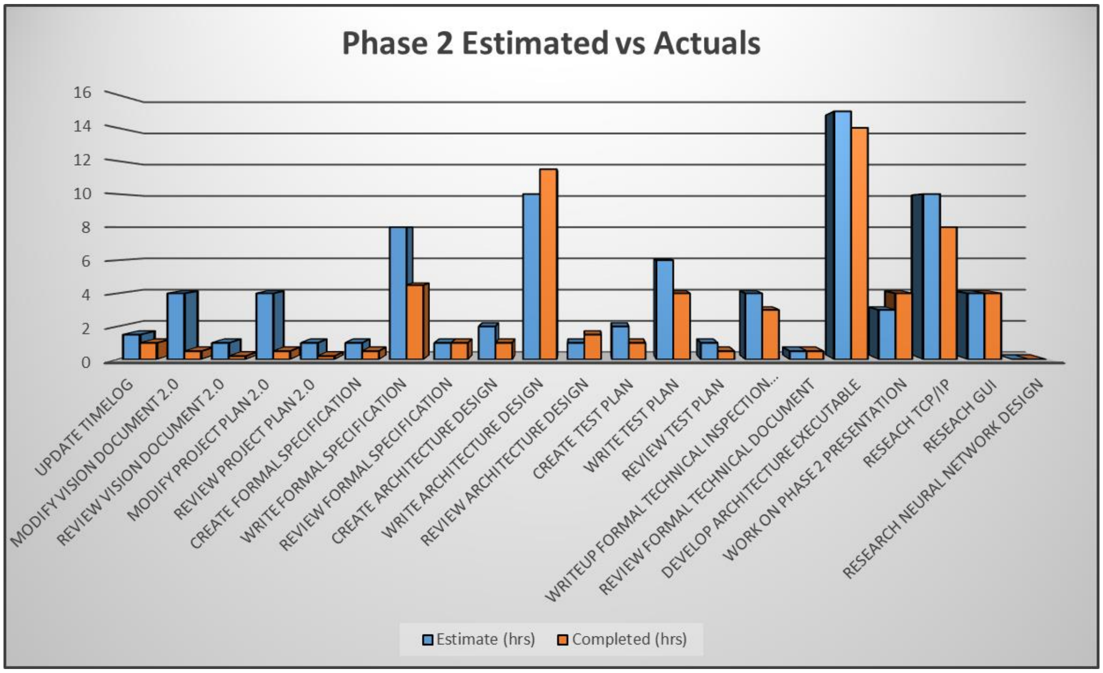


Figure 7. Phase 2 Task Estimates vs Actuals

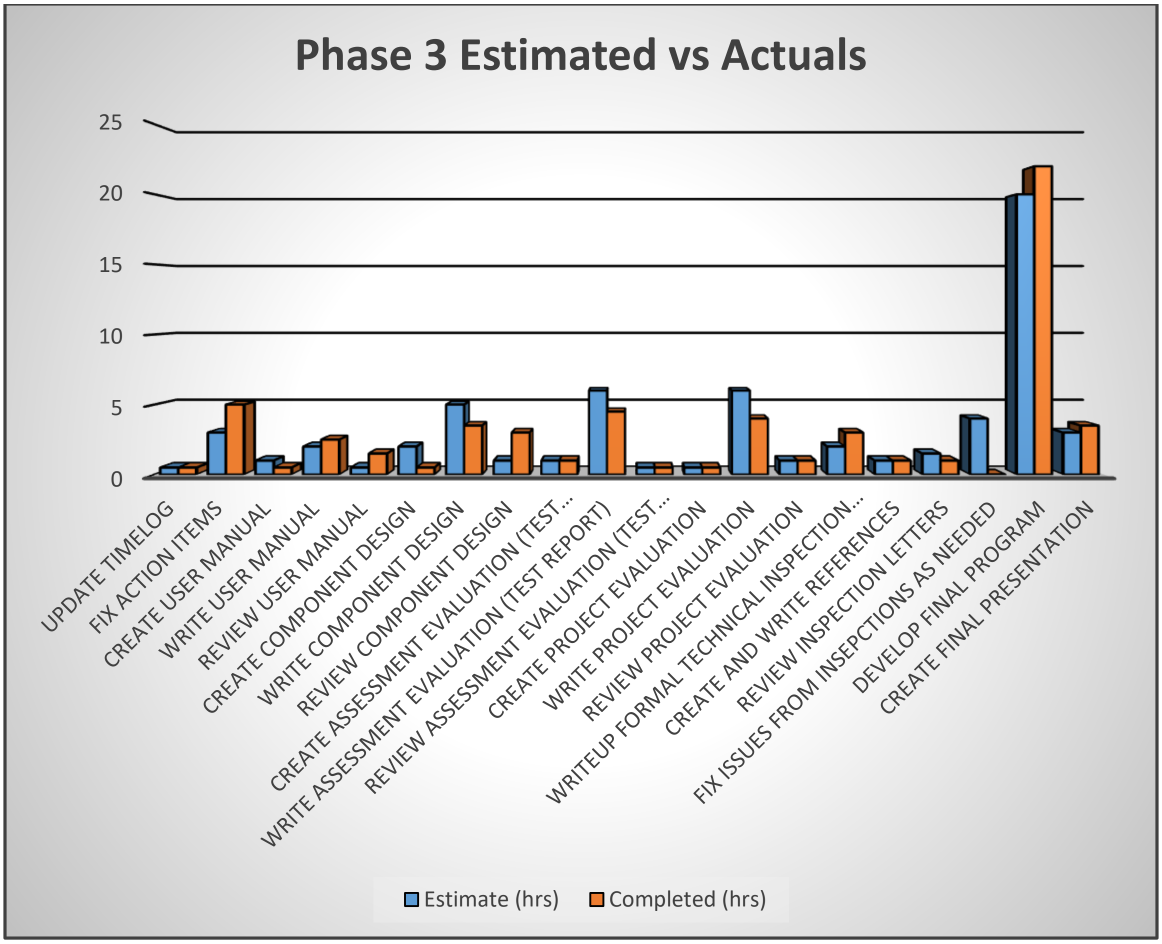


Figure 8. Phase 3 Task Estimates vs Actuals

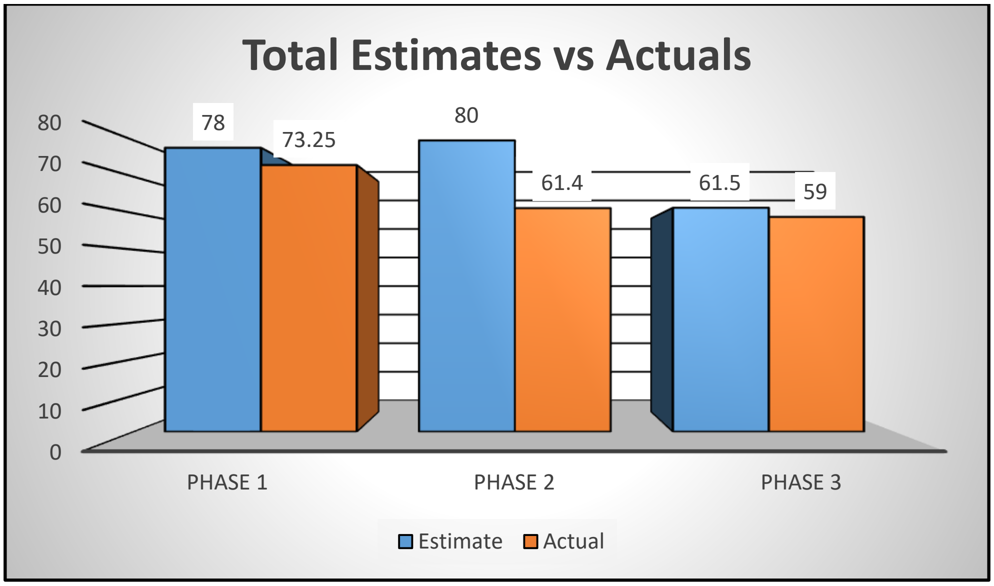


Figure 9. Overall Task Estimates vs Actuals

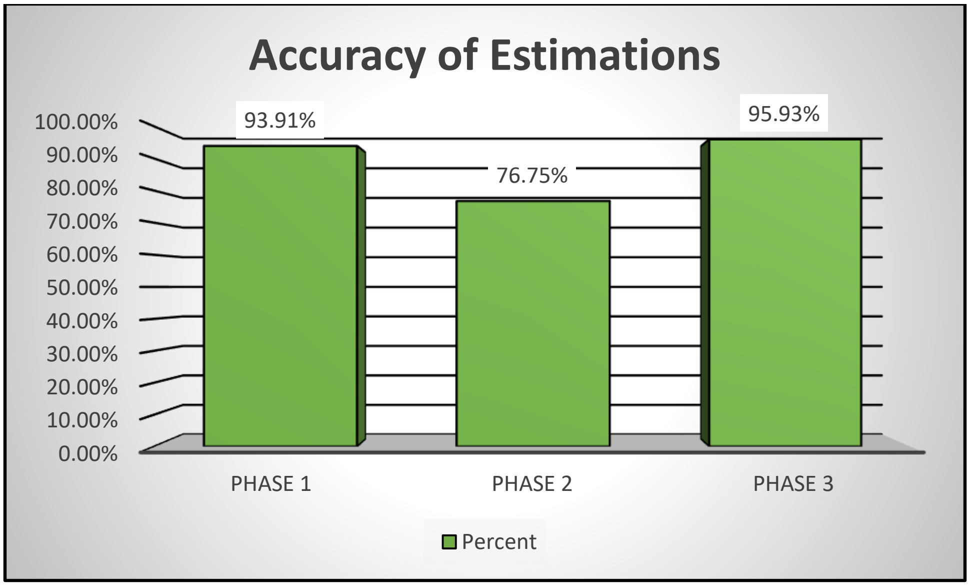


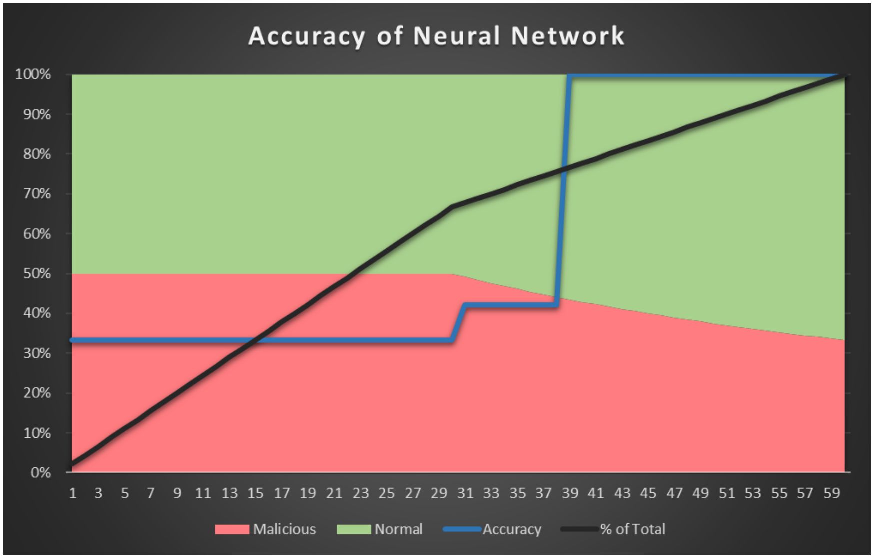
Figure 9. Overall Task Estimates Accuracy

4 Lessons Learned

4.1 Neural Networks

Prior to this project, I knew very little about how neural networks really worked. This project greatly increased my knowledge in this area. I learned how to implement a neural network using matrix multiplication and sigmoid functions. One important point I learned through the development of my neural network is that they are very difficult to make extremely accurate. Much of the accuracy depends on the training data.

4.1. Accuracy of Neural Network



4.2 Accuracy of Neural Network Detail

The Chart above shows the maximum accuracy the Neural Network was able to achieve based on the percentage of values used compared to the total available values. The test iterated over 60 iterations with a normal to malicious ratio equaling 1-to-1 until the 66% mark of the total available values.

