Spring 2018

A Study On Effects Of Data Poisoning On HMMs

Rachel Gonsalves
San Jose State University

A Study On Effects Of Data Poisoning On HMMs

A Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Rachel Gonsalves

May 2018

The Designated Project Committee Approves the Project Titled

A Study On Effects Of Data Poisoning On HMMs

by

Rachel Gonsalves

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2018

Dr. Mark Stamp	Department of Computer Science
Dr. Thomas Austin	Department of Computer Science
Fabio Di Troia	Department of Computer Science

ABSTRACT

A Study On Effects Of Data Poisoning On HMMs

by Rachel Gonsalves

With the ever increasing use of burgeoning volumes of data, machine learning systems involving minimal human oversight are crucial for classification and analysis tasks. Machine learning algorithms used for such purposes have revolutionized the way we sort, classify, and analyze data.

The accuracy of any machine learning algorithm depends heavily on the data it is trained on. In some circumstances, an attacker can attempt to poison the training data to subvert a machine learning system. In this research, we analyze the effects of training data poisoning attacks on hidden Markov models (HMMs), in the context of malware classification. With the increase in percentage of data poisoning, HMM is still able to classify most files correctly. Hence we find that HMMs are able to classify at high and low level of poisoning.

ACKNOWLEDGMENTS

I am very grateful to Dr. Stamp for his guidance and patience. I would also like to thank Dr Austin for his invaluable feedback and Fabio Di Troia for his insight and support. I'm also very thankful to my family and friends for their constant encouragement, love and support.

TABLE OF CONTENTS

CHAPTER

1 Introduction 1

2 Background 3

2.1 Hidden Markov Model 3

2.2 Hidden Markov Model for Malware Detection 4

2.3 ROC Curves 4

2.4 Related Work 5

3 Methodology and Results 6

3.1 Methodology 6

3.2 Dataset 6

3.3 Tests 7

3.3.1 Effects of Data Poisoning on Winwebsec 7

3.3.2 Effects of Data Poisoning ZeroAccess 10

3.3.3 Effects of Data Poisoning on Zbot 11

3.3.4 Results for All Malware Families 14

4 Conclusion and Future work 15

4.1 Conclusion 15

4.2 Future Work 15

LIST OF REFERENCES 16

APPENDIX

APPENDIX A 18

A.1 Graphs for Data Poisoning on Winwebsec 18

A.2 Graphs for Data Poisoning on ZeroAccess 36

A.3 Graphs for Data Poisoning on Zbot 54

LIST OF FIGURES

1	Poisoning of Training Set	1
2	Hidden Markov Model	4
3	Winwebsec ROC Curve and AUC with Pure Dataset	8
4	Winwebsec Scatter Plot for with Pure Dataset	9
5	ZeroAccess ROC Curve and AUC with Pure Dataset	10
6	ZeroAccess Scatter Plot for with Pure Dataset	11
7	Zbot ROC Curve and AUC with Pure Dataset	12
8	Zbot Scatter Plot for with Pure Dataset	13
9	Effects of Data Poisoning on All Malware Families	14
A.10	Winwebsec ROC Curve and AUC with 5% Data Poisoning	18
A.11	Winwebsec Scatter Plot for 5% Data Poisoning	19
A.12	Winwebsec ROC Curve and AUC with 10% Data Poisoning	20
A.13	Winwebsec Scatter Plot for 10% Data Poisoning	21
A.14	Winwebsec ROC Curve and AUC with 15% Data Poisoning	22
A.15	Winwebsec Scatter Plot for 15% Data Poisoning	23
A.16	Winwebsec ROC Curve and AUC with 20% Data Poisoning	24
A.17	Winwebsec Scatter Plot for 20% Data Poisoning	25
A.18	Winwebsec ROC Curve and AUC with 25% Data Poisoning	26
A.19	Winwebsec Scatter Plot for 25% Data Poisoning	27
A.20	Winwebsec ROC Curve and AUC with 30% Data Poisoning	28
A.21	Winwebsec Scatter Plot for 30% Data Poisoning	29

A.22 Winwebsec ROC Curve and AUC with 35% Data Poisoning . . . 30
A.23 Winwebsec Scatter Plot for 35% Data Poisoning 31
A.24 Winwebsec ROC Curve and AUC with 40% Data Poisoning . . . 32
A.25 Winwebsec Scatter Plot for 40% Data Poisoning 33
A.26 Winwebsec ROC Curve and AUC with 45% Data Poisoning . . . 34
A.27 Winwebsec Scatter Plot for 45% Data Poisoning 35
A.28 ZeroAccess ROC Curve and AUC with 5% Data Poisoning 36
A.29 ZeroAccess Scatter Plot for 5% Data Poisoning 37
A.30 ZeroAccess ROC Curve and AUC with 10% Data Poisoning . . . 38
A.31 ZeroAccess Scatter Plot for 10% Data Poisoning 39
A.32 ZeroAccess ROC Curve and AUC with 15% Data Poisoning . . . 40
A.33 ZeroAccess Scatter Plot for 15% Data Poisoning 41
A.34 ZeroAccess ROC Curve and AUC with 20% Data Poisoning . . . 42
A.35 ZeroAccess Scatter Plot for 20% Data Poisoning 43
A.36 ZeroAccess ROC Curve and AUC with 25% Data Poisoning . . . 44
A.37 ZeroAccess Scatter Plot for 25% Data Poisoning 45
A.38 ZeroAccess ROC Curve and AUC with 30% Data Poisoning . . . 46
A.39 ZeroAccess Scatter Plot for 30% Data Poisoning 47
A.40 ZeroAccess ROC Curve and AUC with 35% Data Poisoning . . . 48
A.41 ZeroAccess Scatter Plot for 35% Data Poisoning 49
A.42 ZeroAccess ROC Curve and AUC with 40% Data Poisoning . . . 50
A.43 ZeroAccess Scatter Plot for 40% Data Poisoning 51
A.44 ZeroAccess ROC Curve and AUC with 45% Data Poisoning . . . 52

A.45 ZeroAccess Scatter Plot for 45% Data Poisoning 53
A.46 Zbot ROC Curve and AUC with 5% Data Poisoning 54
A.47 Zbot Scatter Plot for 5% Data Poisoning 55
A.48 Zbot ROC Curve and AUC with 10% Data Poisoning 56
A.49 Zbot Scatter Plot for 10% Data Poisoning 57
A.50 Zbot ROC Curve and AUC with 15% Data Poisoning 58
A.51 Zbot Scatter Plot for 15% Data Poisoning 59
A.52 Zbot ROC Curve and AUC with 20% Data Poisoning 60
A.53 Zbot Scatter Plot for 20% Data Poisoning 61
A.54 Zbot ROC Curve and AUC with 25% Data Poisoning 62
A.55 Zbot Scatter Plot for 25% Data Poisoning 63
A.56 Zbot ROC Curve and AUC with 30% Data Poisoning 64
A.57 Zbot Scatter Plot for 30% Data Poisoning 65
A.58 Zbot ROC Curve and AUC with 35% Data Poisoning 66
A.59 Zbot Scatter Plot for 35% Data Poisoning 67
A.60 Zbot ROC Curve and AUC with 40% Data Poisoning 68
A.61 Zbot Scatter Plot for 40% Data Poisoning 69
A.62 Zbot ROC Curve and AUC with 45% Data Poisoning 70
A.63 Zbot Scatter Plot for 35% Data Poisoning 71

CHAPTER 1

Introduction

With the sheer rise in the amount and variety of data being generated, it has become crucial that we have techniques to classify data independent of human supervision. Machine learning techniques prove useful not only in classifying data but also in identifying malware. Malware can affect computers, leak sensitive data [1], cause denial of service attacks and cause much damage to crucial systems in the current world. Many machine learning techniques are being used today to detect malware and prevent attacks. In order to escape detection and carry out attacks successfully, attackers come up with innovative ideas; one such idea is to compromise the training data of a machine learning model, which causes incorrect learning and thus confuses the model and leads to decreased accuracy but with higher and lower levls of posoinonig the model still [2].

The attacks that involve influencing the models can be categorized as [3]:

- Causative
- Exploratory

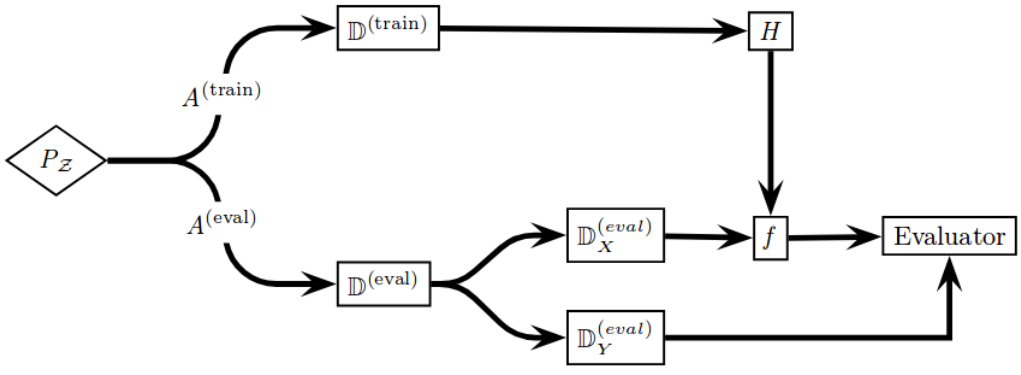

Figure 1: Poisoning of Training Set

The accuracy and efficiency of a machine learning algorithm depends on the training data. When the training set for a machine learning algorithm is poisoned, such an attack is called a causative attack. [3]. Poisoning a model slowly over a period of time is called the Boiling Frog attack [2]. Figure 1 describes how an attack on the training set works [3]. P_z indicates the true distribution. D^{train} and D^{eval} represent the training and testing set respectively. H is the machine learning algorithm.f is the hypothesis that is evaluated by comparing the results. An example of a causative attack against a spam filter is described as follows in [4]: Attackers try and circumvent spam filter re-training by sending non-intrusive traffic, carefully constructed to resemble the upcoming spam. This causes the defending filter to be mis-trained which results in an inability to effectively block the spam. This would be as follows: the spam sales pitch "You need a new phone? Really, do buy now!" is recast as "Do you really need buy a new phone, now!?"; while both these phrases have markedly different meanings they are treated the same by the spam filter. [4] In an exploratory attack, the attacker observes the effects of instances designed for the learning model but does not directly influence the learning [5] [2].

The goal of a malware detector is to ensure secure learning [3]. In the current experiment, we check how susceptible HMMs are to data poisoning. For a given training set, when data is poisoned, the model is tested on a benign and malware test data. When the model training set is poisoned, we quantify the change in performance of the model.

The remainder of this paper is structured as follows. Relevant background topics are discussed in Chapter 2. This chapter includes an introduction to Hidden Markov Models(HMMs), on which our training and testing methods and some related work. In Chapter 3, we discuss the implementation, present the results and observations. Chapter 4 contains the conclusion and a brief discussion of future work.

CHAPTER 2

Background

2.1 Hidden Markov Model

Hidden Markov model is a machine learning technique which is used for statistical pattern analysis [6]. To get a deeper understanding of the concept and an overview of the terminology used, we look at an example from the paper 'A Revealing Introduction to Hidden Markov Models' [7] and understand the terminologies. Suppose we want to predict the temperatures of some 100 years ago and know that relation between the size of growth of tree rings and the weather. The weather is categorized to hot(H) and cold(C). The sizes of tree rings to small(S), medium(M) and large(L). Given an observation of tree ring sizes we try to find whether it was hot or cold during the growth of the ring. These states are unknown thus known as hidden states. The model is represented by observation matrix, transition matrix and the initial state distribution.

The transition matrix A is a $N * N$ matrix where is N is the number of states, observation matrix is $N * M$ where M is the number of observed symbols. Initial state distribution is the probability of starting at any given state and is give by a list of size N, these represent probabilities corresponding to each state. Each matrix is row stochastic. In the Figure 2 [7], $X_0 - X_n$ represents the hidden states.

T = length of the observation sequence

N = number of states in the model

M = number of observation symbols

$Q = q_0, q_1, ..., q_{N-1}$ = distinct states of the Markov process

$V = 0, 1, ..., M - 1$ = set of possible observations

A = state transition probabilities

B = observation probability matrix

π = initial state distribution

$O = (O_0, O_1, ..., O_{T-1})$ = observation sequence.

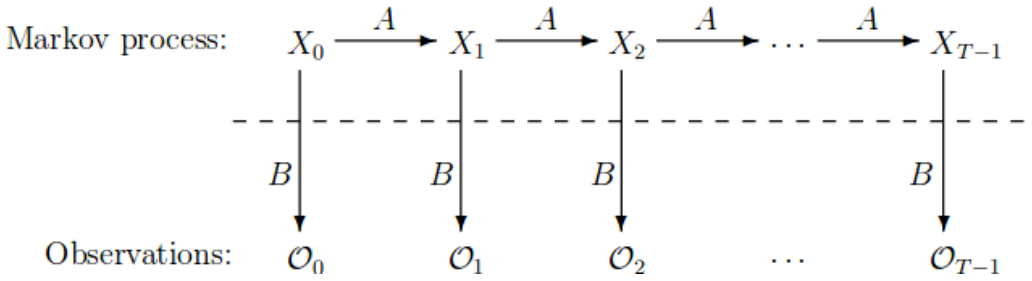

Figure 2: Hidden Markov Model

HMM is used to solve mainly three problems:

Problem 1: Given a sequence of observation to find the probability of the observed sequence

Problem 2: Finding the state sequence that best fits the given model.

Problem 3: To find a model that fits best the observed data [7].

2.2 Hidden Markov Model for Malware Detection

For malware classification in HMM, the model is trained on the opcode sequence of the malware. This type of analysis is known as static analysis because it does not involve execution and monitoring [8]. Malware detection with the help of HMM is statistical as it trains on statistical features [9].

2.3 ROC Curves

For Receiver Operating Classifier(ROC) curves, we need to calculate the True Positive Rate(TPR) and False Positive Rate(FPR).For a given classifier, we need to understand that samples can be classified in 4 ways [10]. When a given malware

sample is correctly classified as malware, it is considered to be a true positive. When a malware sample is classified as benign it is a false negative. When a sample is benign and classified as benign it is a true negative.When a benign sample is wrongly classified as malware it is False Positive. TPR is the total number of true positives upon the total number of positive samples, TPR also termed as the sensitivity [11]. A True Negative Rate(TNR) is the total number of true negatives up the total number of benign samples and is known as the specificity. [12]. The x-axis of the ROC curve represents the FPR and y-axis represents the TPR. The threshold passes through the point (TPR,FPR) [10].

2.4 Related Work

When it comes to data poisoning, a lot of work has been done to observe its effects on various machine learning techniques. Data poisoning attacks mostly occur when a system is adaptive [13]. It is also most common when data is gathered from unreliable sources. One of the methods used to improve the efficiency and decrease the model vulnerability is that the model can be trained to reject a sample which causes a decrease in efficiency. The sample would be considered as an outlier. This technique is also known as Reject On Negative Impact(RONI) or data sanitization [14] [13]. Blacklisting and white listing sources requires a lot of effort and it also blocks some good traffic. This does not prove to be effective in the long run either [15]. Another method, weighted bagging, was used for the training data; this helps make the model robust against such attacks [16] [5] . The concepts of data sanitization, weighted bagging and defining upper and lower bounds for testing data are useful when dealing with adversarial machine learning and are decribed in detail in [17] [18].

CHAPTER 3

Methodology and Results

3.1 Methodology

Hidden Markov models are useful in classification of malware. HMM is used to distinguish between malware and benign files. [7] For the detection of malware, the model is trained on the opcode sequence. This trained model is tested on opcode sequence of benign files as well as malware. For the purpose of this experiment, the HMM is trained on 100 files each of 3 malware families. The A, B and π matrix are initalized to around $1/N$, $1/M$ and $1/N$ per row respectively. The 26 most frequently occuring opcodes were mapped and the rest were considered to be space. For the current model the values were initialized as follows: $N = 2$, $M = 27$. M is 27 for the number of opcodes and space inserted for every opcode not in the 26. N is 2 as it gave the best classification after experimenting with N ranging 2 to 6. For each training 100 files are used.

3.2 Dataset

In order to test the effects of data poisoning on the model, the model was initially trained on a pure training set so the changes in the efficiency of the model can be observed once the data is poisoned. For the purpose of this experiment, malware files from the Malicia dataset were used. The model was trained on 3 malware families based on the number of samples available in the training set: Winwebsec [19], Zbot [20] and Zeroaccess [21].

For each malware family, the training set and test set contained 100 files. Once the model was trained and tested, the data was poisoned gradually by adding files one by one to the training set, training and then testing the model. The decrease in efficiency of the model is checked by evaluating the Area Under the Curve (AUC).

3.3 Tests

This section includes the results of data poisoning on 3 malware families. 100 files from each malware family were selected randomly. For the purpose of testing, a sequence with $T = 15000$ was used from each sample file to score the samples.

3.3.1 Effects of Data Poisoning on Winwebsec

The model was trained on 100 files from the Winwebsec malware family, and tested on 100 samples of malware and benign files, the training data was poisoned with the addition of one malware file and was tested again. From Figure 3 to Figure A.27, we see the scatter plots and ROC curve of the tested data. In the results we see that the AUC of the model decreases with addition of each benign sample. In the pure training set, an AUC of 0.64 is observed 3, which gradually goes to 0.95 A.22 with a data poisoning of 45%

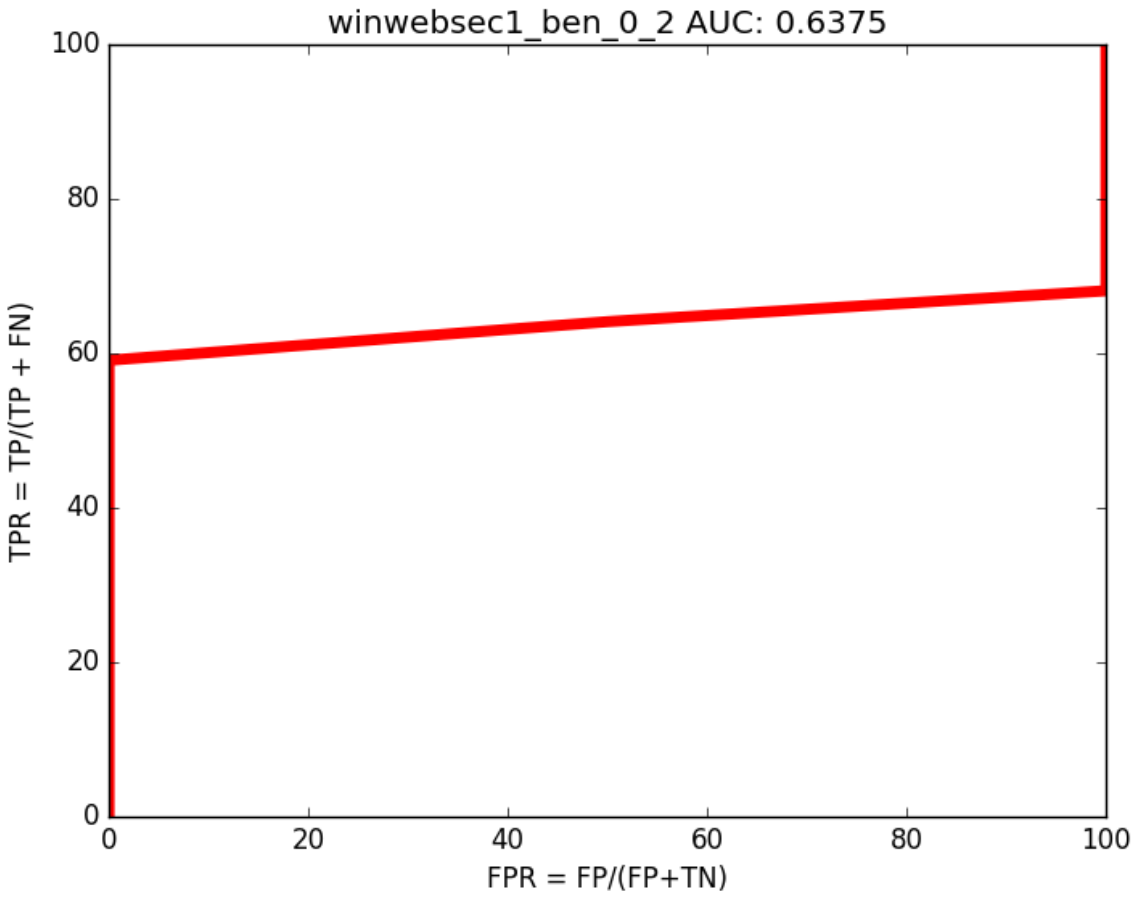

Figure 3: Winwebsec ROC Curve and AUC with Pure Dataset

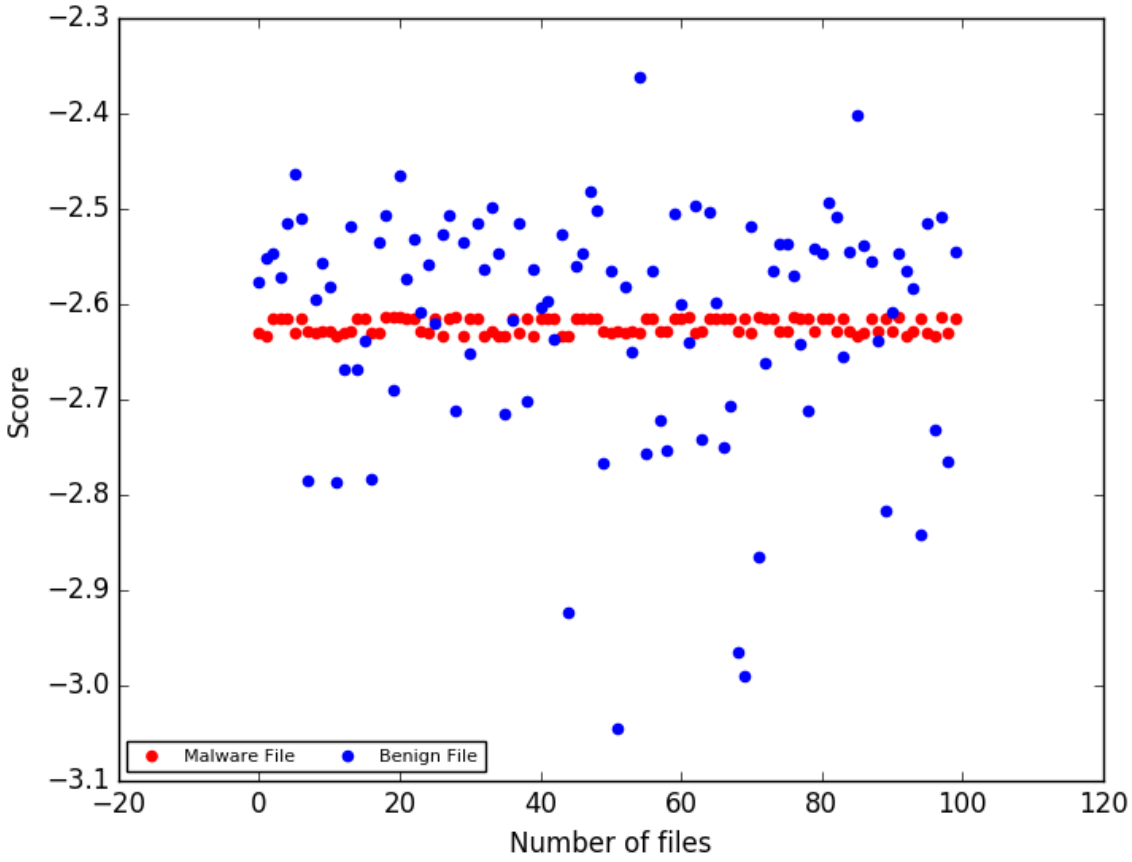

Figure 4: Winwebsec Scatter Plot for with Pure Dataset

3.3.2 Effects of Data Poisoning ZeroAccess

The model was trained on 100 files from the ZeroAccess malware family, and tested on 100 samples of malware and benign files, the training data was poisoned with the addition of one benign file after each test and was tested again. From Figure 5 to Figure A.41, we see the scatter plots and ROC curve of the tested data. The results were evident as we see that the AUC of the model decreases with addition of each benign sample. The results were evident as we see that the AUC of the model decreases with addition of each benign sample. In the pure training set, an AUC of 0.93 is observed Figure 5, which gradually reduces to 0.19 which can be reversed to an auc of 0.81 Figure A.40 with a data poisoning of 45%

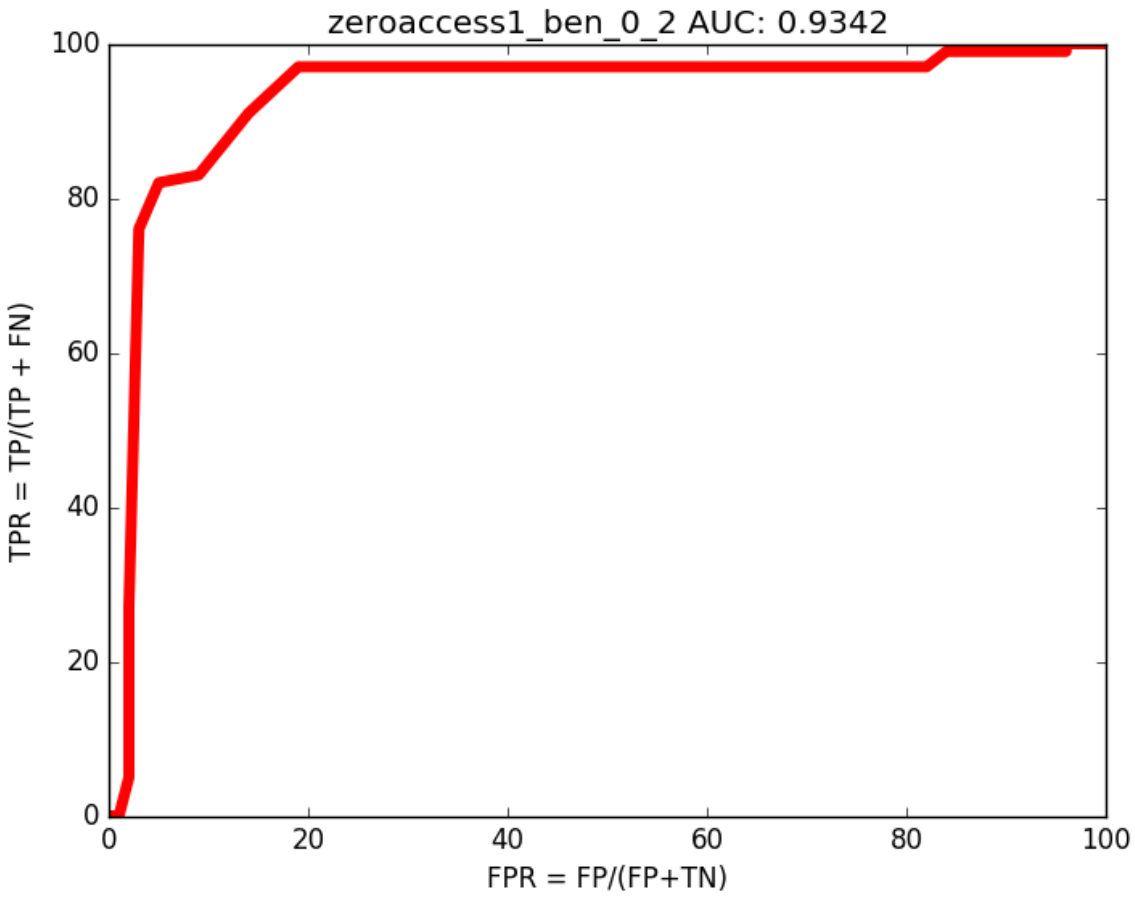

Figure 5: ZeroAccess ROC Curve and AUC with Pure Dataset

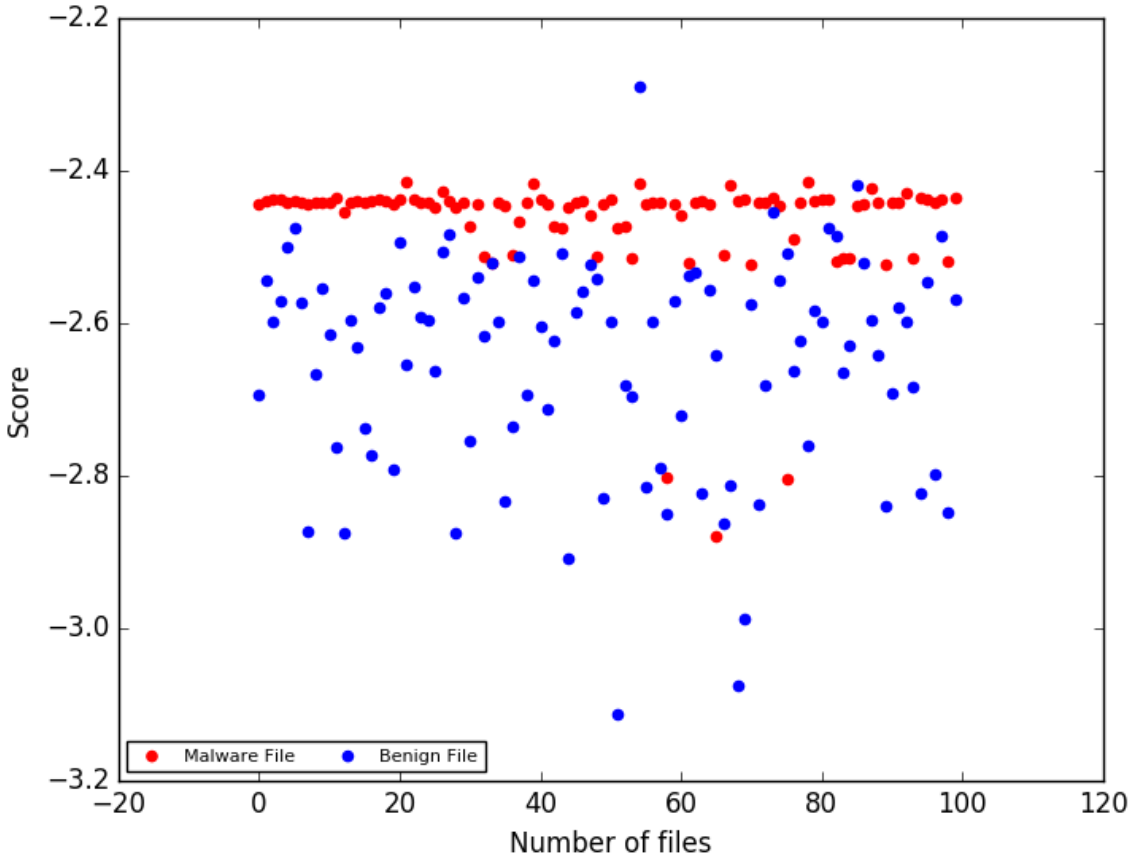

Figure 6: ZeroAccess Scatter Plot for with Pure Dataset

3.3.3 Effects of Data Poisoning on Zbot

The model was trained on 100 files from the Zbot malware family, and tested on 100 samples of malware and benign files, the training data was poisoned with the addition of one benign file at a time and was tested again. From Figure 7 to Figure A.63, we see the scatter plots and ROC curve of the tested data. In the results we see that the AUC of the model does not decreaset with addition of each benign sample. In the pure training set, an AUC of 0.71 is observed Figure 7, which gradually increaseses to 0.79 Figure A.58 with a data poisoning of 45%

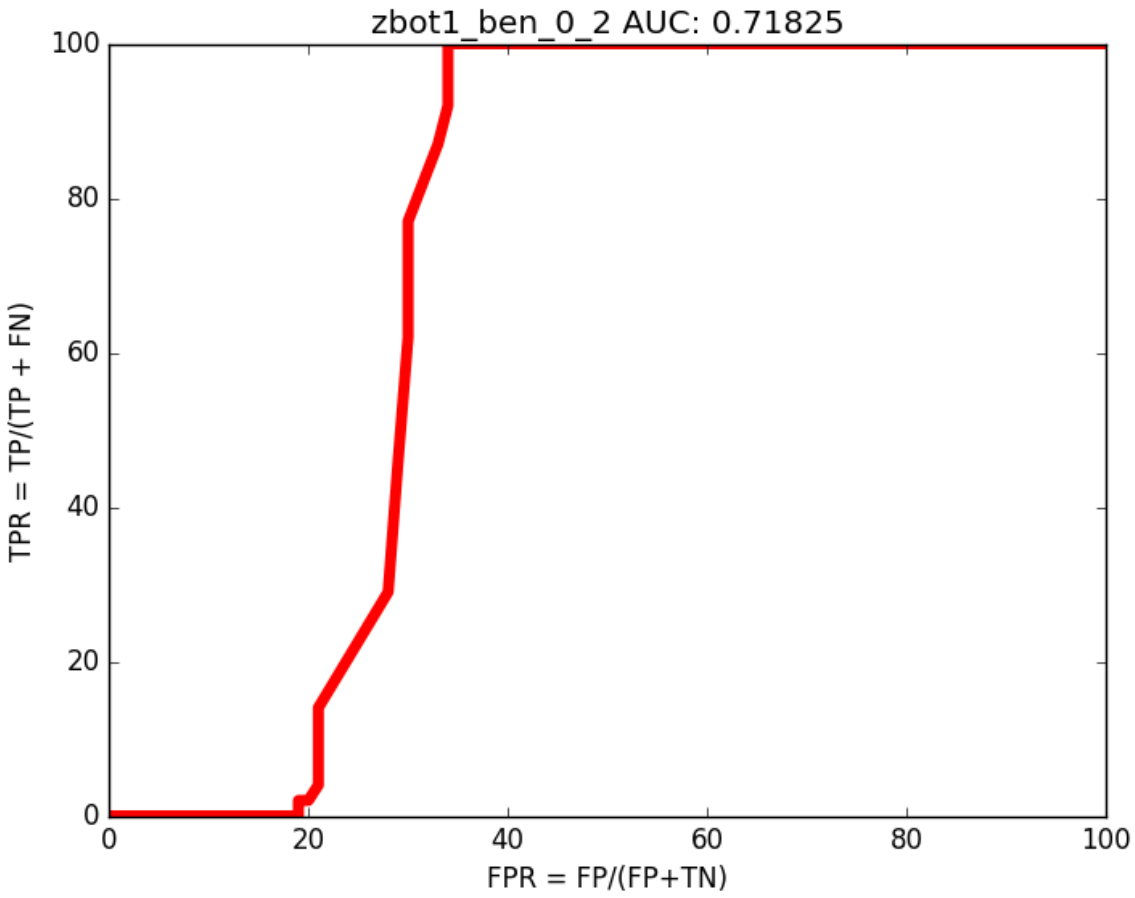

Figure 7: Zbot ROC Curve and AUC with Pure Dataset

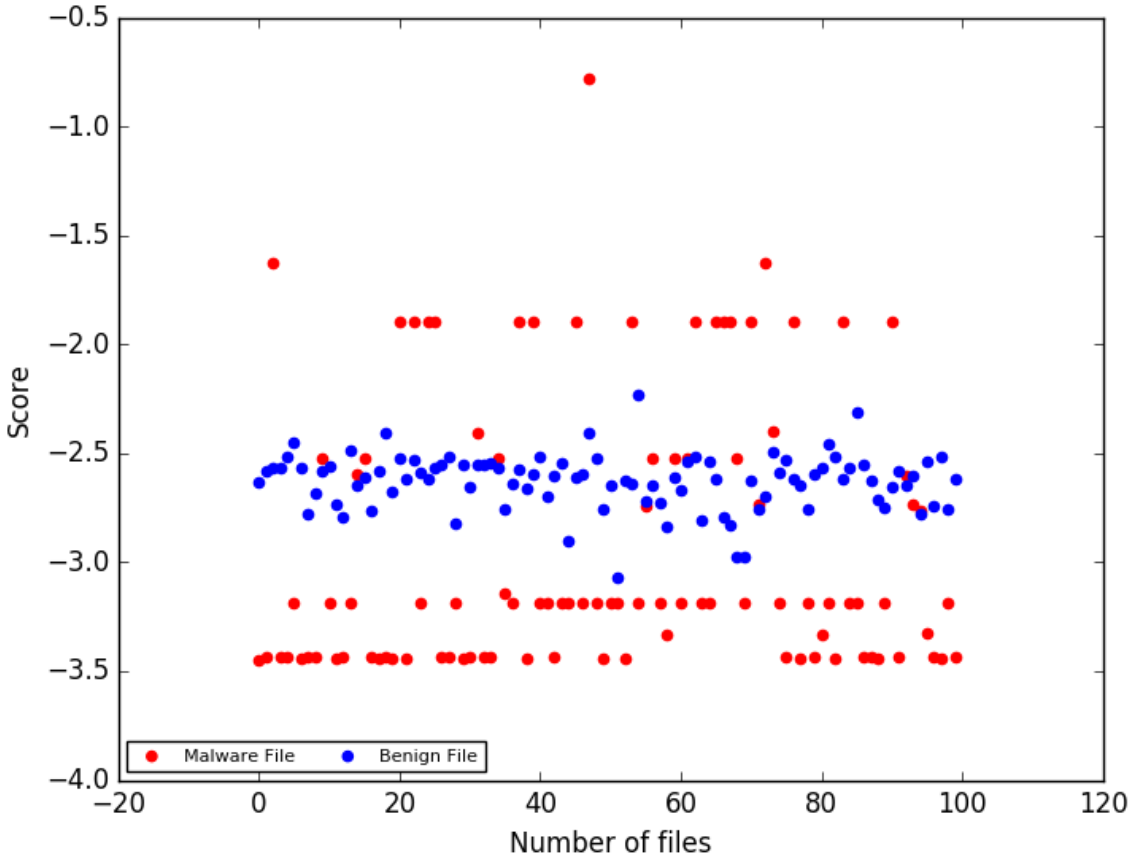

Figure 8: Zbot Scatter Plot for with Pure Dataset

3.3.4 Results for All Malware Families

The following Figure 9 is a graphical representation of the change in AUC with the increase in percentage of data poisoning for all malware families.The x axis represents the AUC and the y axis represents the percentage of data poisoning.

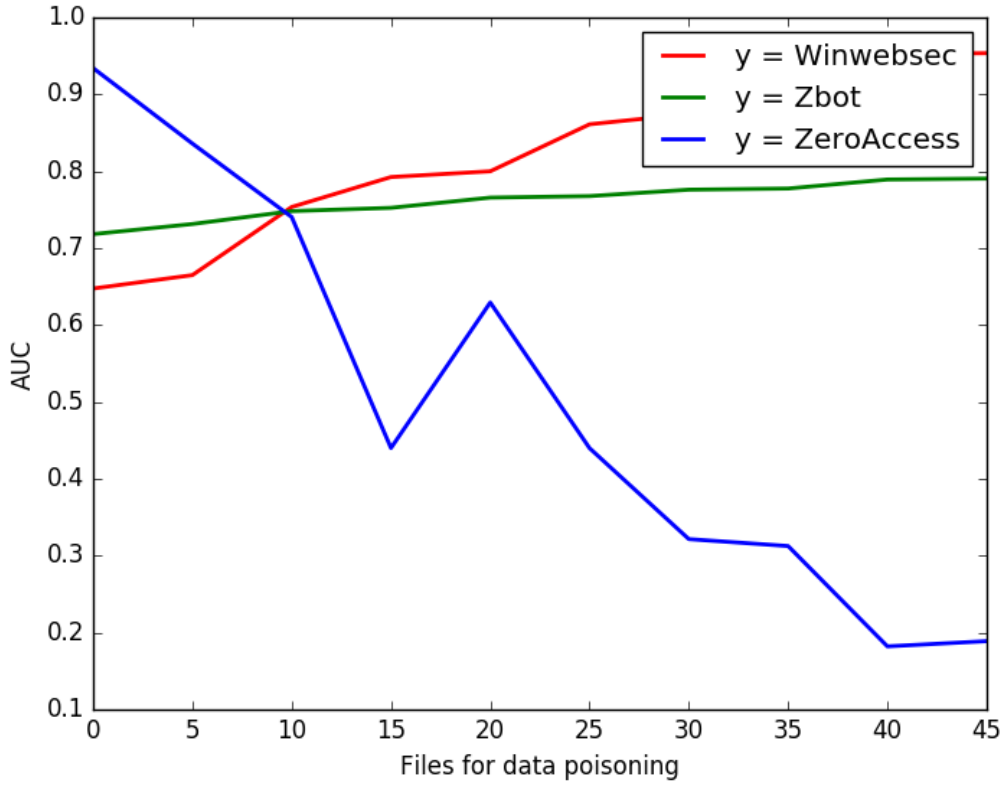

Figure 9: Effects of Data Poisoning on All Malware Families

CHAPTER 4

Conclusion and Future work

4.1 Conclusion

HMM was trained on three malware families, these trained models were tested on malware and benign samples. The models trained on pure datasets performed well. Multiple HMM models were trained with training data poisoned from 0% to 45% for each family incrementing the poisoning by 5. As the percentage of poisoning increased, the model was still able to classify the files at a higher and lower level of poisnoning.

4.2 Future Work

HMMs trained on API calls as features are stronger and hence it would be a good comparison to see how HMMs trained dynamically perform with this type of an attack

LIST OF REFERENCES

[1] C. Annachhatre, T. Austin, and M. Stamp, ‘‘Hidden markov models for malware classification,’’ vol. 11, 05 2014.

[2] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, ‘‘Adversarial machine learning,’’ in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, ser. AISec ’11. New York, NY, USA: ACM, 2011, pp. 43--58. [Online]. Available: http://doi.acm.org/10.1145/2046684.2046692

[3] M. Barreno, B. A. Nelson, A. D. Joseph, and D. Tygar, ‘‘The security of machine learning,’’ EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2008-43, Apr 2008. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-43.html

[4] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, ‘‘The security of machine learning,’’ *Machine Learning*, vol. 81, no. 2, pp. 121--148, Nov 2010. [Online]. Available: https://doi.org/10.1007/s10994-010-5188-5

[5] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, ‘‘Can machine learning be secure?’’ in *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS ’06. New York, NY, USA: ACM, 2006, pp. 16--25. [Online]. Available: http://doi.acm.org/10.1145/1128817.1128824

[6] L. R. Rabiner, ‘‘A tutorial on hidden markov models and selected applications in speech recognition,’’ *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257--286, Feb 1989.

[7] M. Stamp, ‘‘A revealing introduction to hidden markov models,’’ 2015. [Online]. Available: http://www.cs.sjsu.edu/faculty/stamp/RUA/HMM.pdf

[8] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, ‘‘A survey on automated dynamic malware-analysis techniques and tools,’’ *ACM Comput. Surv.*, vol. 44, no. 2, pp. 6:1--6:42, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/2089125.2089126

[9] W. Wong and M. Stamp, ‘‘Hunting for metamorphic engines,’’ *Journal in Computer Virology*, vol. 2, no. 3, pp. 211--229, Dec 2006. [Online]. Available: https://doi.org/10.1007/s11416-006-0028-7

[10] M. Stamp, *Introduction to Machine Learning with Applications in Information Security*. Chapman and Hall/CRC, 2017.

[11] T. Fawcett, ‘‘An introduction to ROC analysis,’’ *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861--874, Jun 2006.

[12] A. P. Bradley, ‘‘The use of the area under the ROC curve in the evaluation of machine learning algorithms,’’ *Pattern Recognit.*, vol. 30, no. 7, pp. 202,258, Jul 1997.

[13] B. Lagesse, C. Burkard, and J. Perez, ‘‘Securing pervasive systems against adversarial machine learning,’’ in *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, March 2016, pp. 1--4.

[14] B. Nelson, M. Barreno, F. Jack Chi, A. D. Joseph, B. I. P. Rubinstein, U. Saini, C. Sutton, J. D. Tygar, and K. Xia, *Misleading Learners: Co-opting Your Spam Filter*. Boston, MA: Springer US, 2009, pp. 17--51. [Online]. Available: https://doi.org/10.1007/978-0-387-88735-7_2

[15] M. Kantarciouglu, B. Xi, and C. Clifton, ‘‘Classifier evaluation and attribute selection against active adversaries,’’ *Data Min. Knowl. Discov.*, vol. 22, no. 1-2, pp. 291--335, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1007/s10618-010-0197-3

[16] B. Biggio, I. Corona, G. Fumera, G. Giacinto, and F. Roli, ‘‘Bagging classifiers for fighting poisoning attacks in adversarial classification tasks,’’ in *Multiple Classifier Systems*, C. Sansone, J. Kittler, and F. Roli, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 350--359.

[17] A. D. Shieh and D. F. Kamm, ‘‘Ensembles of one class support vector machines,’’ in *Multiple Classifier Systems*, J. A. Benediktsson, J. Kittler, and F. Roli, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 181--190.

[18] J. Steinhardt, P. W. Koh, and P. Liang, ‘‘Certified defenses for data poisoning attacks,’’ *CoRR*, vol. abs/1706.03691, 2017. [Online]. Available: http://arxiv.org/abs/1706.03691

[19] ‘‘Win32/Winwebsec threat description - Windows Defender Security Intelligence.’’ [Online]. Available: https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Win32%2FWinwebsec

[20] ‘‘Trojan.Zbot | Symantec.’’ [Online]. Available: https://www.symantec.com/security_response/writeup.jsp?docid=2010-011016-3514-99

[21] ‘‘Trojan.Zeroaccess | Symantec.’’ [Online]. Available: https://www.symantec.com/security_response/writeup.jsp?docid=2011-071314-0410-99