4.3 Networking

This project really helped me learn more about TCP/UDP protocols and their associated headers. I knew very little about how networking really worked behind the scenes and all that was involved. This lack of knowledge was one of the reasons I chose to pursue this project.

4.4 DOS Attacks

There are many types of malicious attacks on systems. This project really opened my eyes to the different kinds of Denial of Service (DOS) attacks that are being used by attackers. I learned a couple of types of DOS attacks and how to hopefully spot them using packet headers.

4.5 Real-time computing

One of the most difficult parts of the system was trying to compute the information quickly in a somewhat real-time like method. I was not able to measure any timing information of throughput of a packet, but I found it interesting that the system aimed to achieve this. I focused more energy on a proof of concept rather than making the system more efficient due to timing constraints. I think one thing that would help for future development would be to use a compiled language for faster processing of data.

5 Future Work

5.1 Capture Accurate Training Data

The greatest improvement to the current system would be to capture more accurate data to use to train the neural network. The data that was used was found to be lacking in usefulness late into the project.

5.2 Expert System Tracking

This project could benefit from an expert system tracking connections. The data that was used to train the system contained data that would be useful to an expert system. The expert system would allow the IDS to track connection times and the number of packets sent to be used in the evaluation of malicious attacks.

5.3 Prevent Malicious Attacks

The current system only notifies users of a malicious attack. The system could be extended to also prevent attacks when they are found.

5.4 Feedback loop for Neural Network

The neural network can only be trained by the training data. It would benefit the accuracy of decision making by creating a feedback loop to the neural network of false positive and false negatives. This would allow the neural network to learn from the mistakes it makes to be more accurate in the future.

5.5 Efficiency Improvements

The system is not extremely efficient due to the fact it is written in Python. The system would benefit from being written in a compiled language allowing it to process data more quickly.