APPENDIX

APPENDIX A

A.1 Graphs for Data Poisoning on Winwebsec

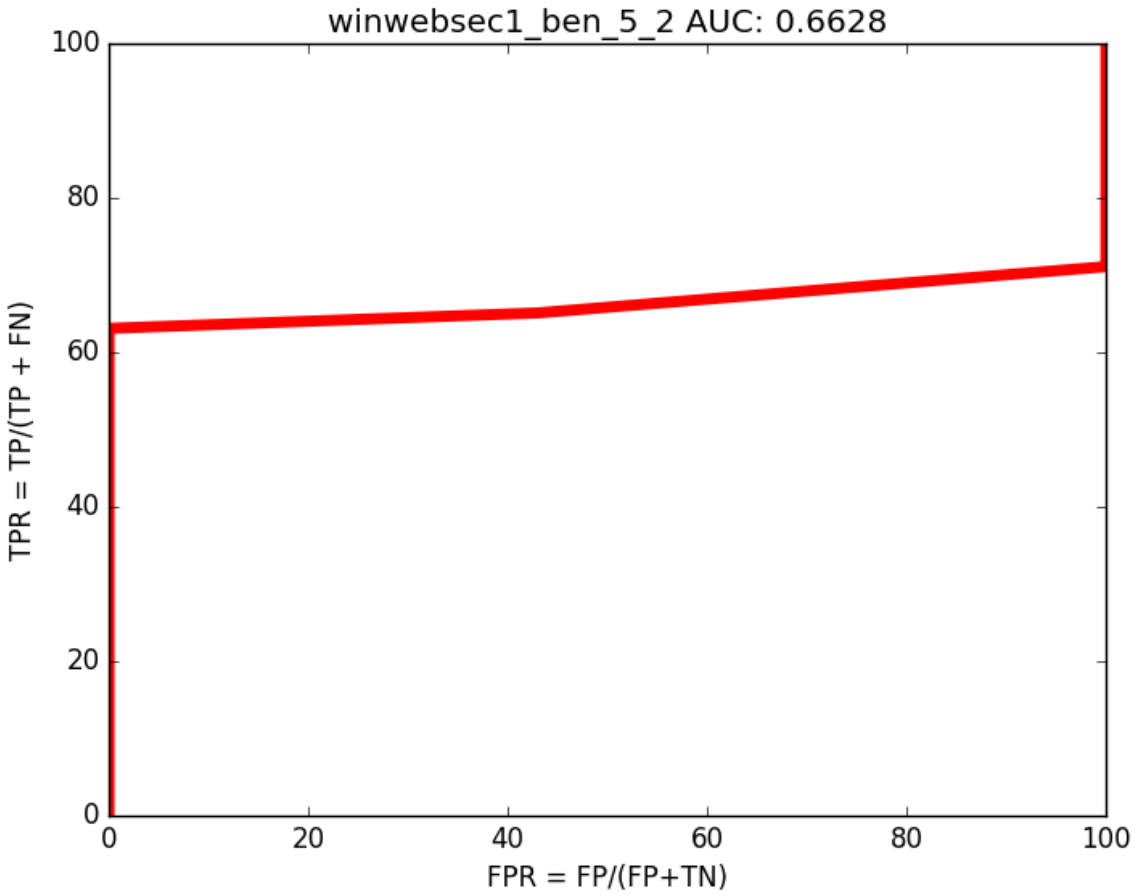

Figure A.10: Winwebsec ROC Curve and AUC with 5% Data Poisoning

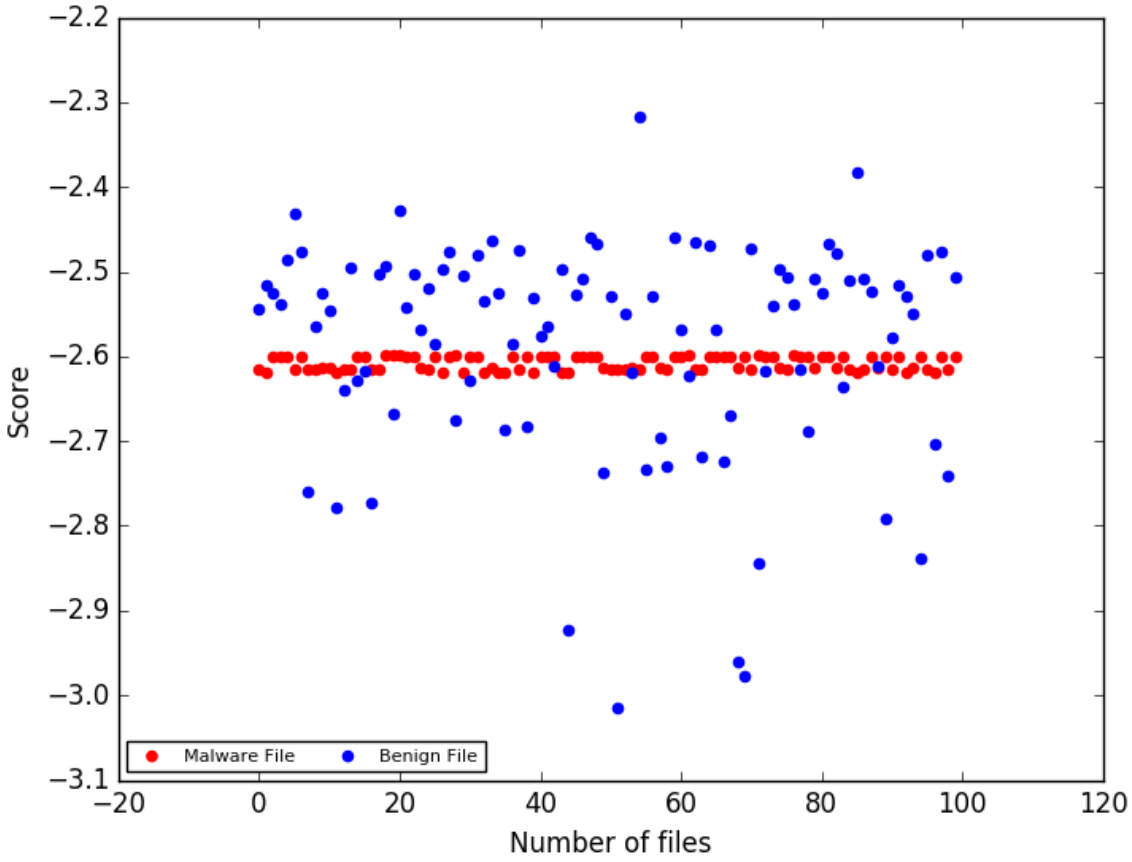

Figure A.11: Winwebsec Scatter Plot for 5% Data Poisoning

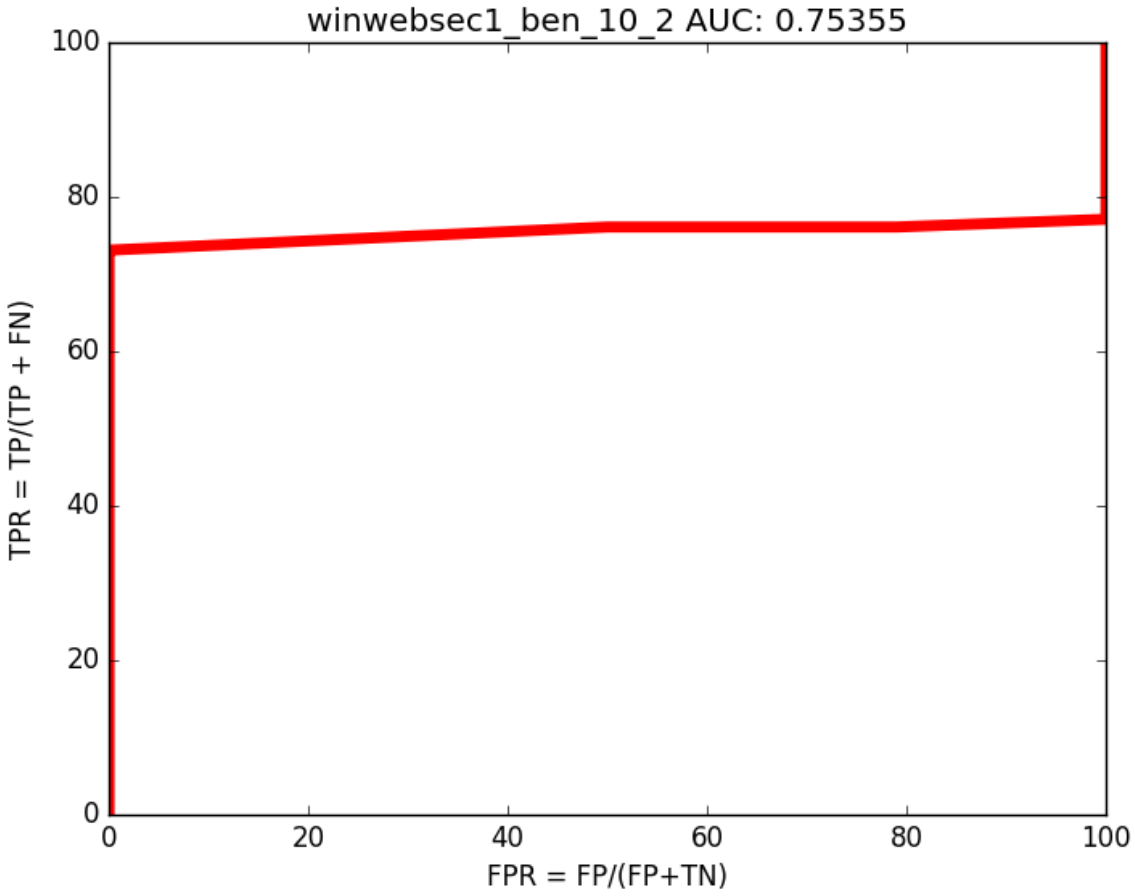

Figure A.12: Winwebsec ROC Curve and AUC with 10% Data Poisoning

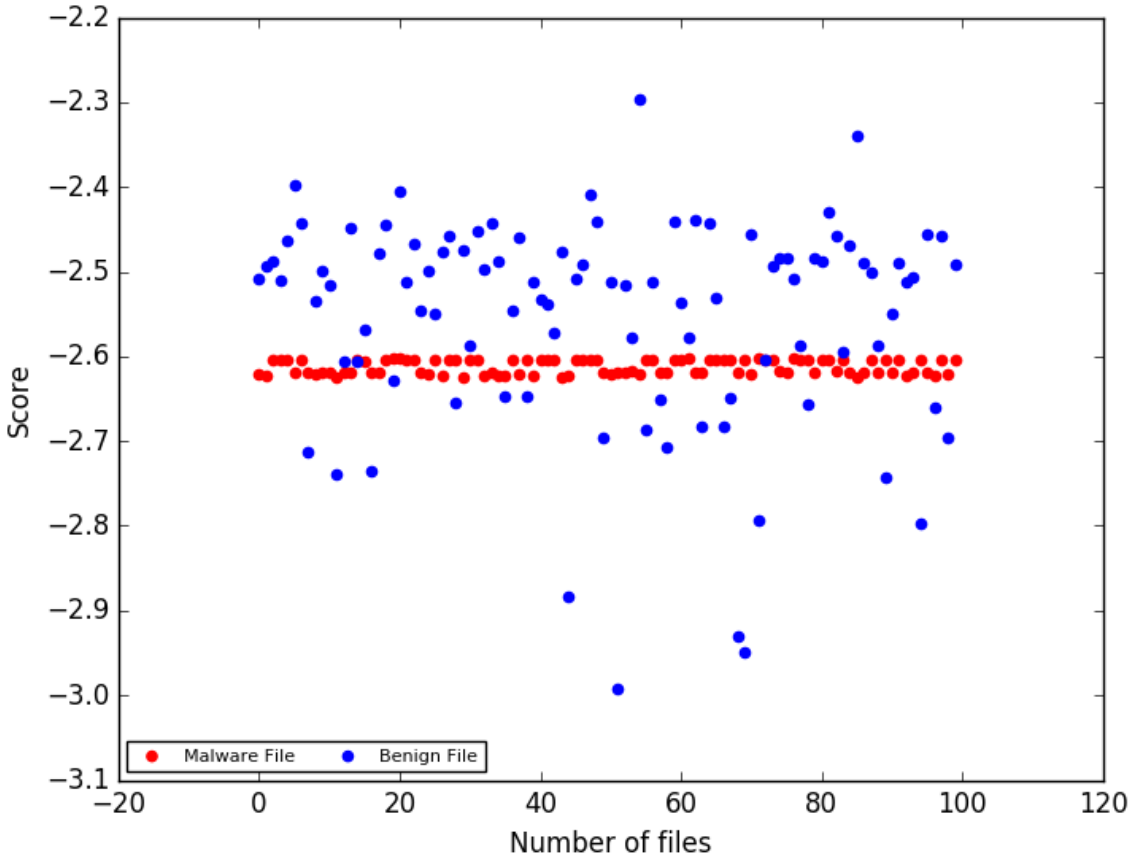

Figure A.13: Winwebsec Scatter Plot for 10% Data Poisoning

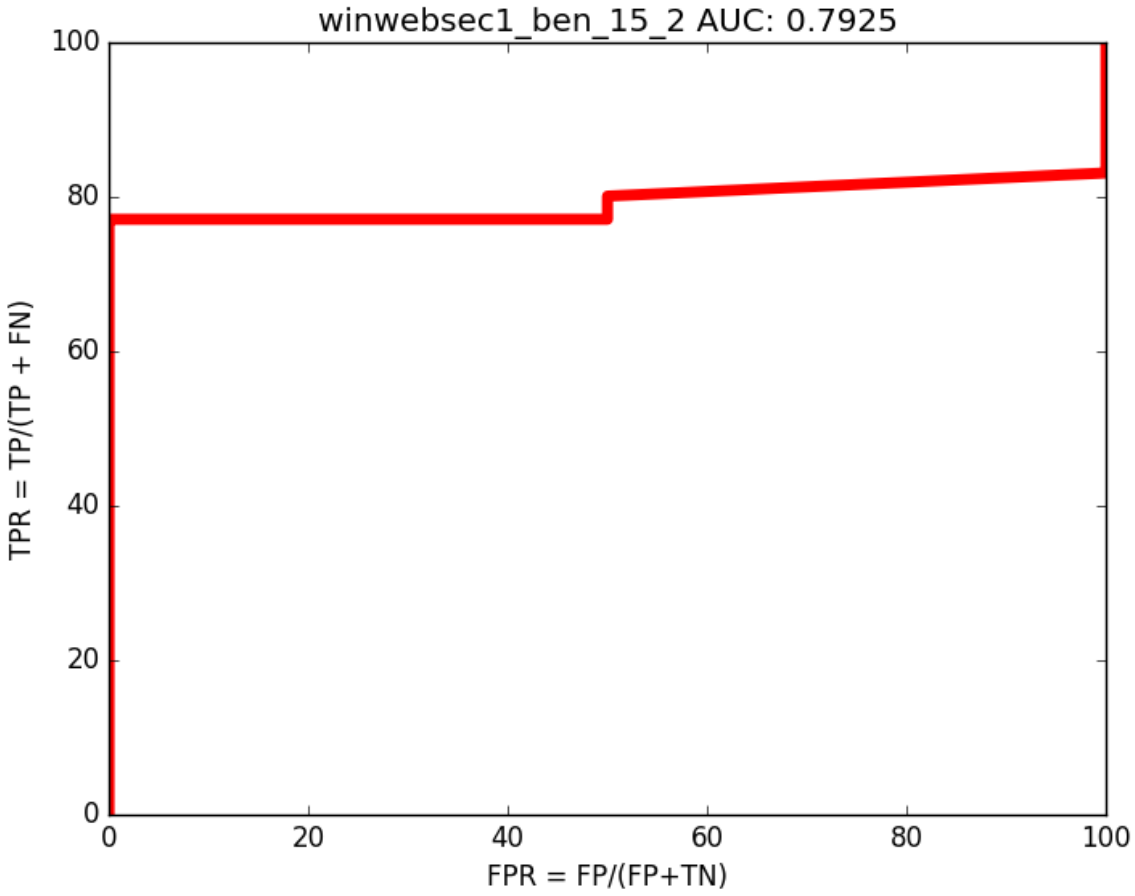

Figure A.14: Winwebsec ROC Curve and AUC with 15% Data Poisoning

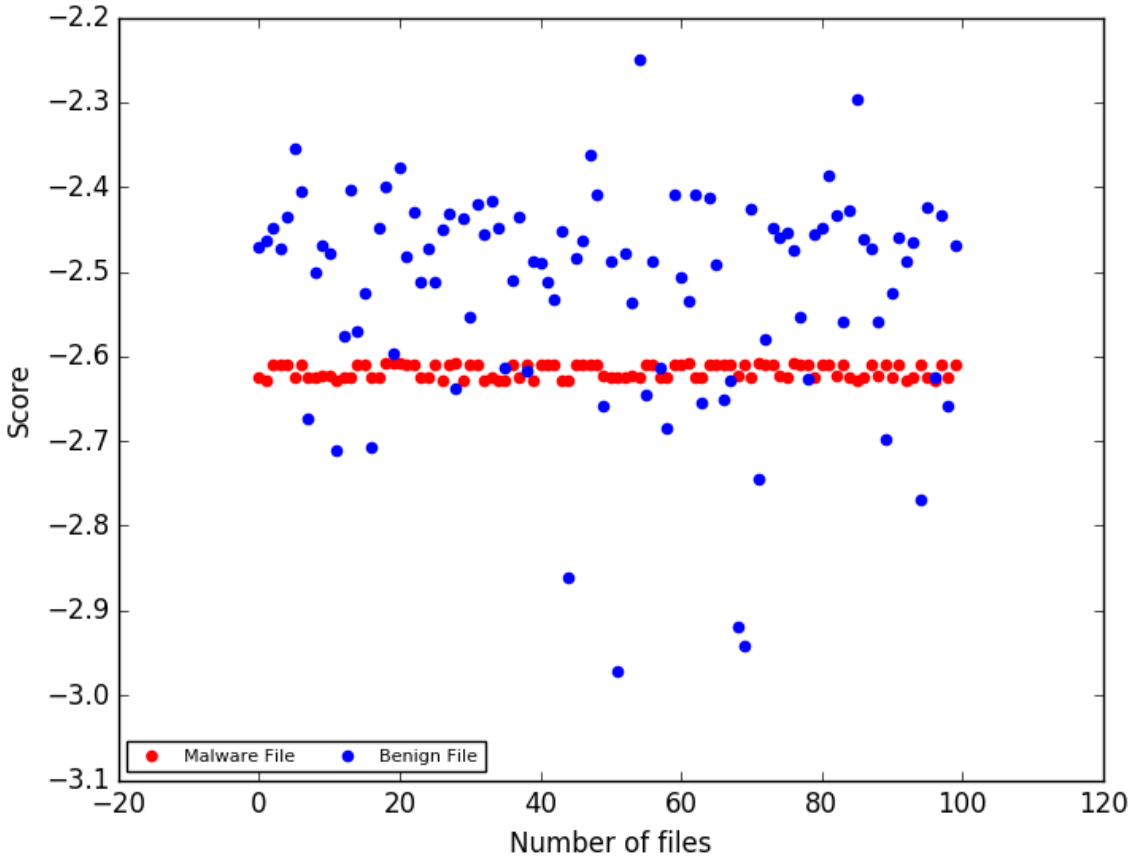

Figure A.15: Winwebsec Scatter Plot for 15% Data Poisoning

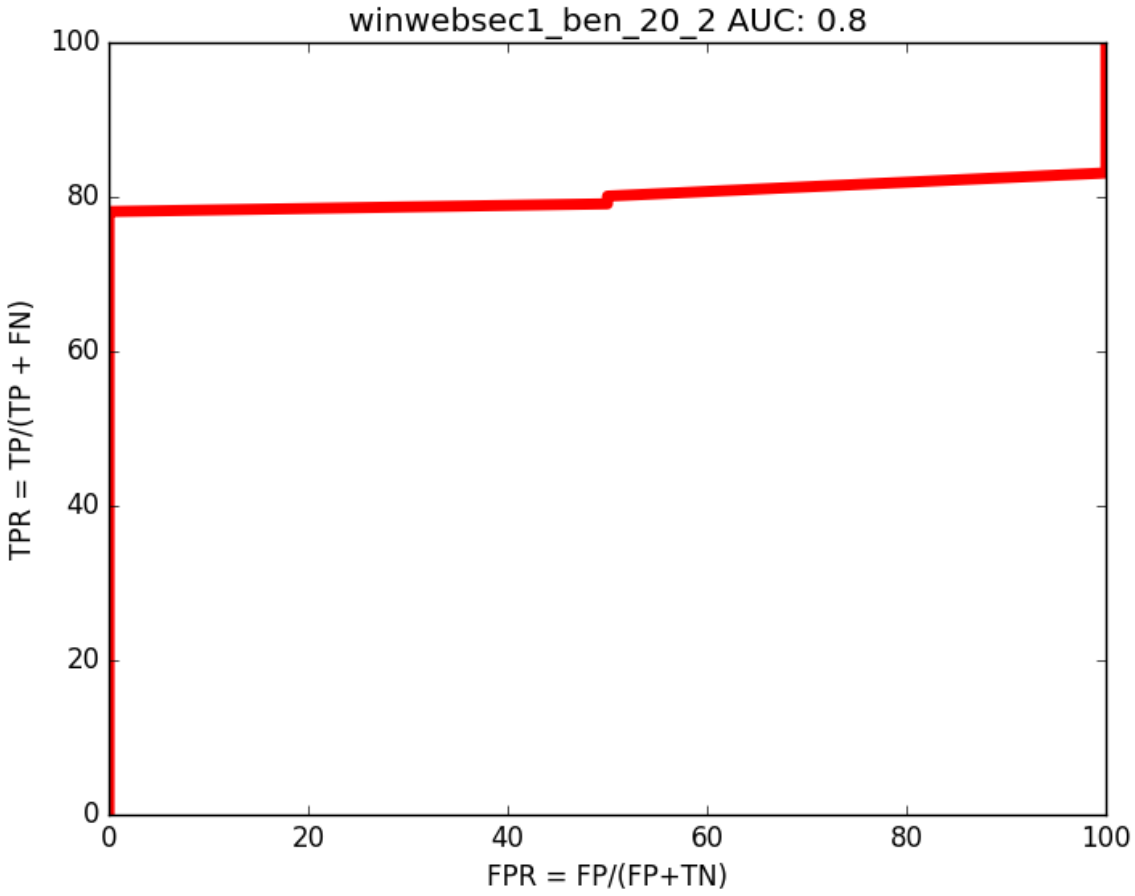

Figure A.16: Winwebsec ROC Curve and AUC with 20% Data Poisoning

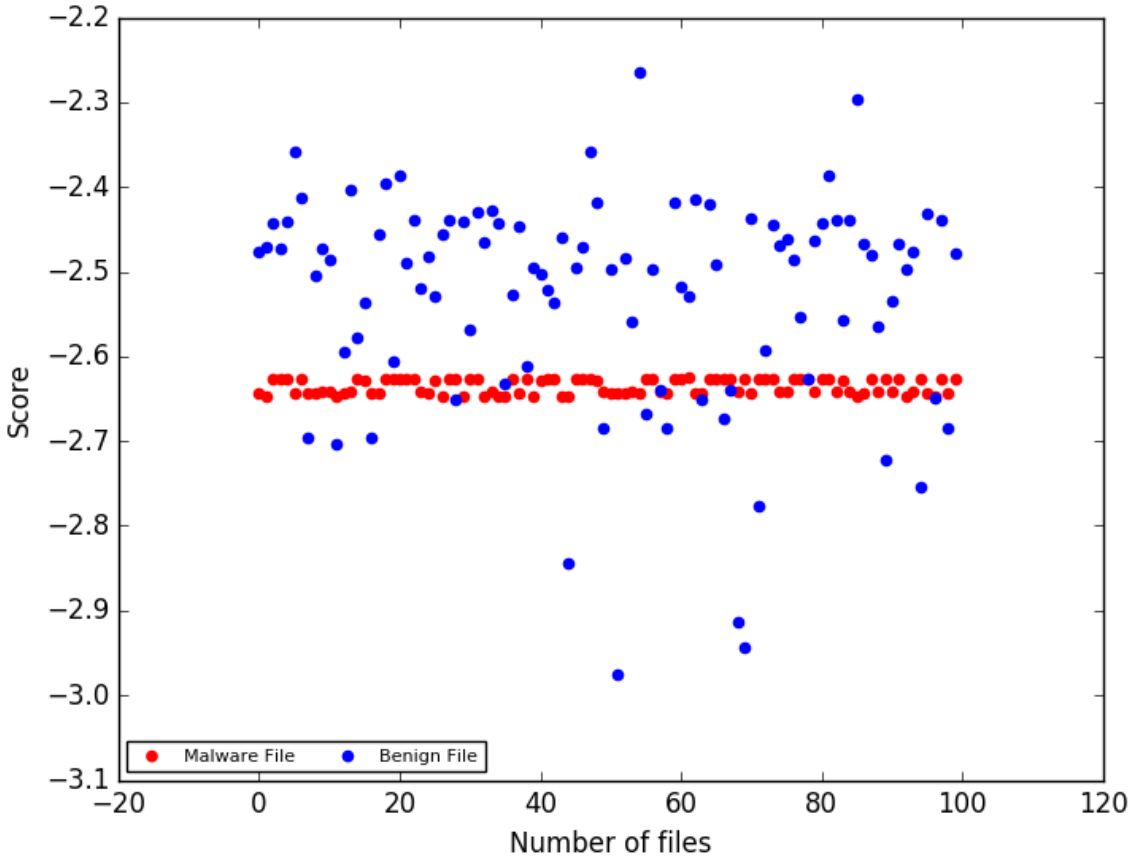

Figure A.17: Winwebsec Scatter Plot for 20% Data Poisoning

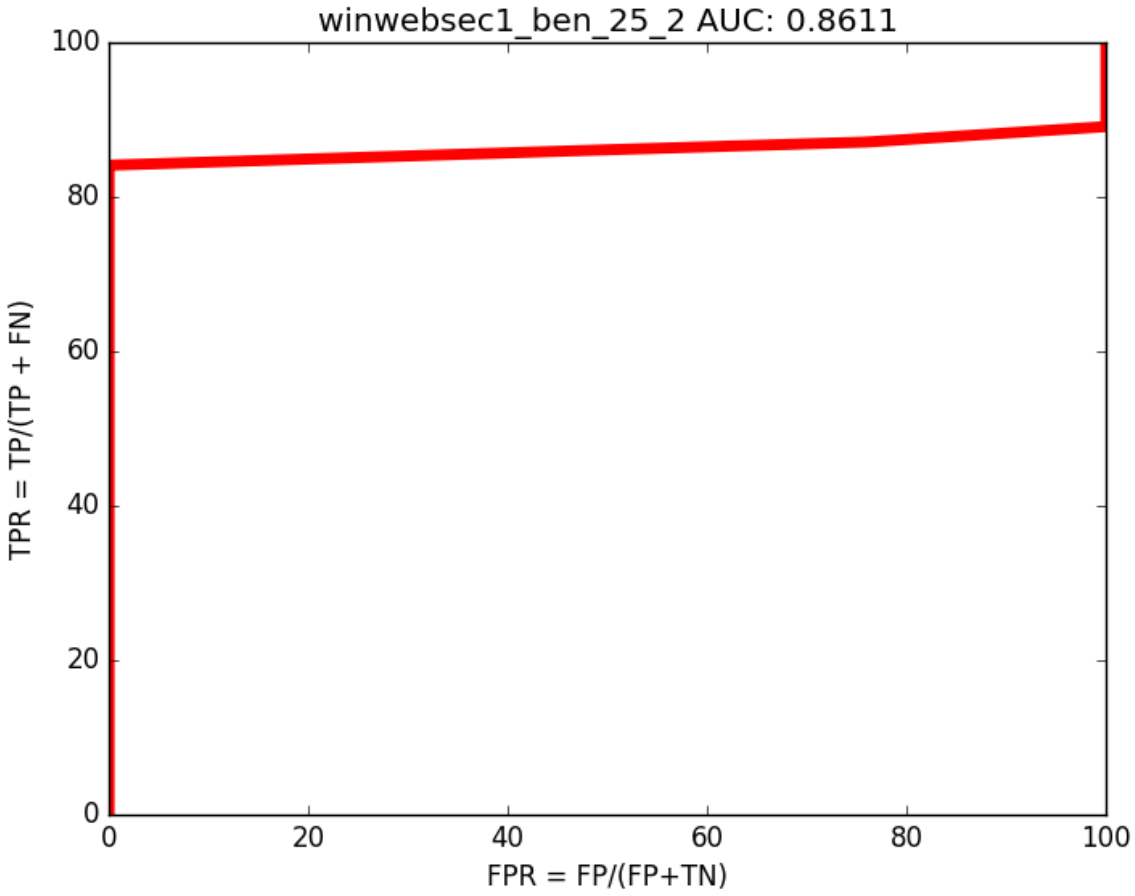

Figure A.18: Winwebsec ROC Curve and AUC with 25% Data Poisoning

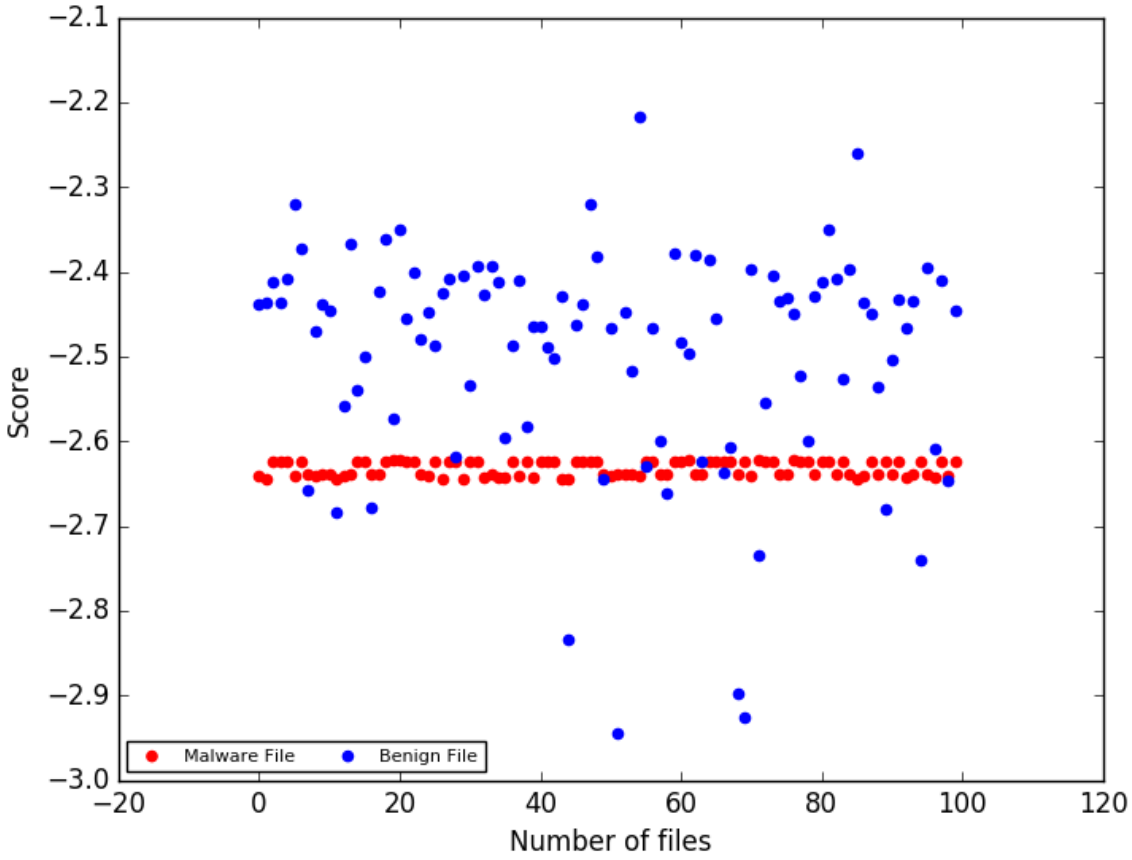

Figure A.19: Winwebsec Scatter Plot for 25% Data Poisoning

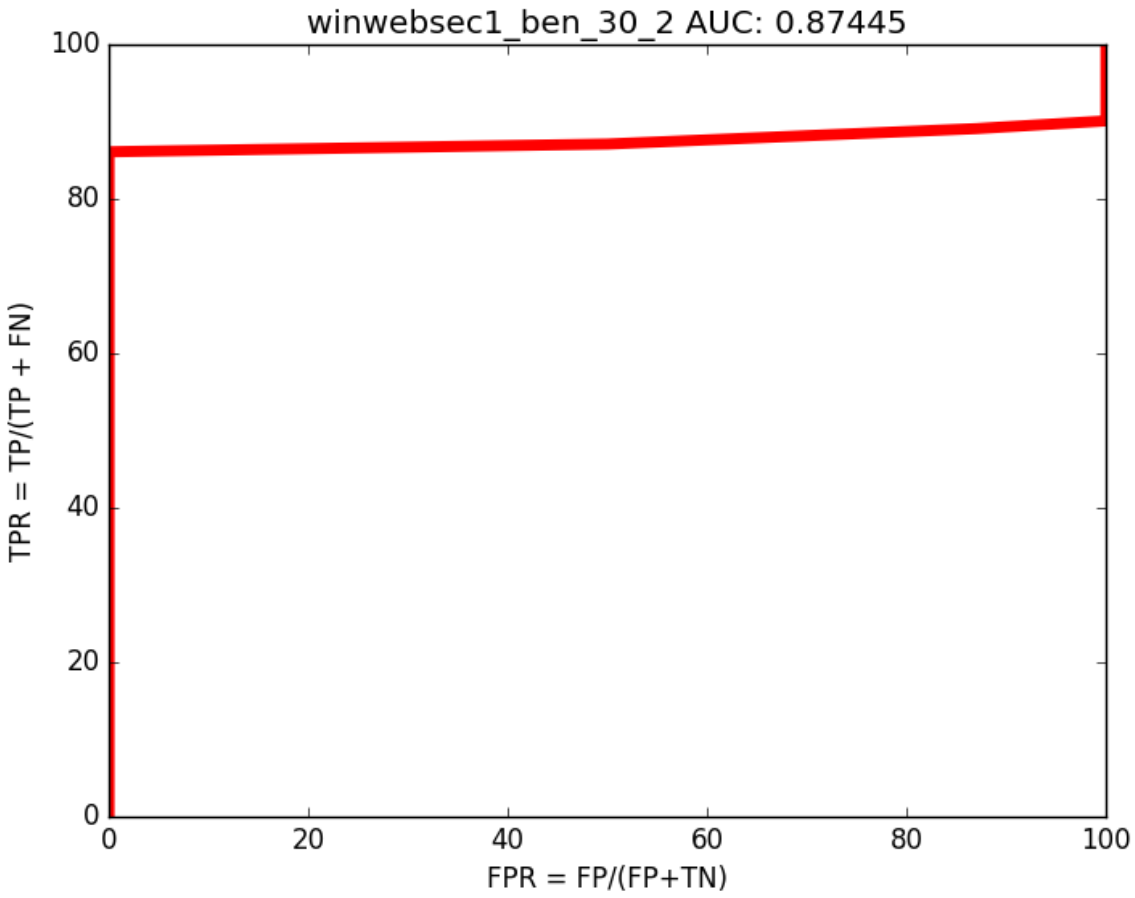

Figure A.20: Winwebsec ROC Curve and AUC with 30% Data Poisoning

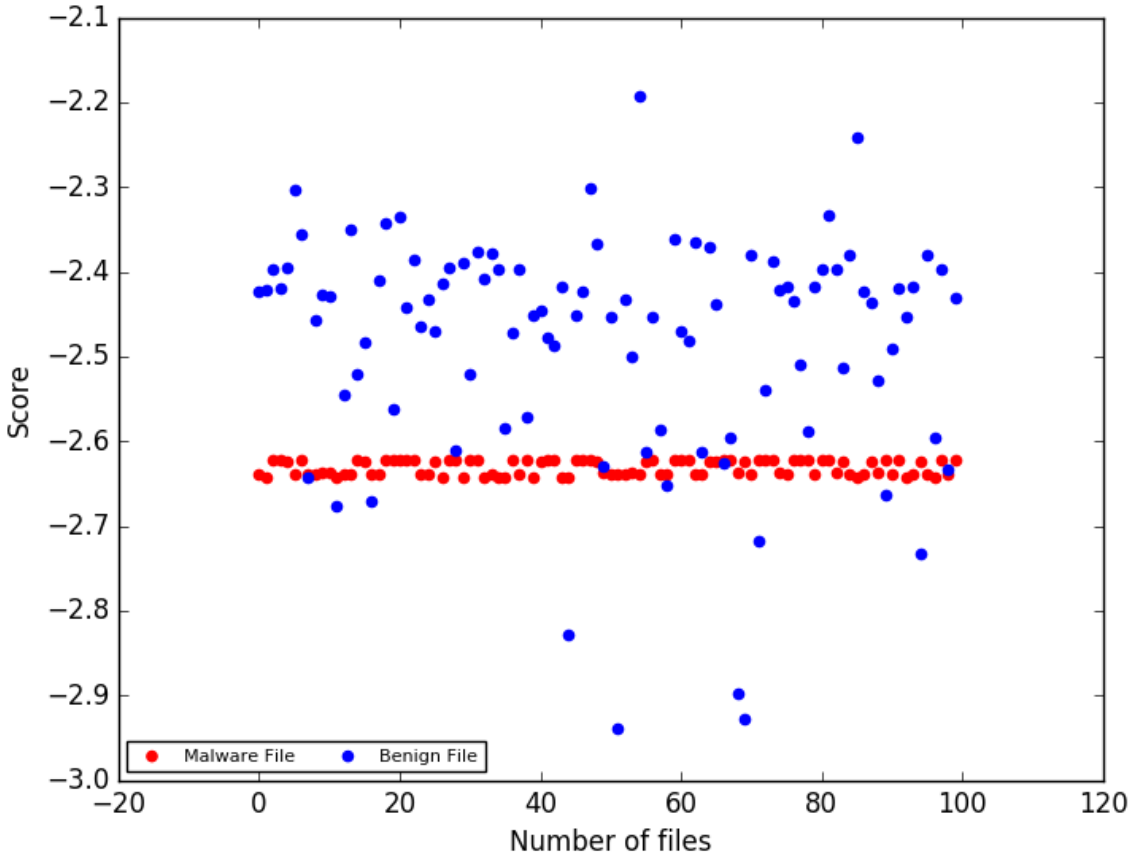

Figure A.21: Winwebsec Scatter Plot for 30% Data Poisoning

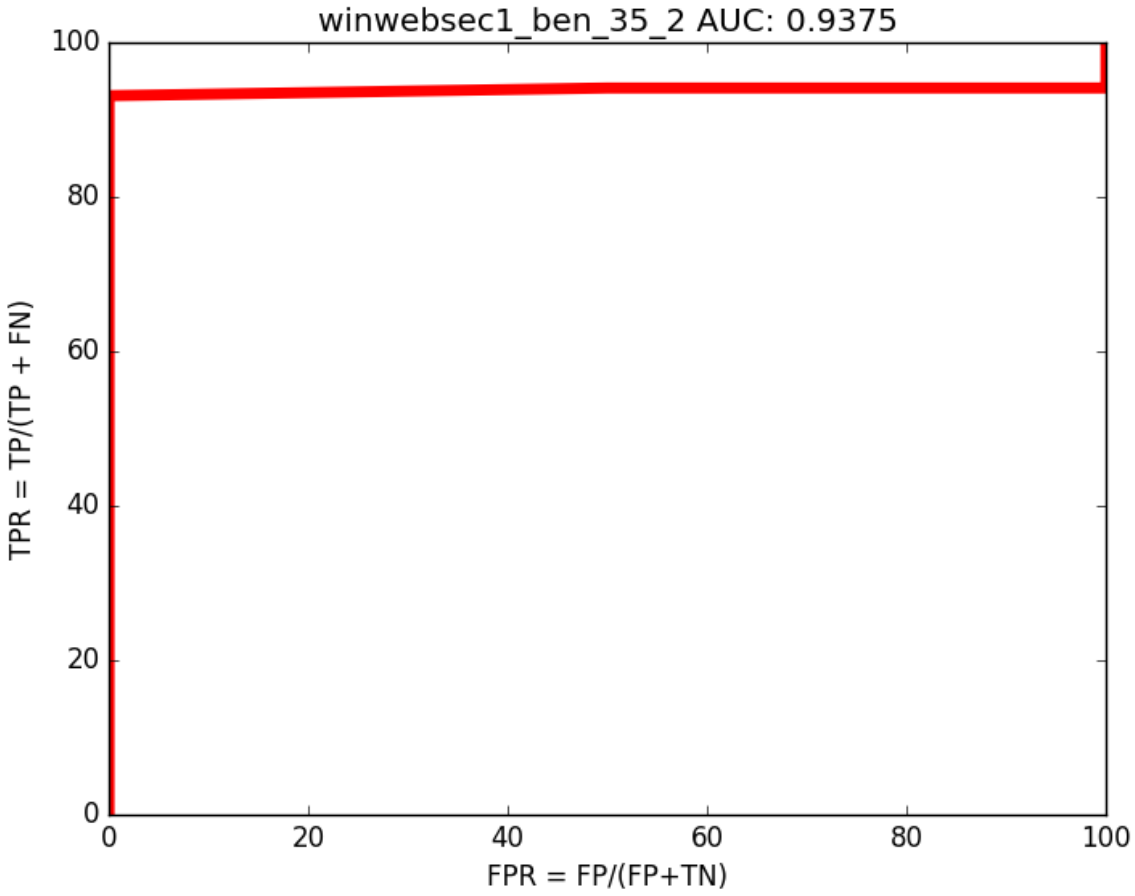

Figure A.22: Winwebsec ROC Curve and AUC with 35% Data Poisoning

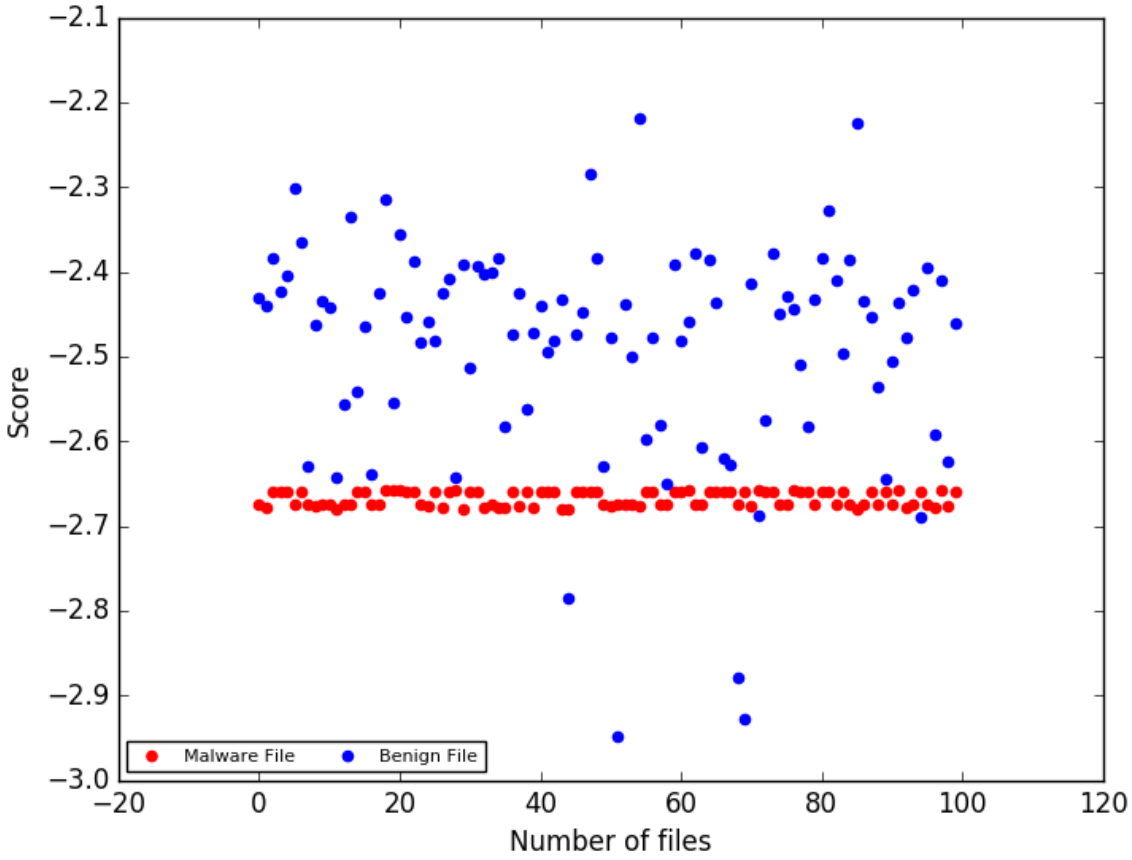

Figure A.23: Winwebsec Scatter Plot for 35% Data Poisoning

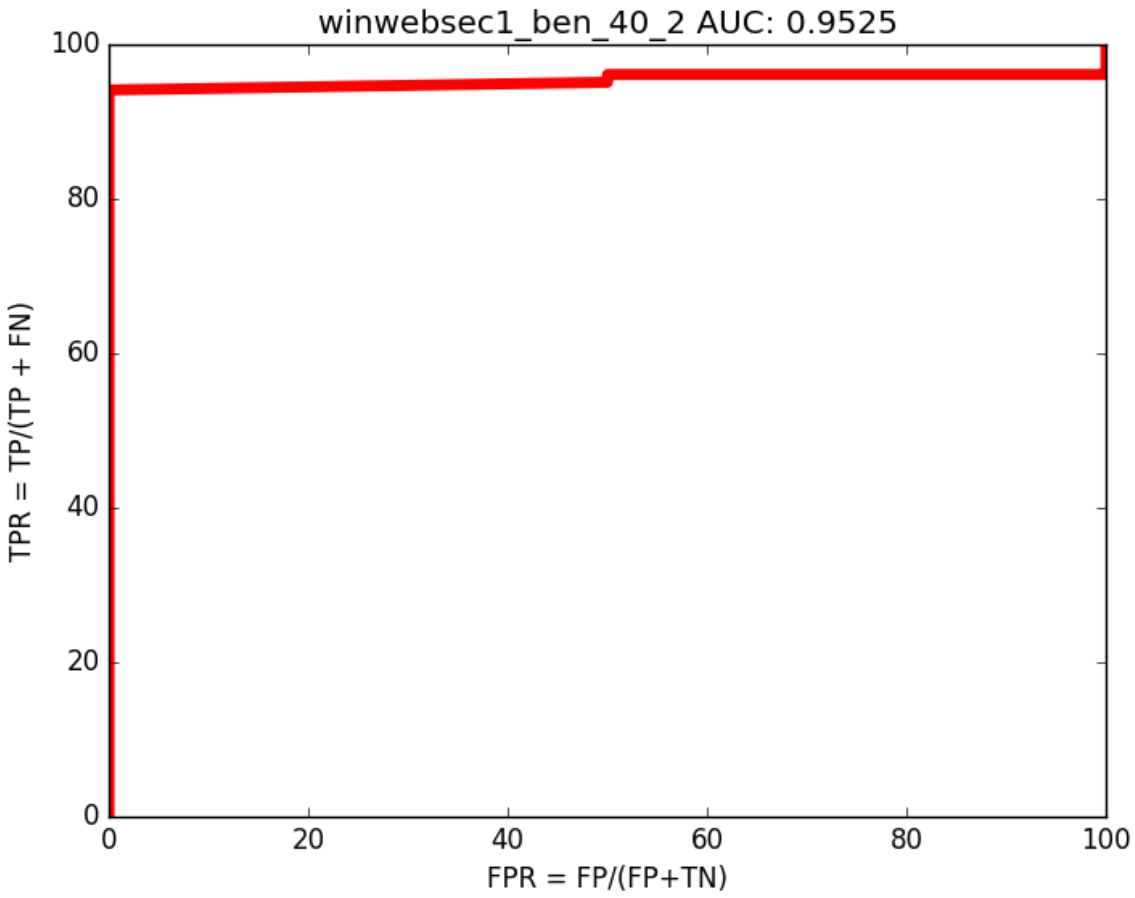

Figure A.24: Winwebsec ROC Curve and AUC with 40% Data Poisoning

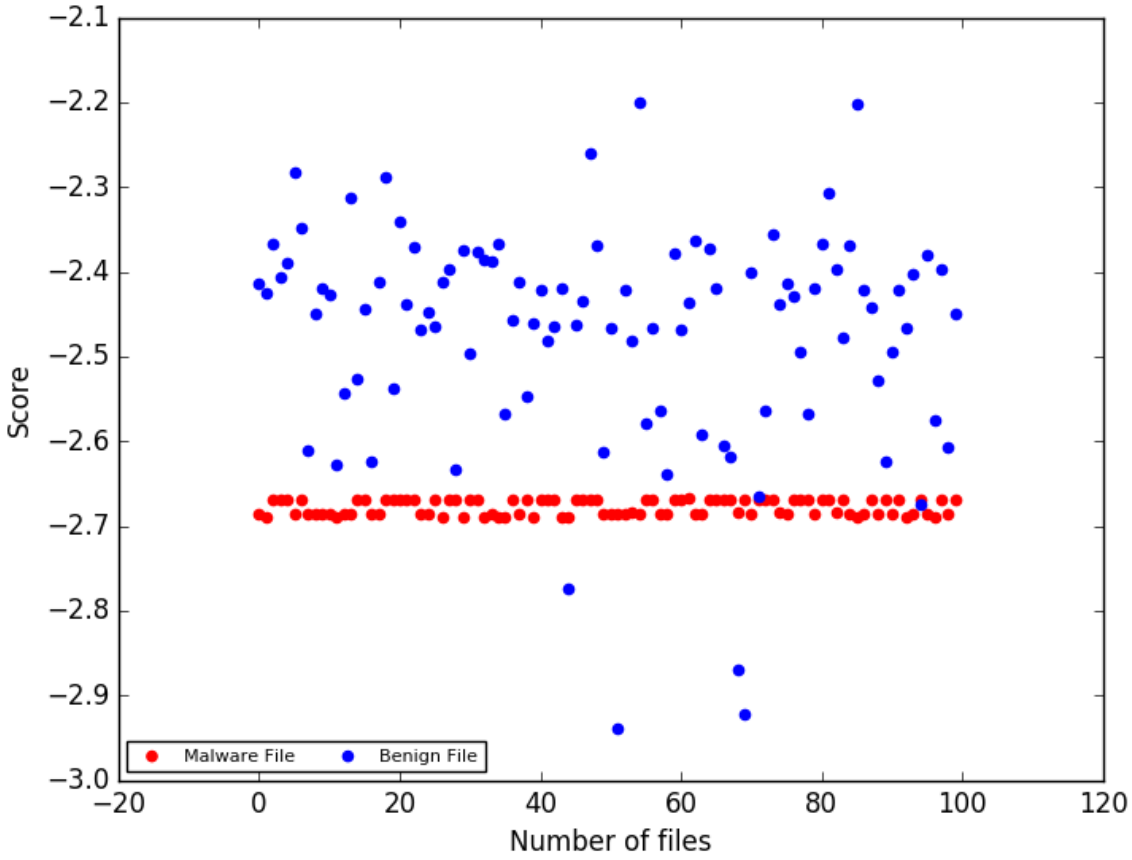

Figure A.25: Winwebsec Scatter Plot for 40% Data Poisoning

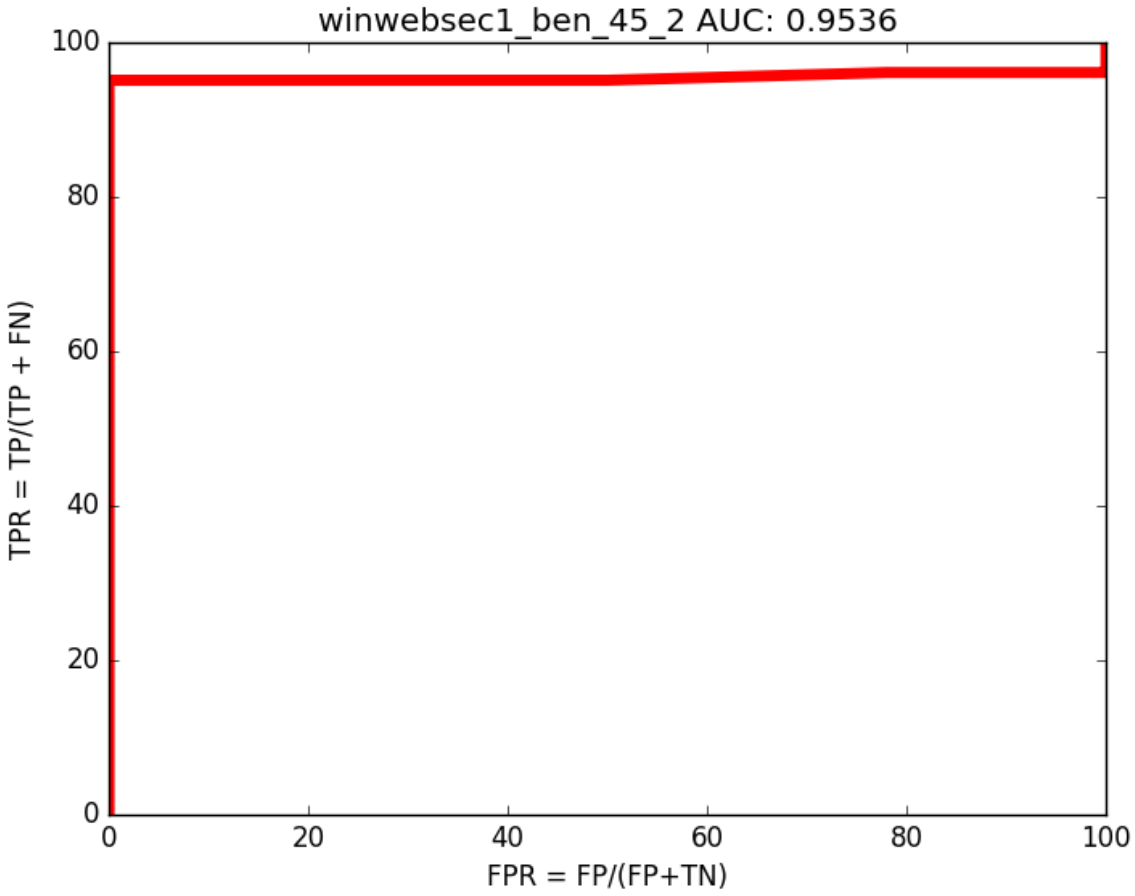

Figure A.26: Winwebsec ROC Curve and AUC with 45% Data Poisoning

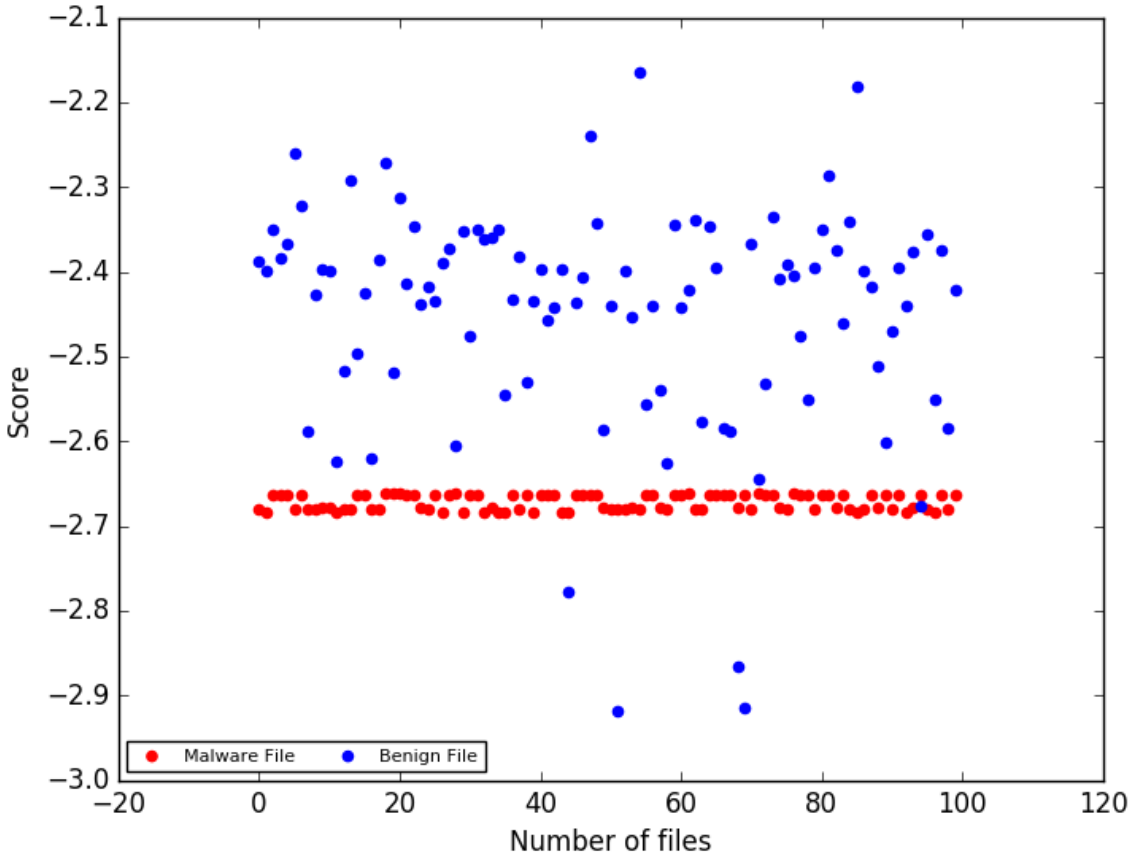

Figure A.27: Winwebsec Scatter Plot for 45% Data Poisoning

A.2 Graphs for Data Poisoning on ZeroAccess

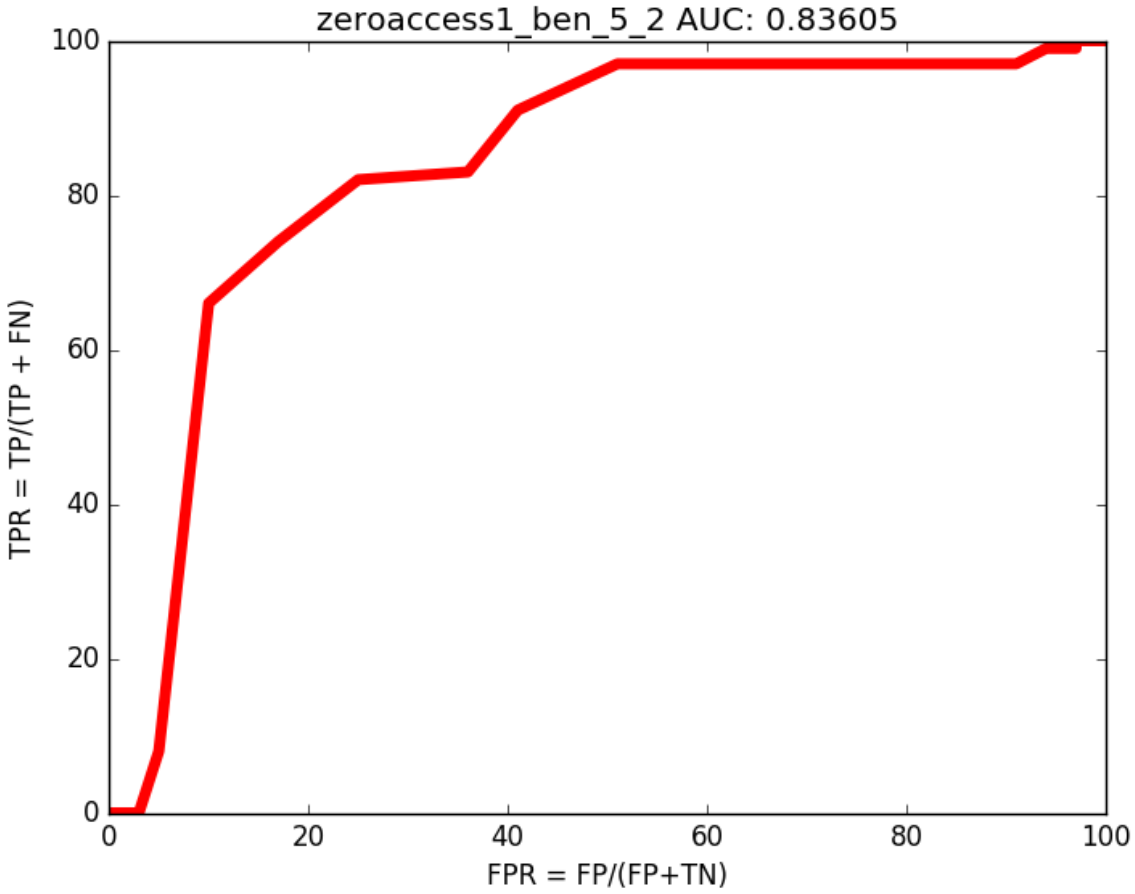

Figure A.28: ZeroAccess ROC Curve and AUC with 5% Data Poisoning

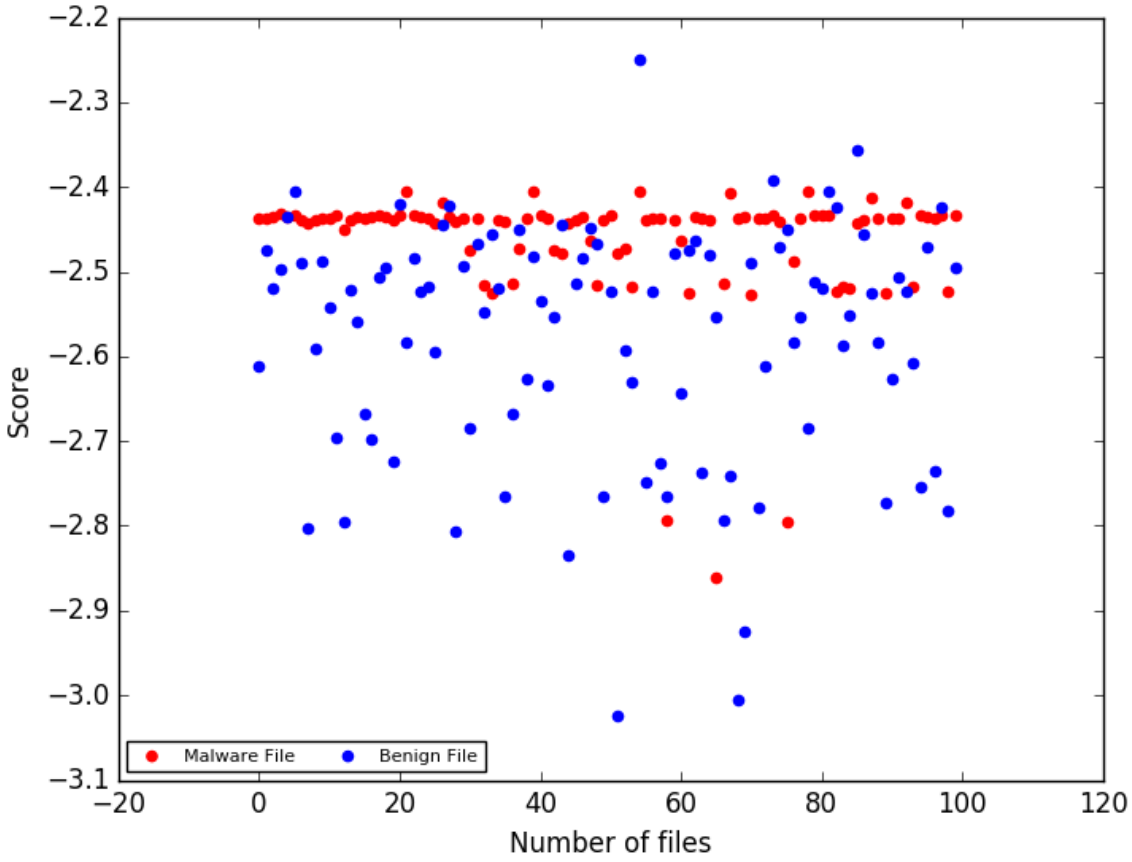

Figure A.29: ZeroAccess Scatter Plot for 5% Data Poisoning

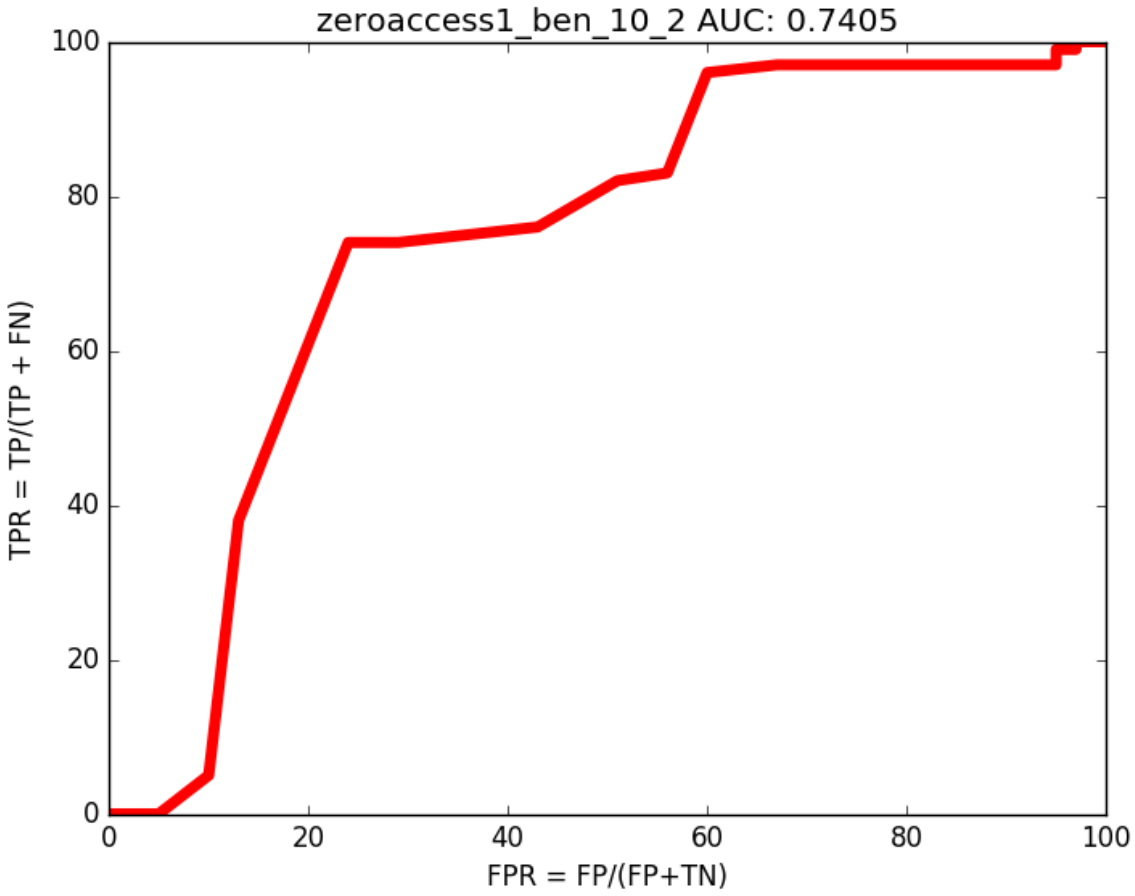

Figure A.30: ZeroAccess ROC Curve and AUC with 10% Data Poisoning

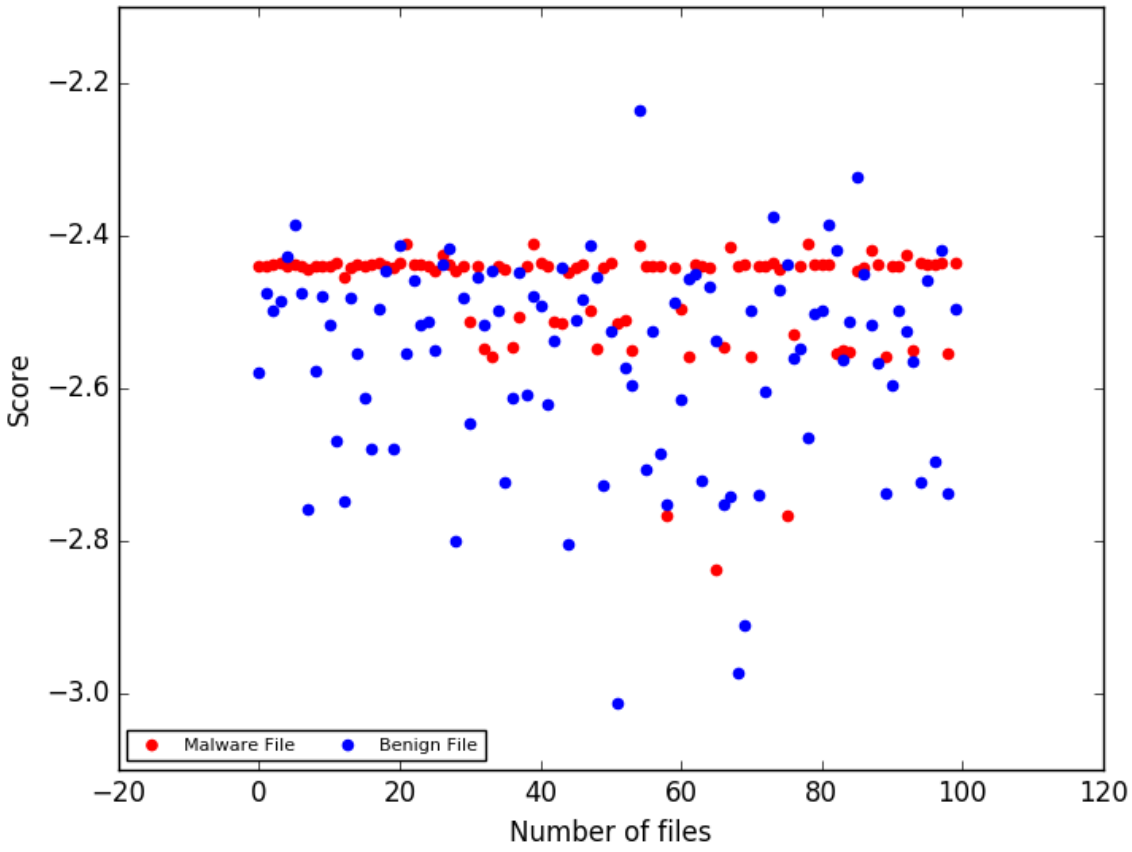

Figure A.31: ZeroAccess Scatter Plot for 10% Data Poisoning

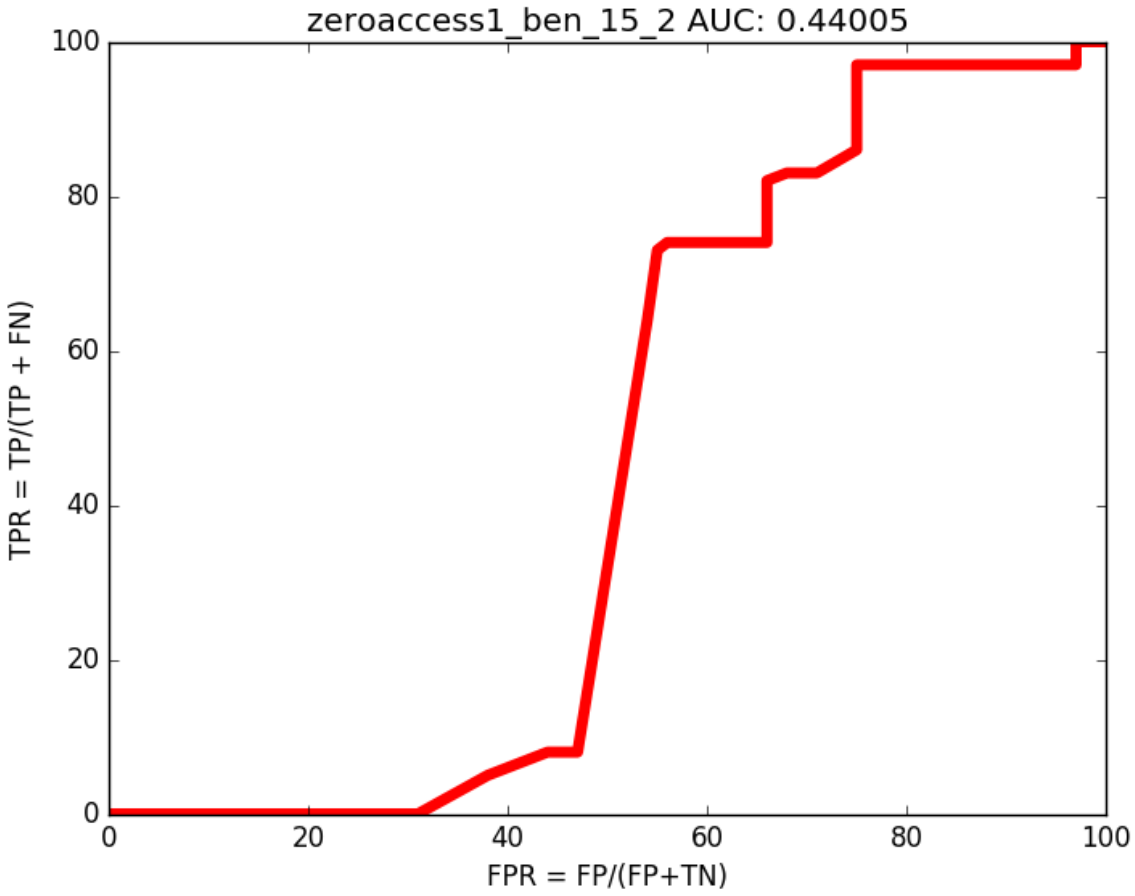

Figure A.32: ZeroAccess ROC Curve and AUC with 15% Data Poisoning

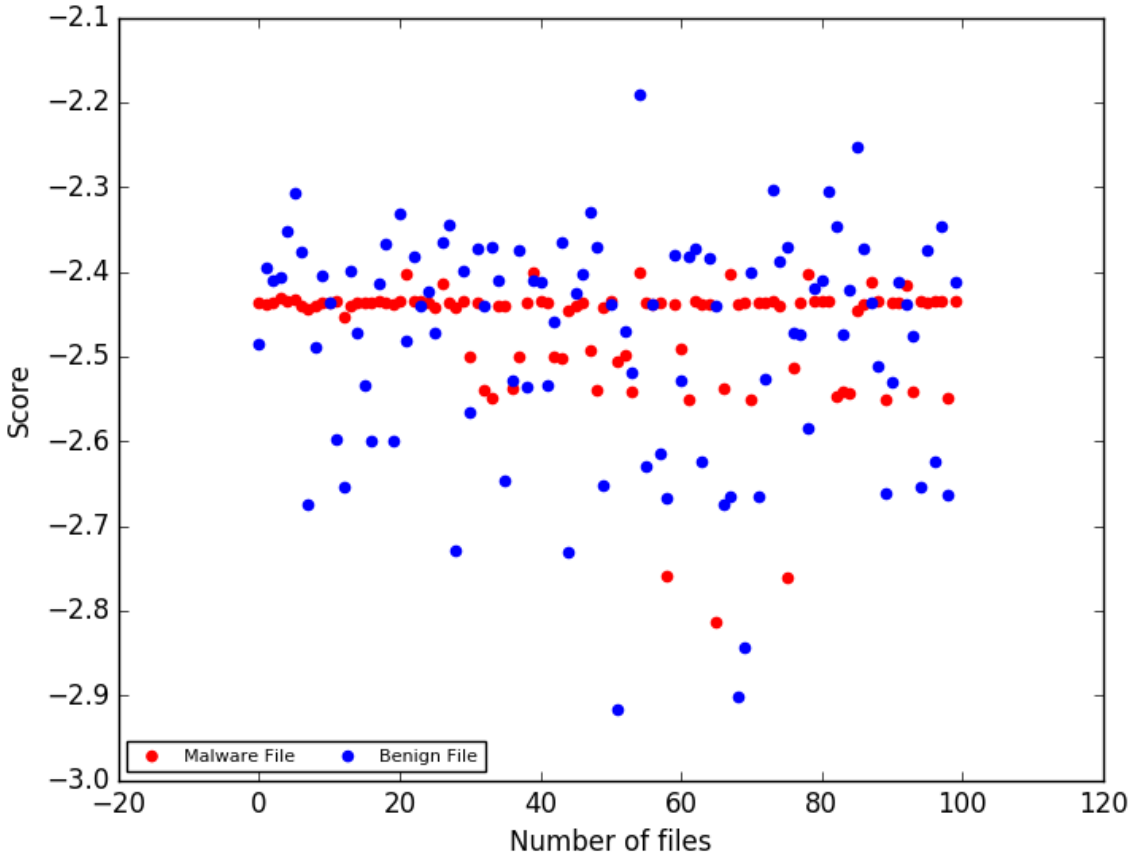

Figure A.33: ZeroAccess Scatter Plot for 15% Data Poisoning

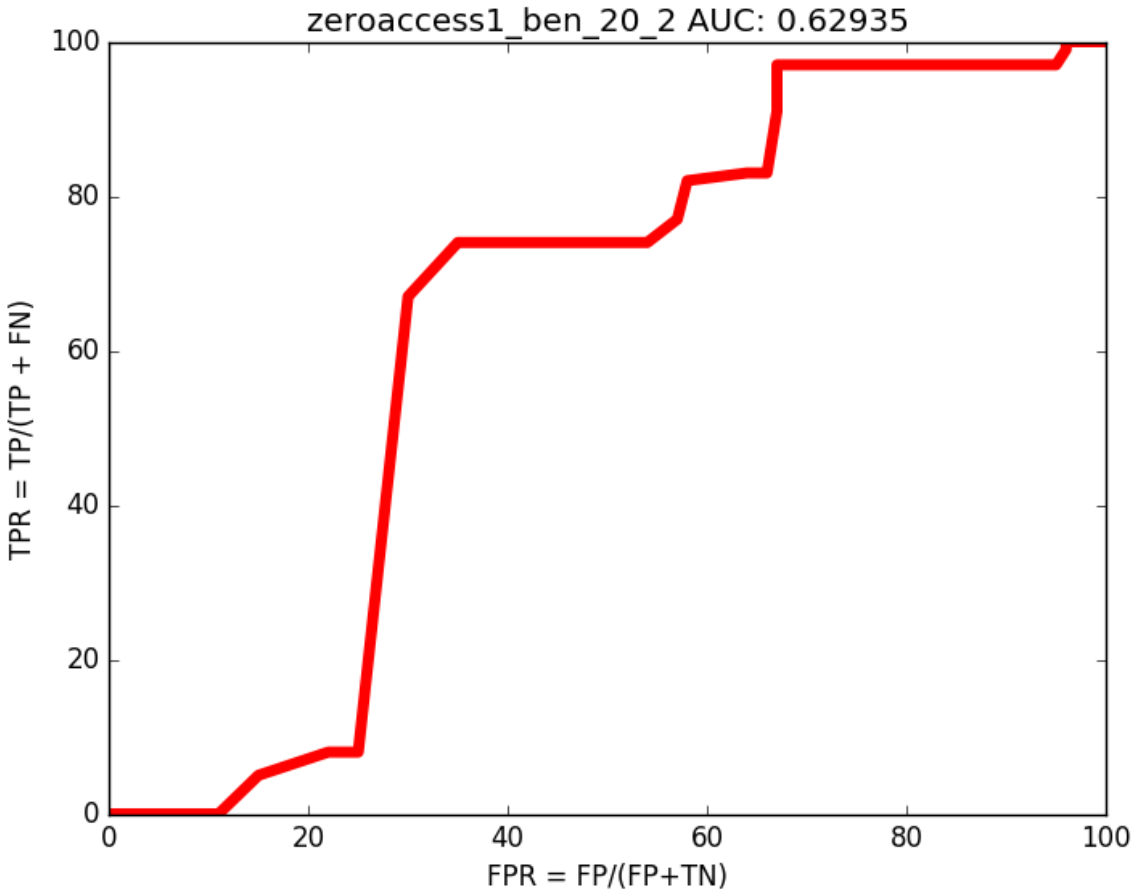

Figure A.34: ZeroAccess ROC Curve and AUC with 20% Data Poisoning

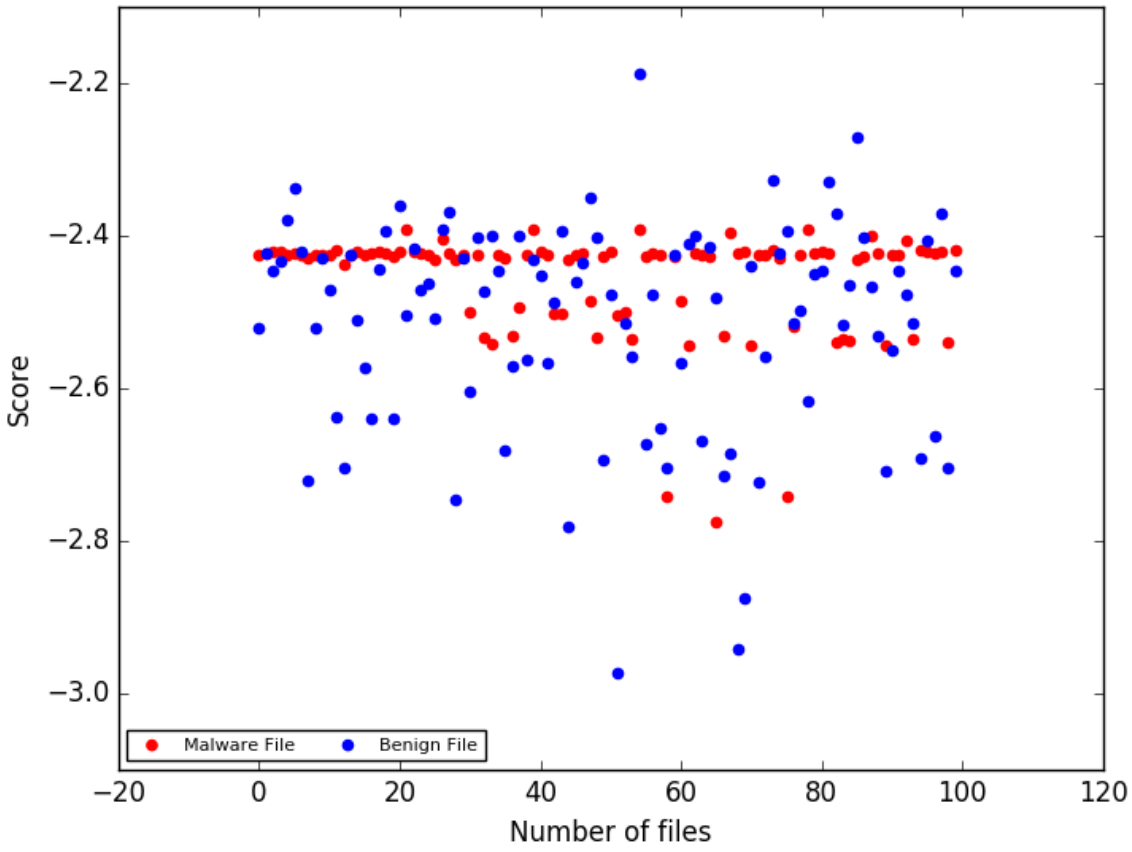

Figure A.35: ZeroAccess Scatter Plot for 20% Data Poisoning

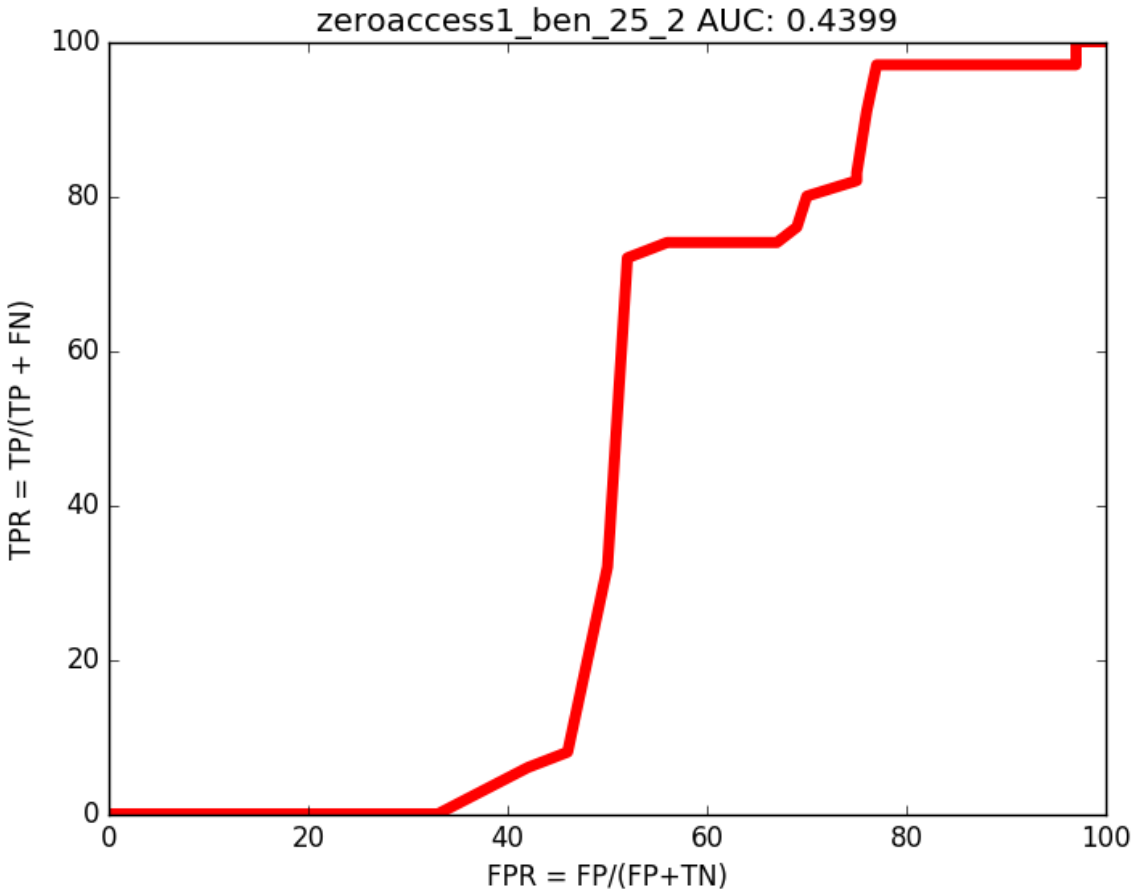

Figure A.36: ZeroAccess ROC Curve and AUC with 25% Data Poisoning

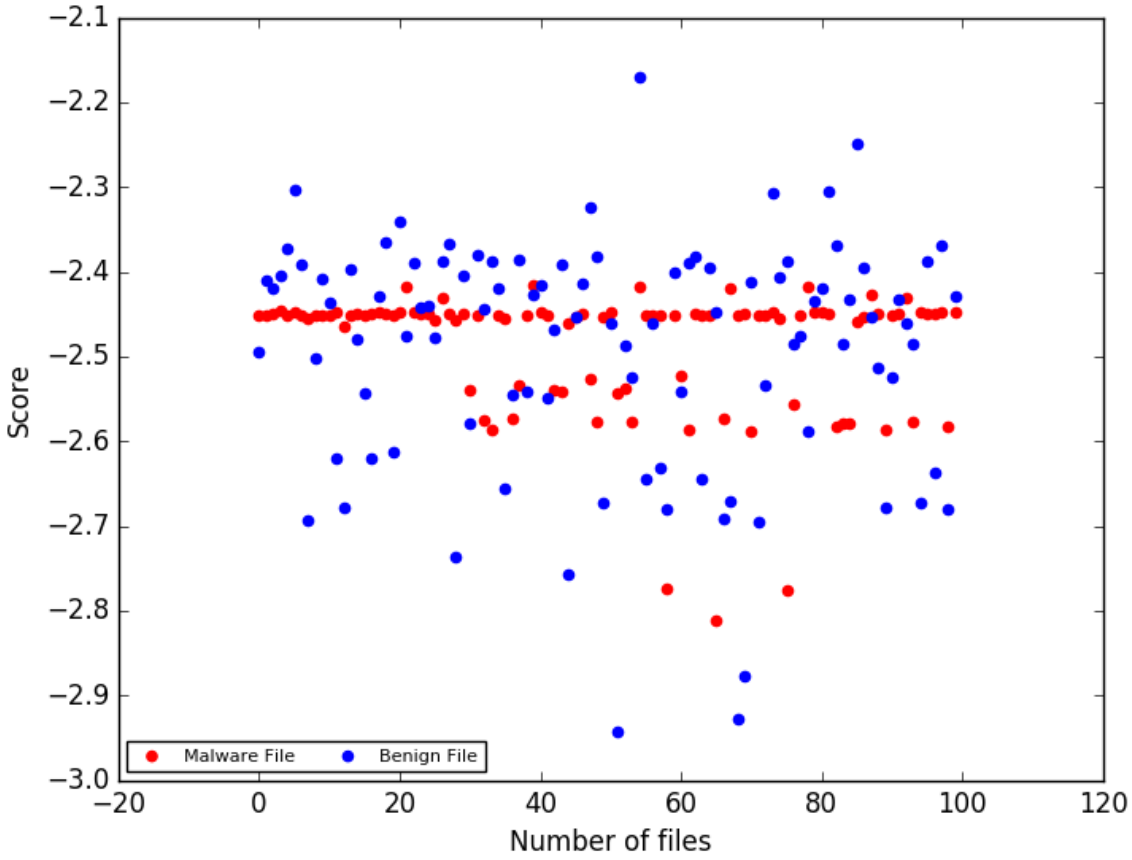

Figure A.37: ZeroAccess Scatter Plot for 25% Data Poisoning

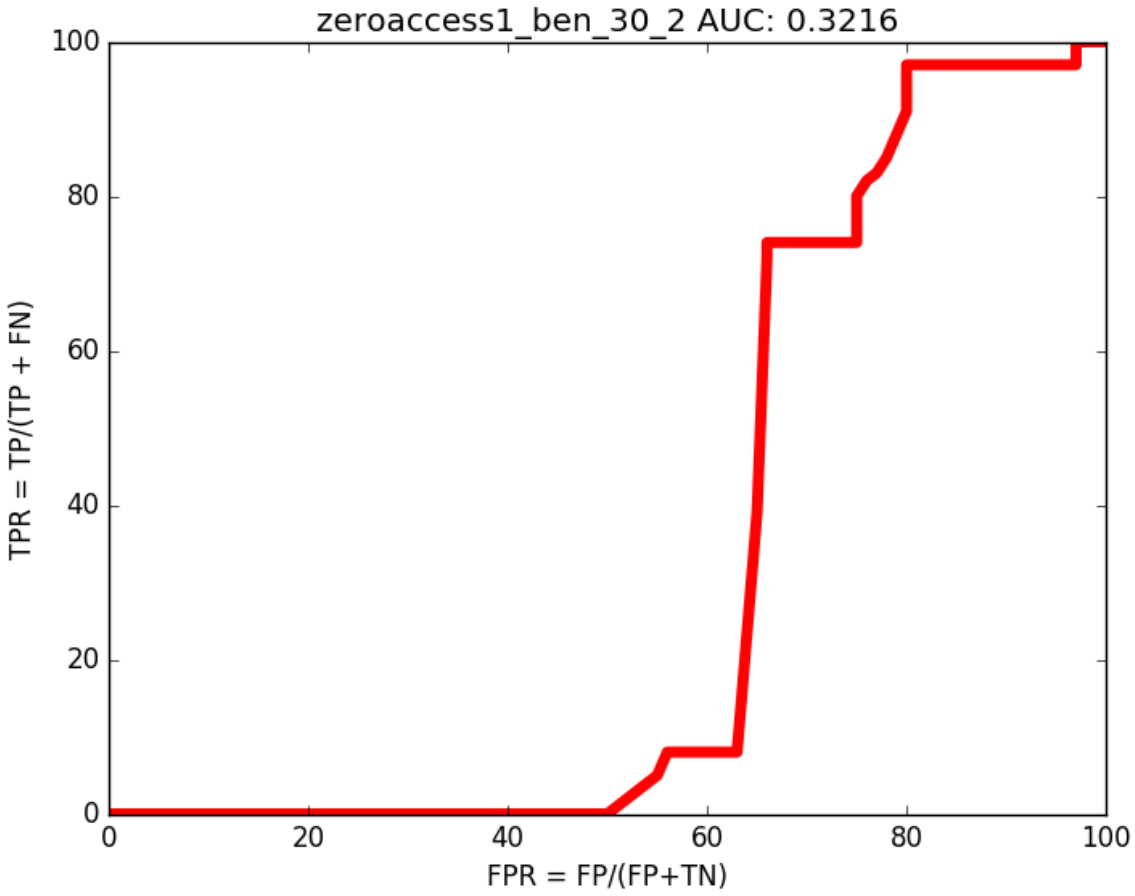

Figure A.38: ZeroAccess ROC Curve and AUC with 30% Data Poisoning

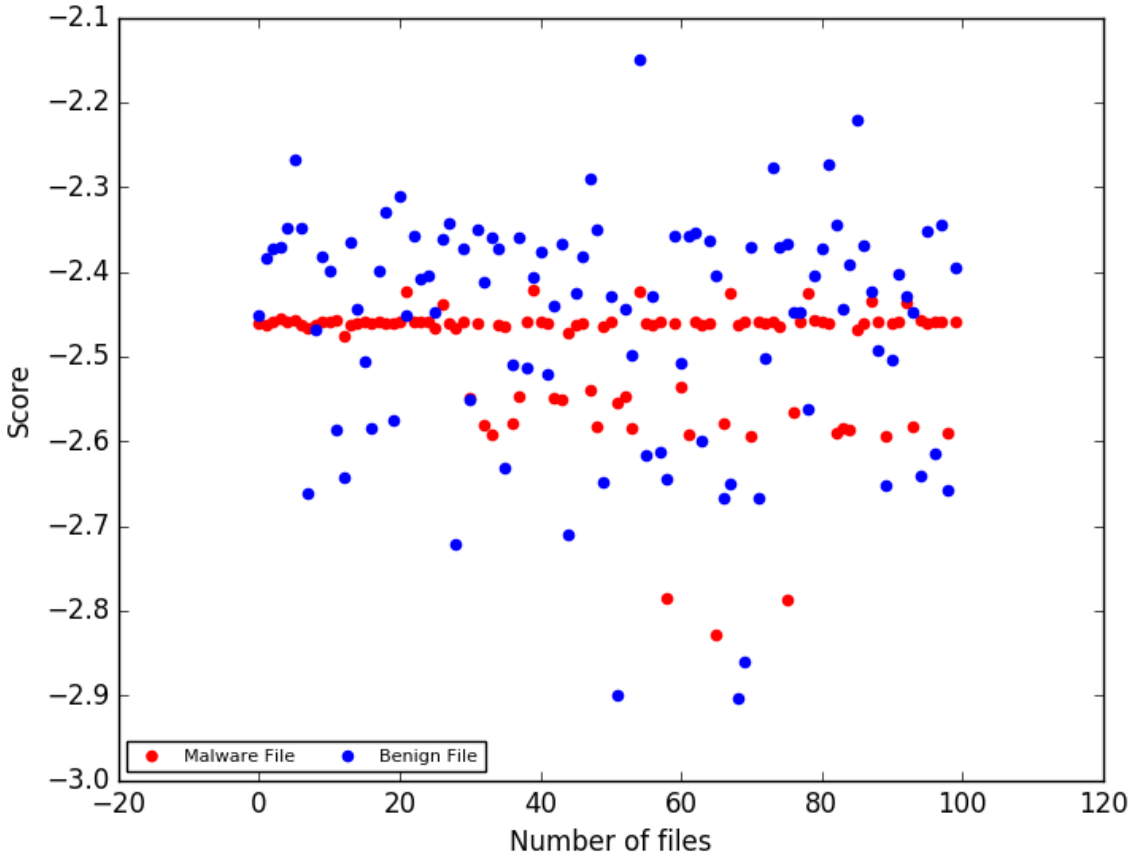

Figure A.39: ZeroAccess Scatter Plot for 30% Data Poisoning

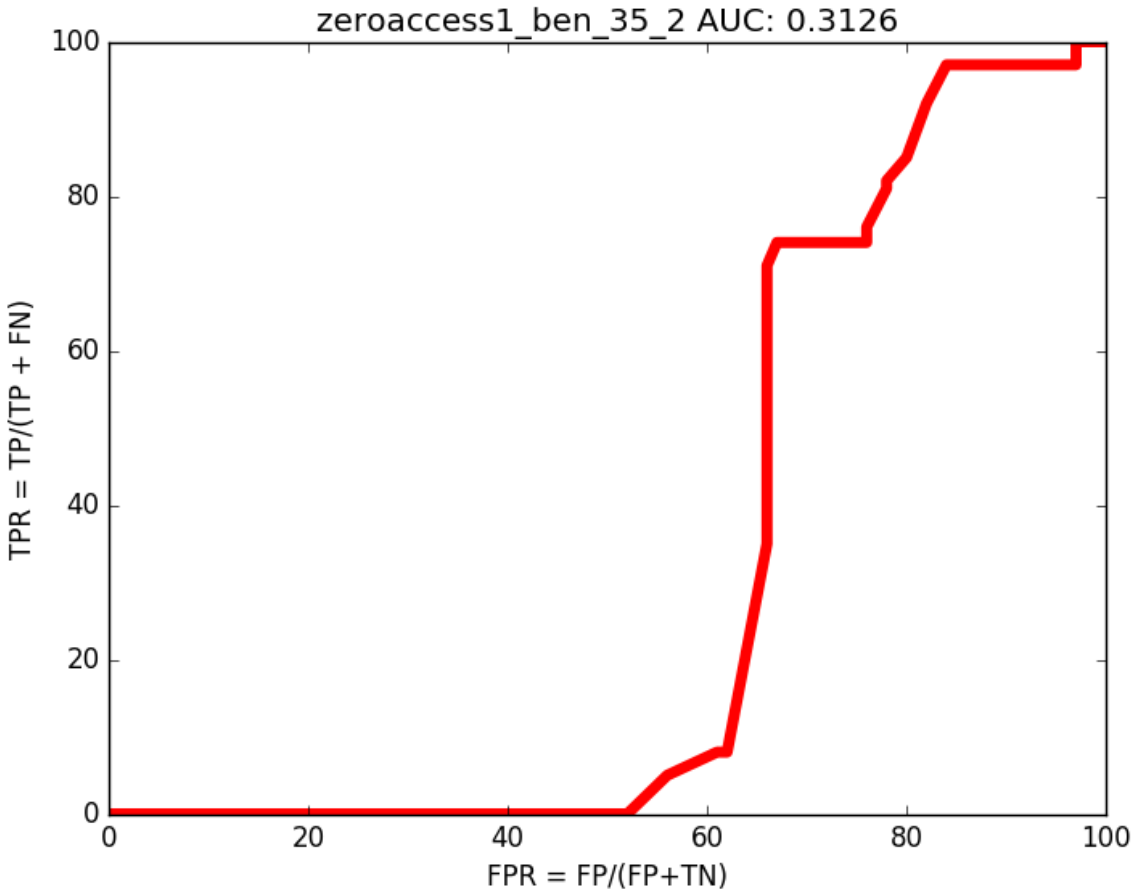

Figure A.40: ZeroAccess ROC Curve and AUC with 35% Data Poisoning

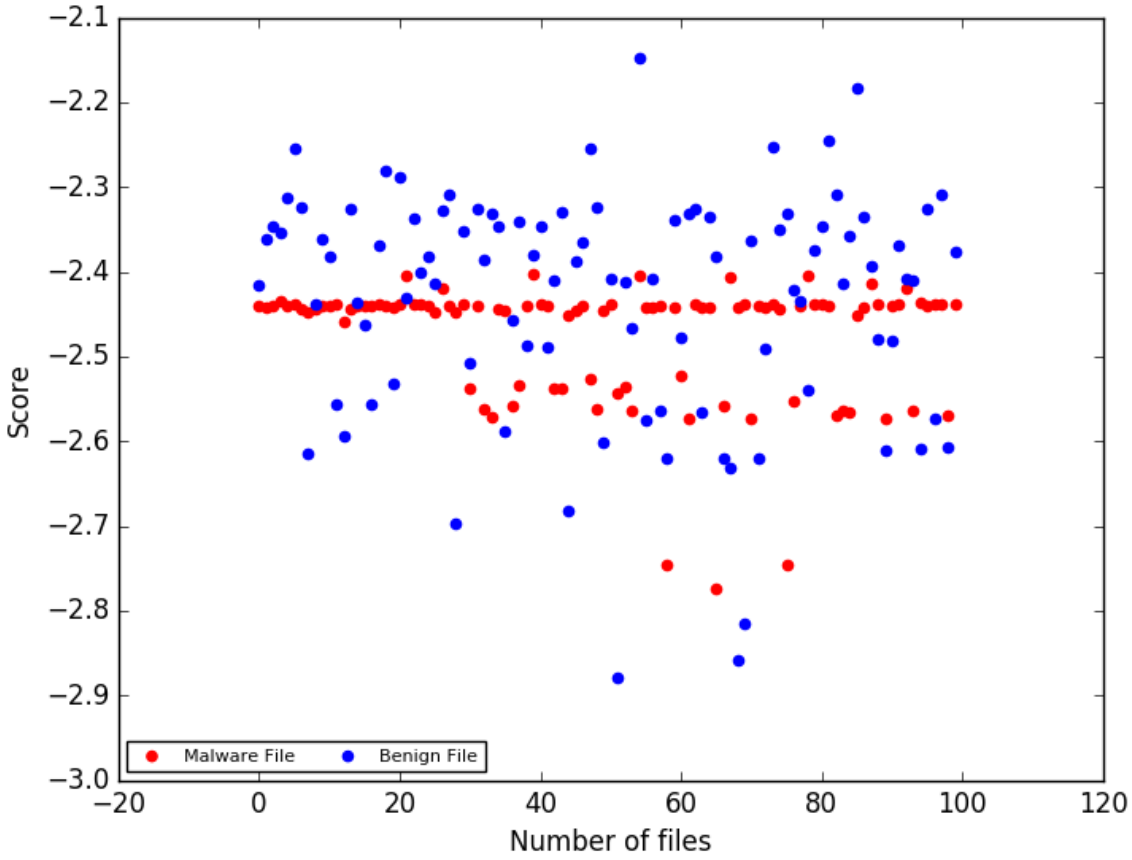

Figure A.41: ZeroAccess Scatter Plot for 35% Data Poisoning

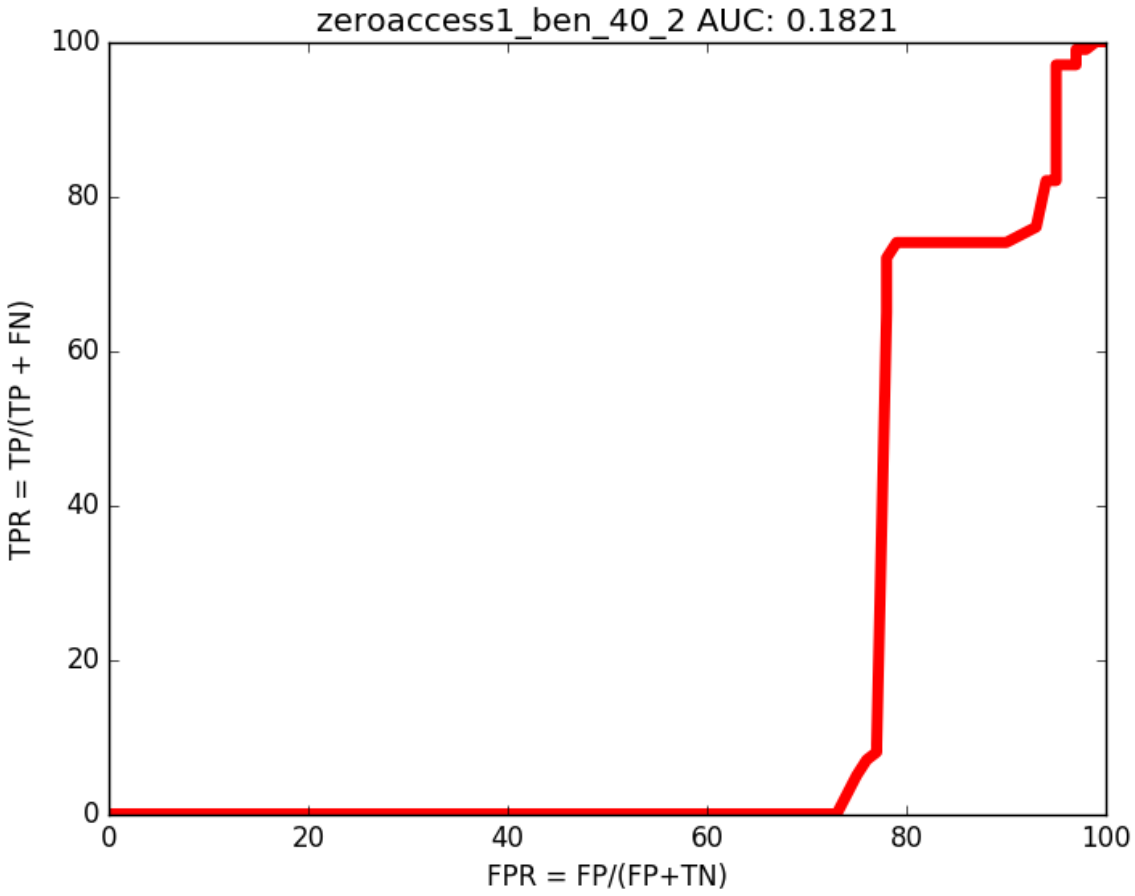

Figure A.42: ZeroAccess ROC Curve and AUC with 40% Data Poisoning

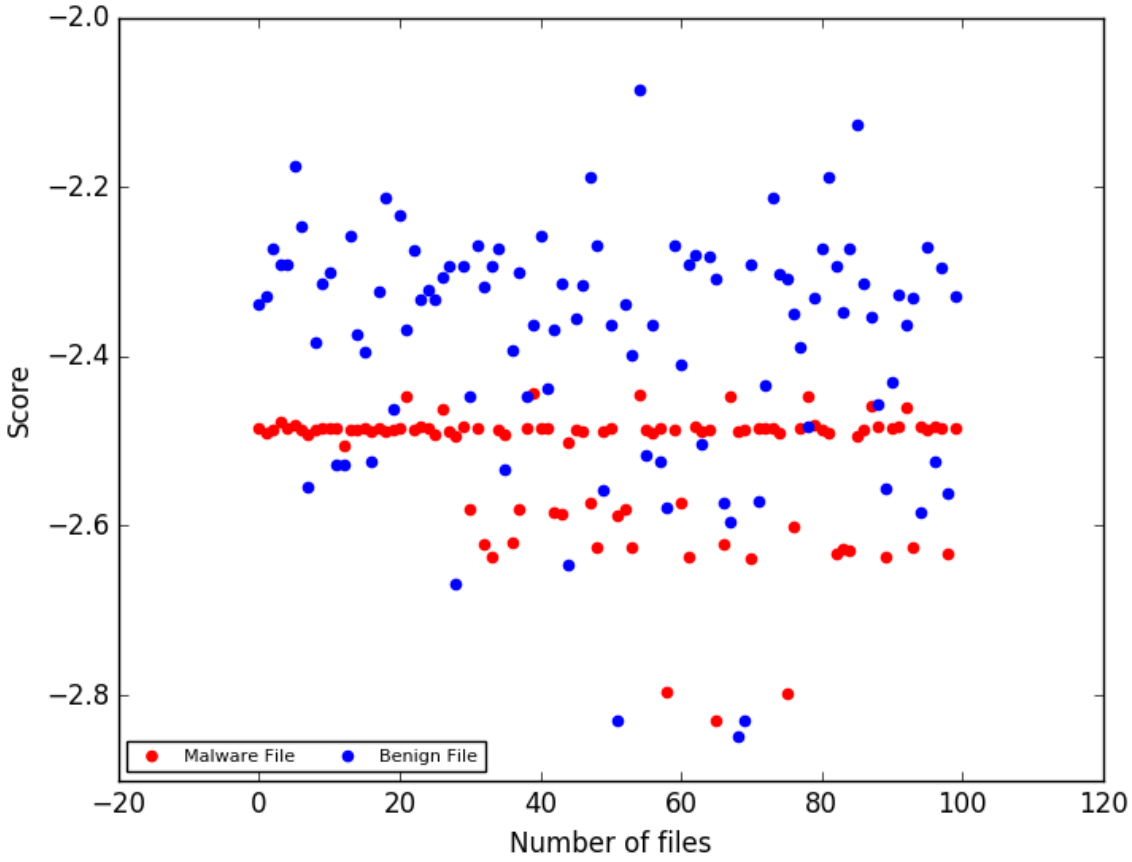

Figure A.43: ZeroAccess Scatter Plot for 40% Data Poisoning

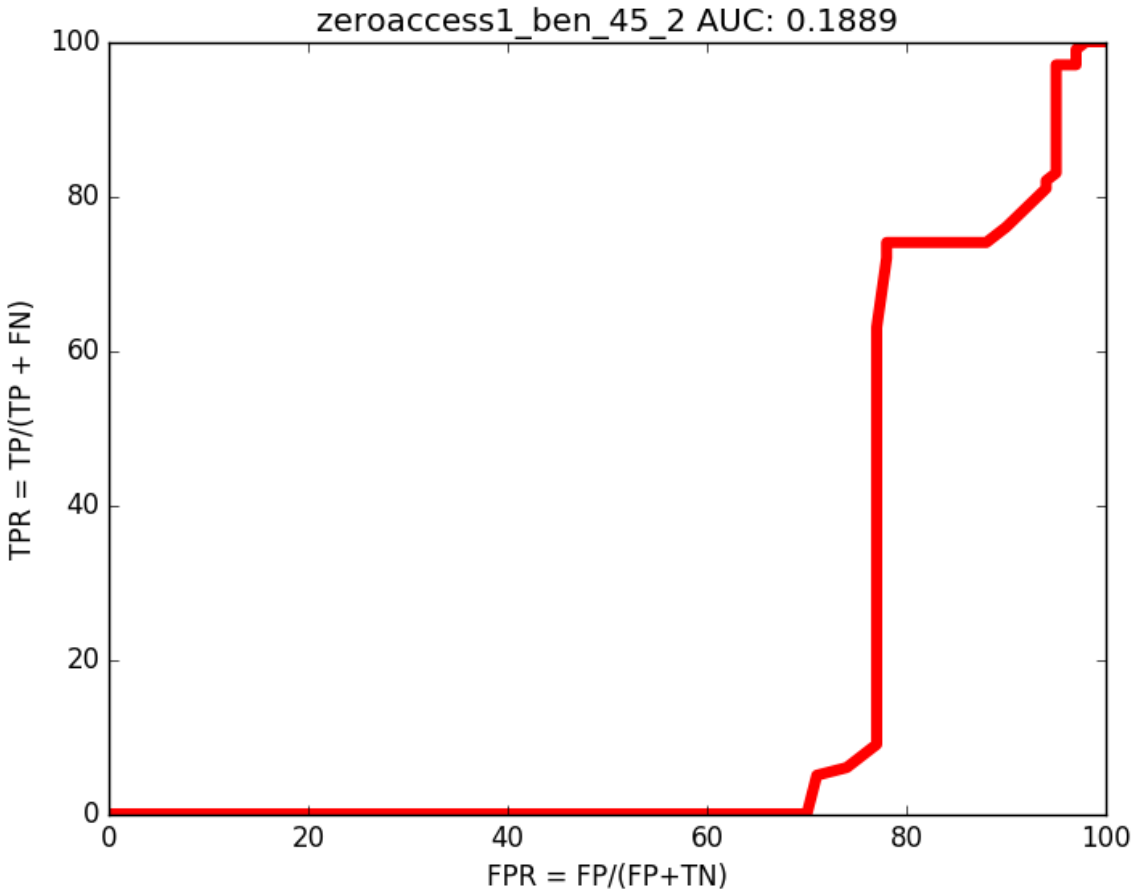

Figure A.44: ZeroAccess ROC Curve and AUC with 45% Data Poisoning

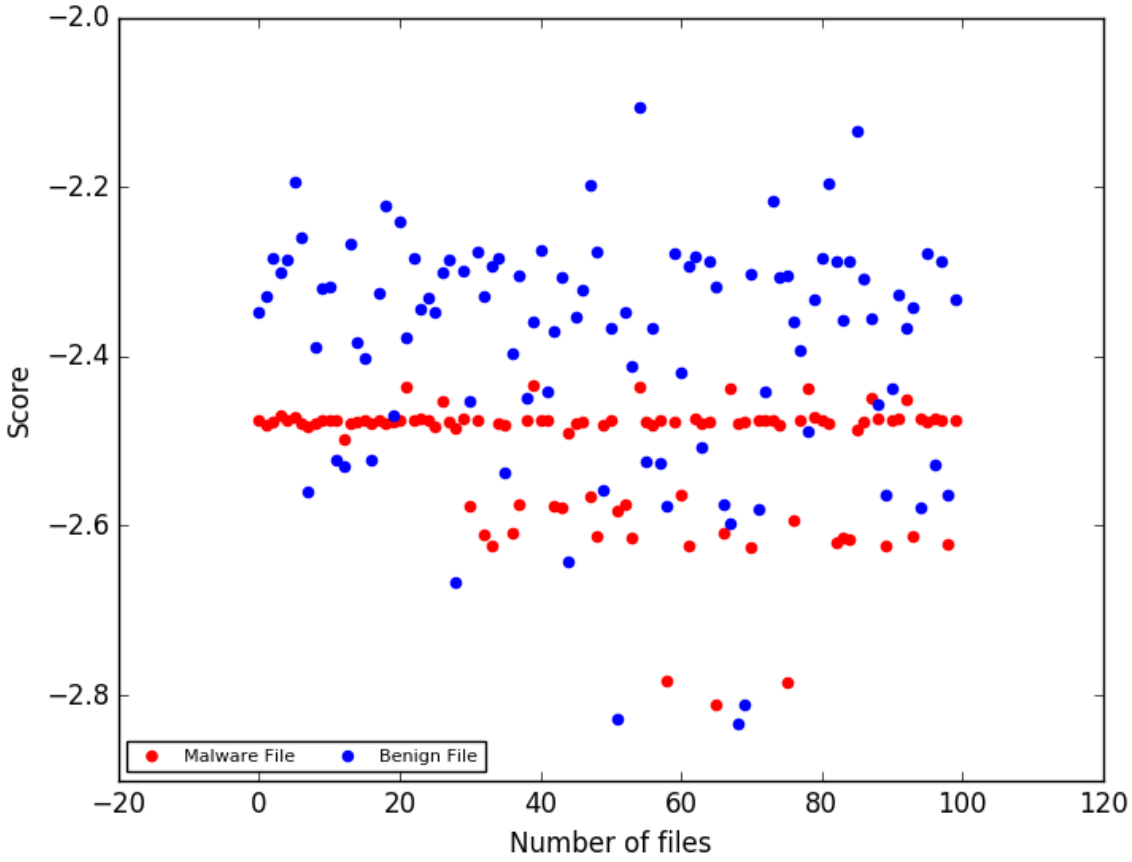

Figure A.45: ZeroAccess Scatter Plot for 45% Data Poisoning

A.3 Graphs for Data Poisoning on Zbot

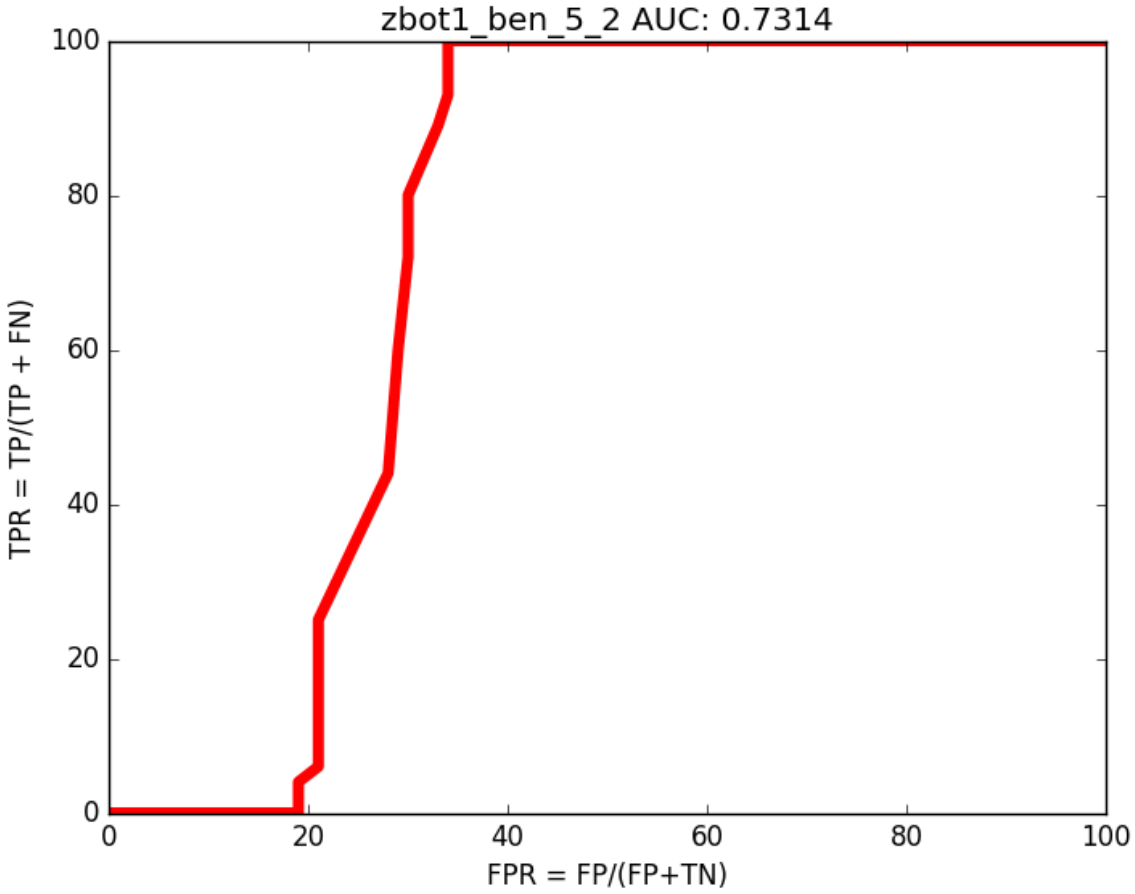

Figure A.46: Zbot ROC Curve and AUC with 5% Data Poisoning

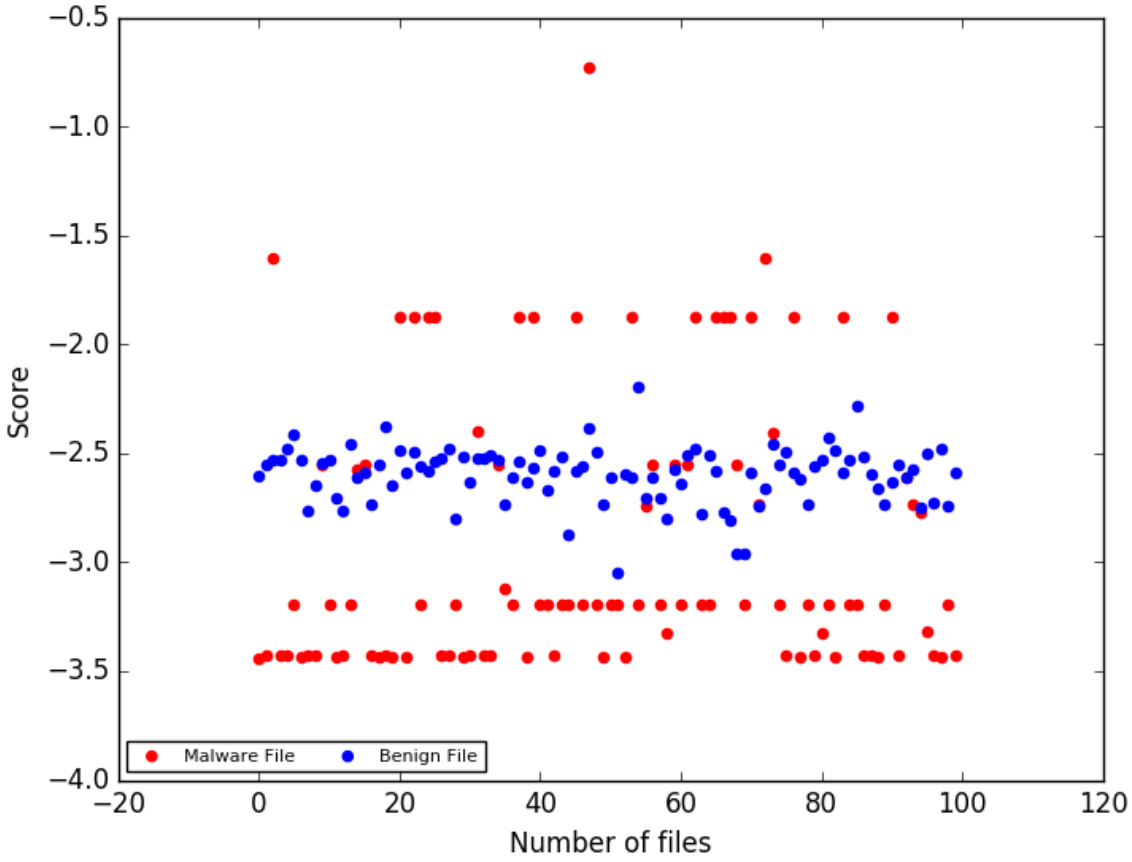

Figure A.47: Zbot Scatter Plot for 5% Data Poisoning

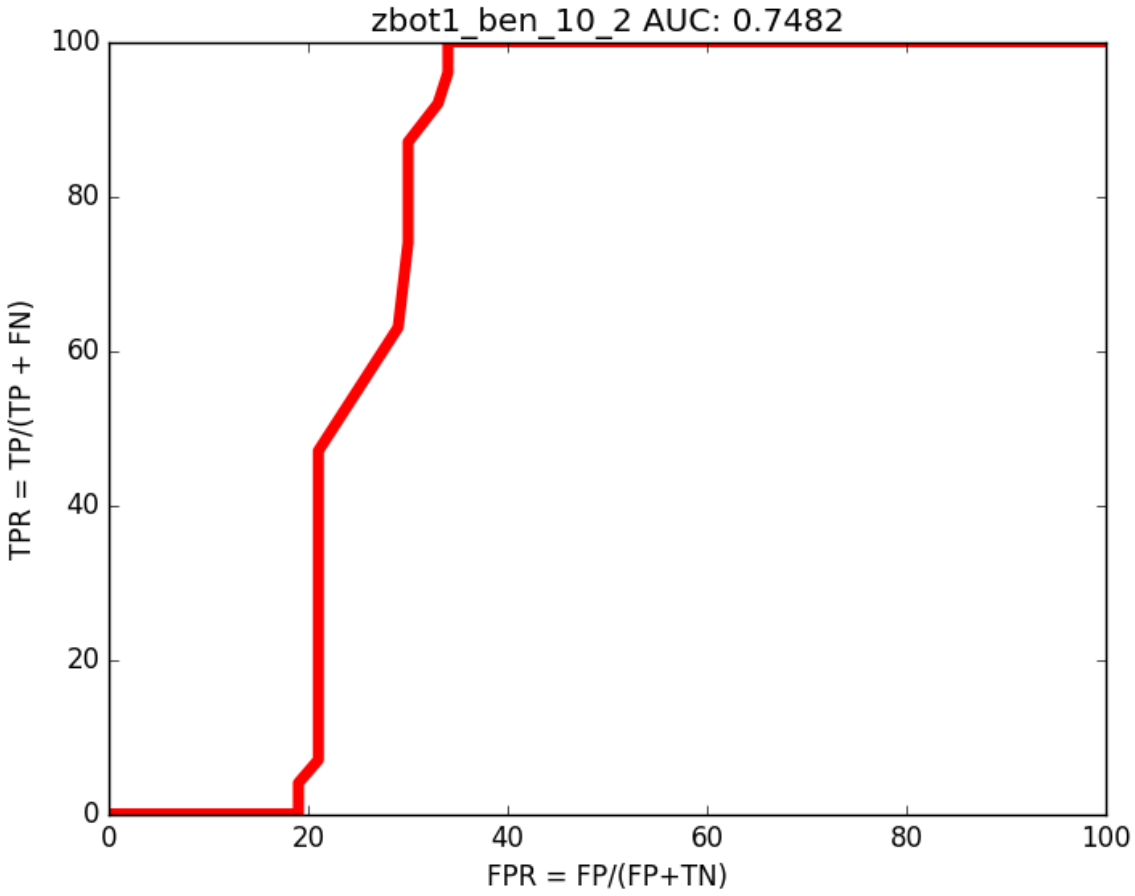

Figure A.48: Zbot ROC Curve and AUC with 10% Data Poisoning

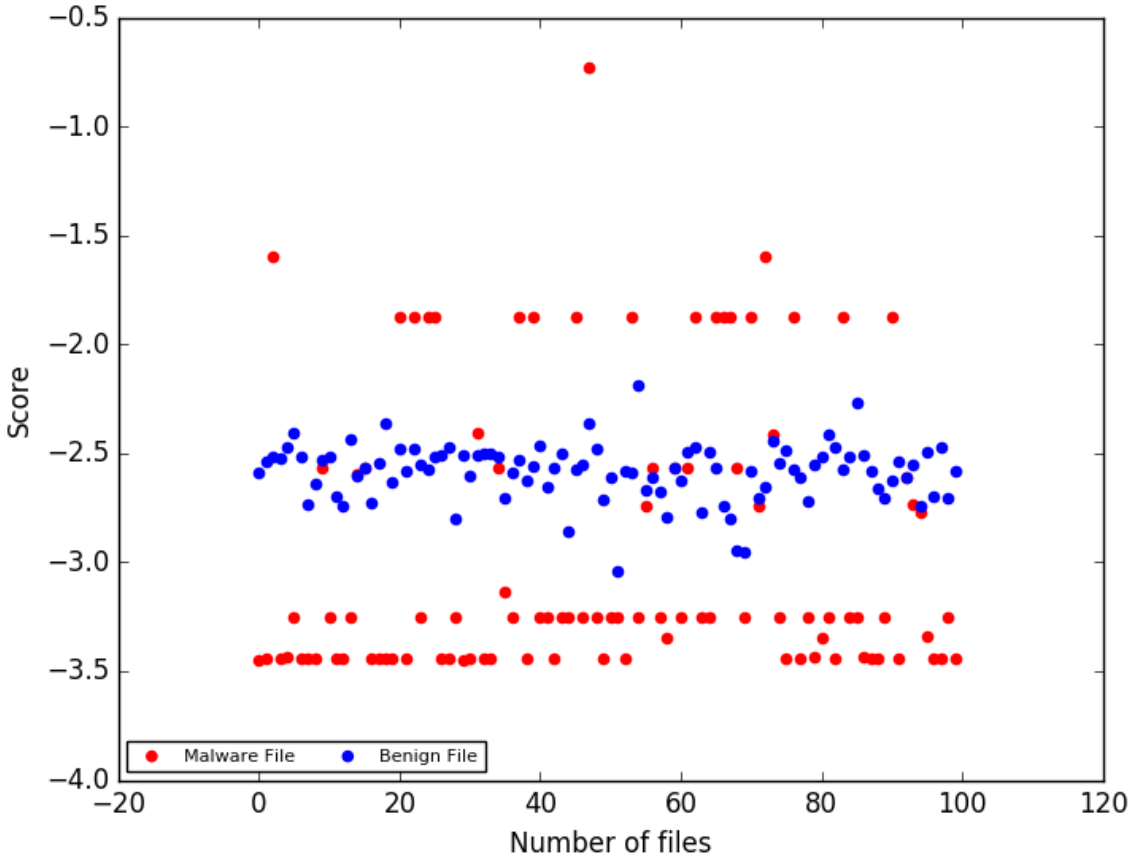

Figure A.49: Zbot Scatter Plot for 10% Data Poisoning

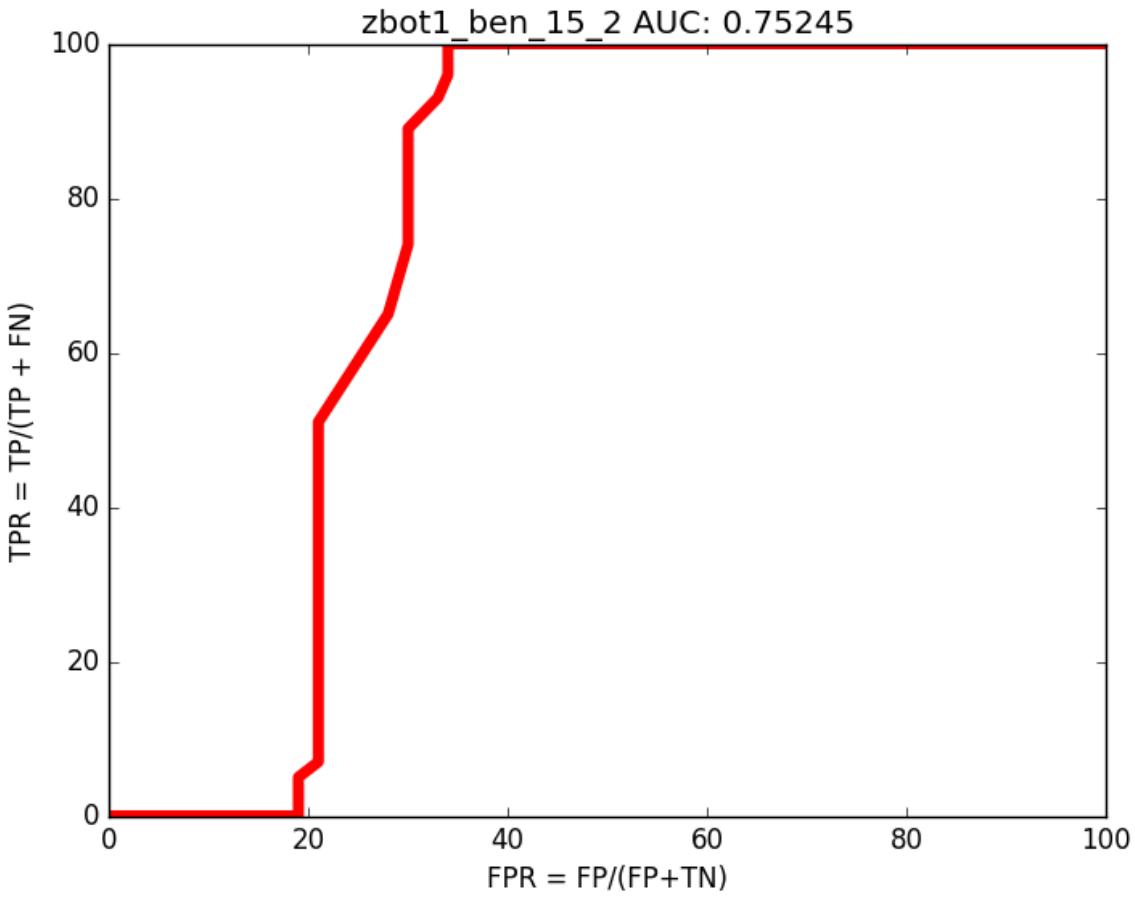

Figure A.50: Zbot ROC Curve and AUC with 15% Data Poisoning

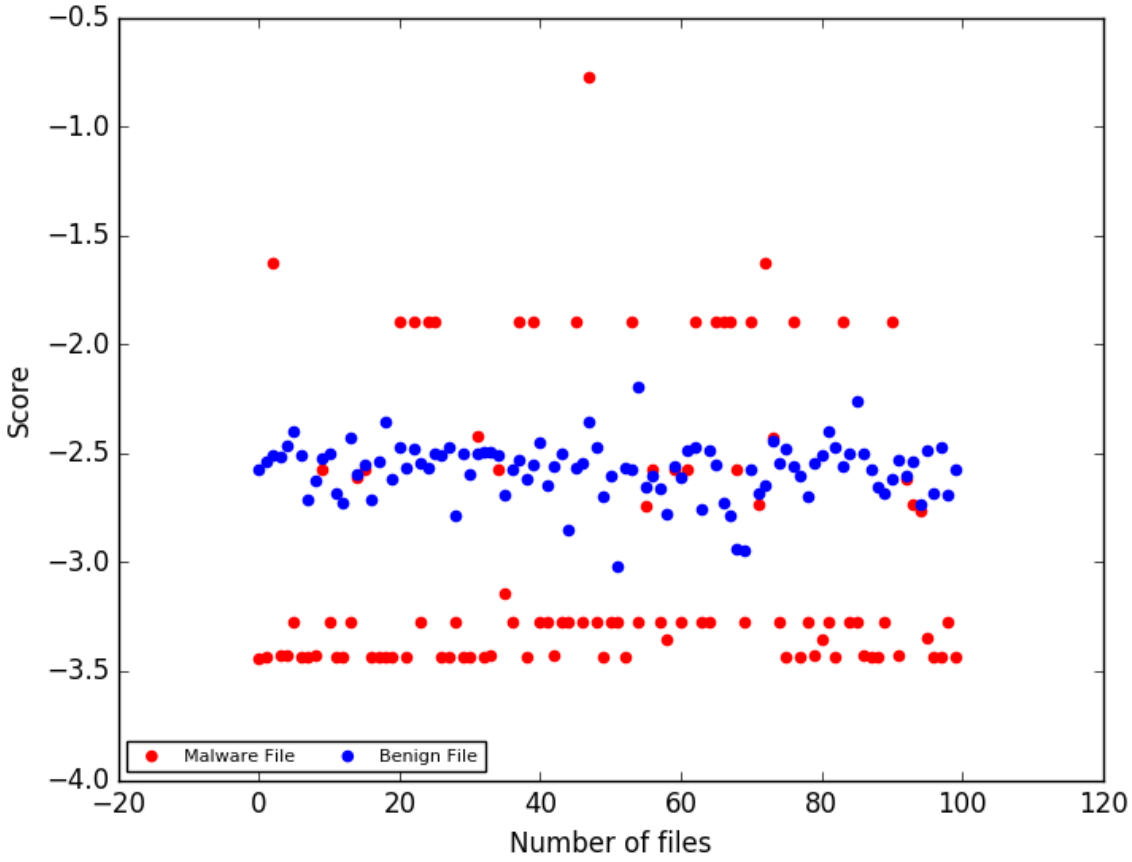

Figure A.51: Zbot Scatter Plot for 15% Data Poisoning

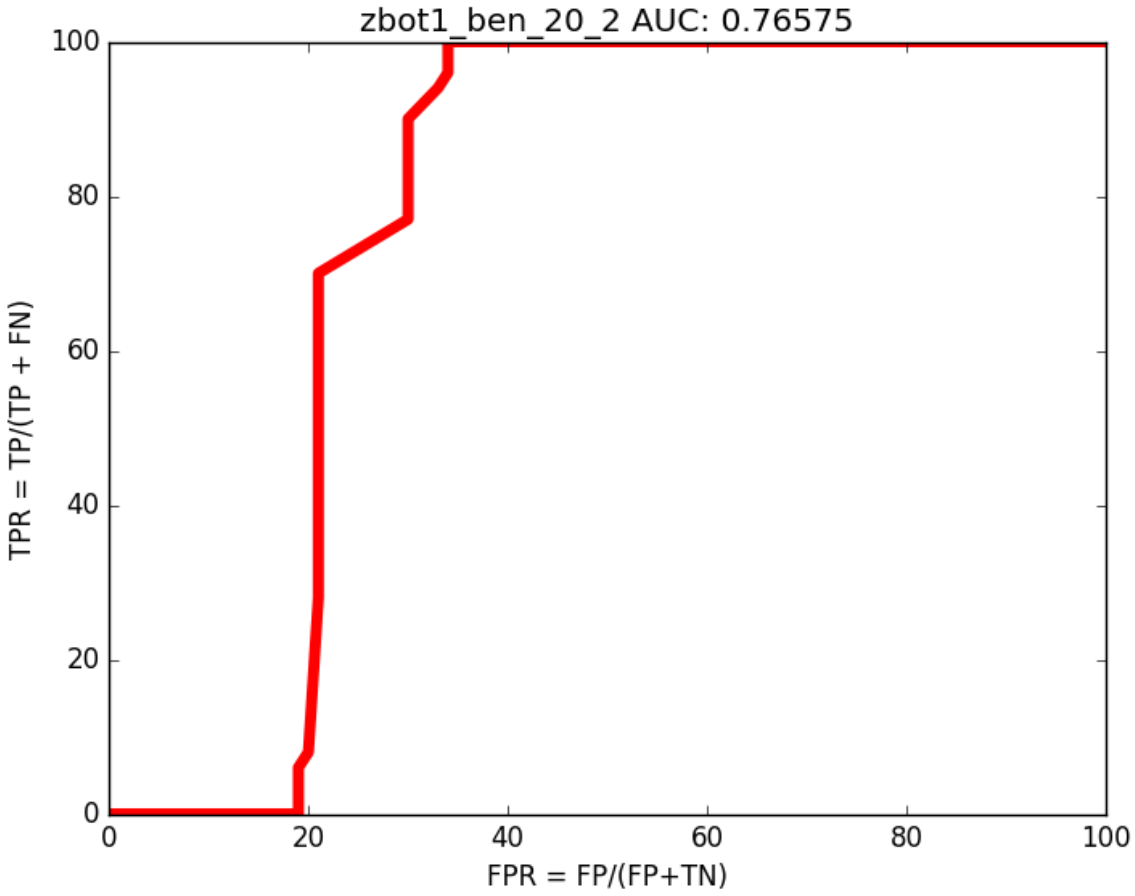

Figure A.52: Zbot ROC Curve and AUC with 20% Data Poisoning

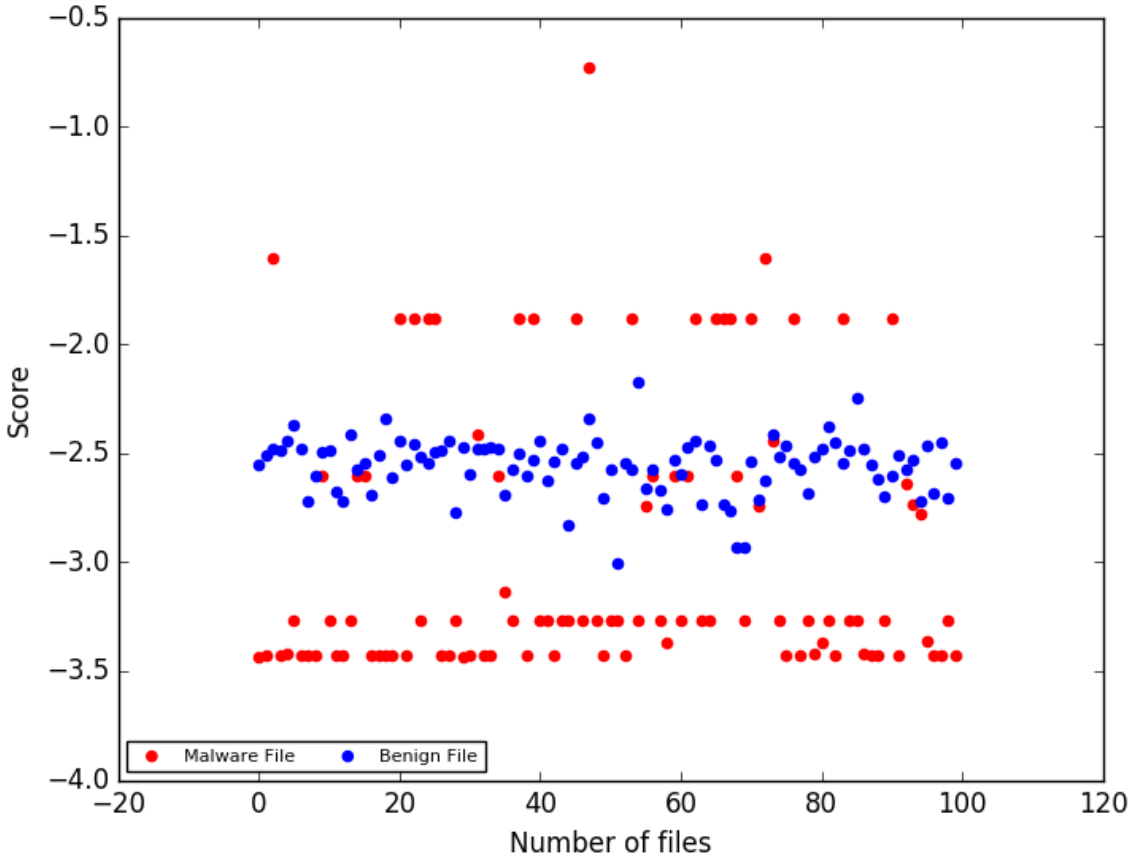

Figure A.53: Zbot Scatter Plot for 20% Data Poisoning

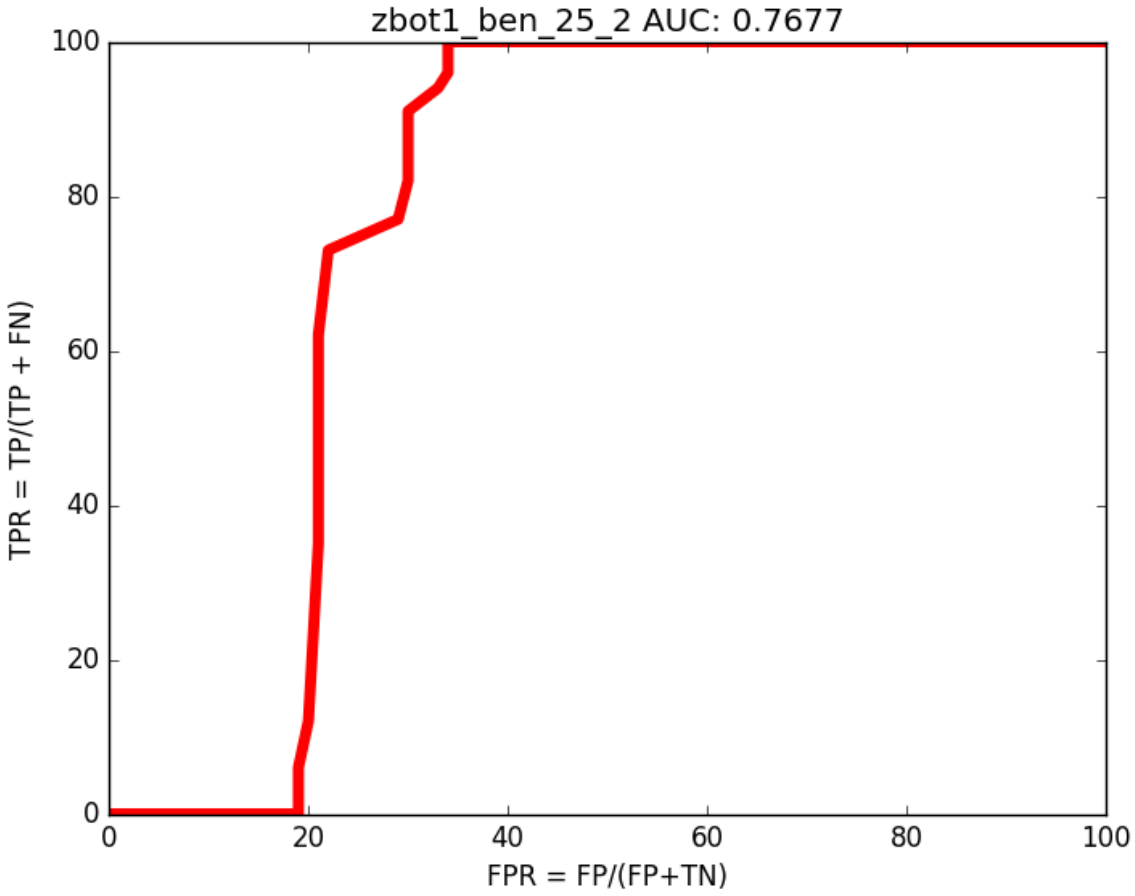

Figure A.54: Zbot ROC Curve and AUC with 25% Data Poisoning

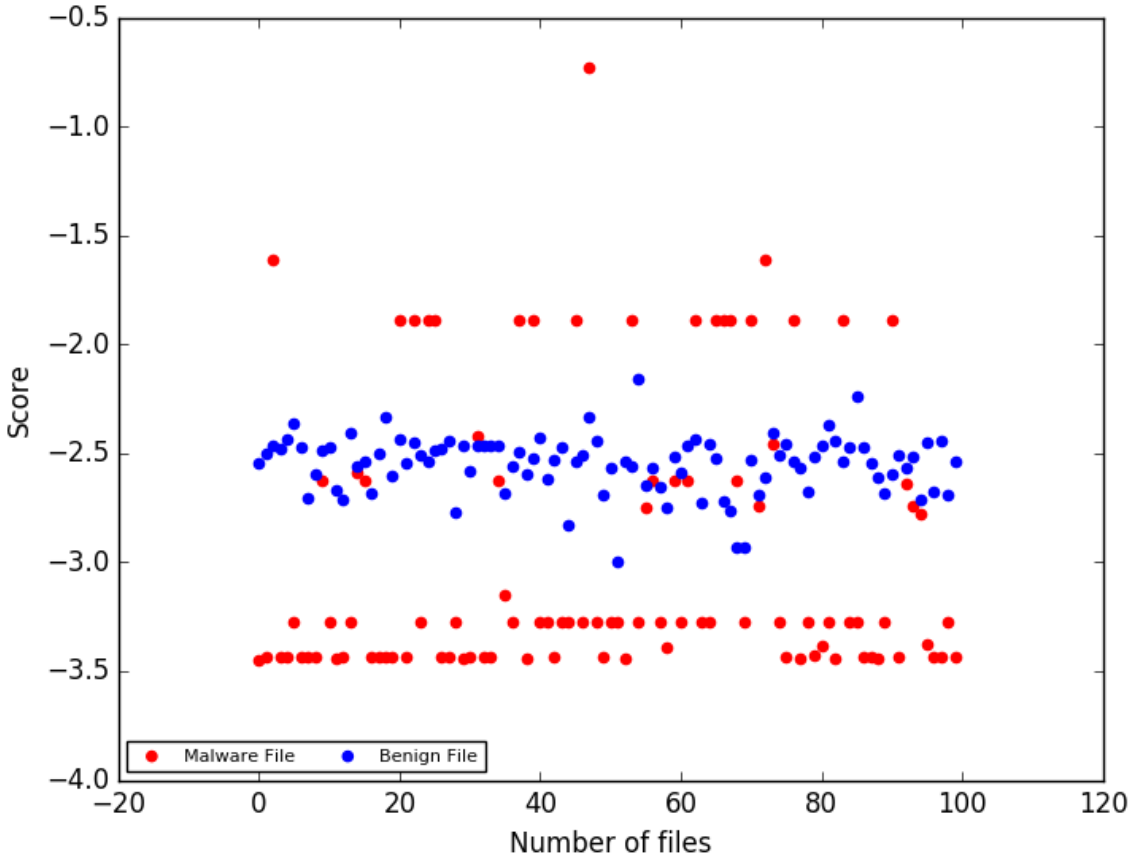

Figure A.55: Zbot Scatter Plot for 25% Data Poisoning

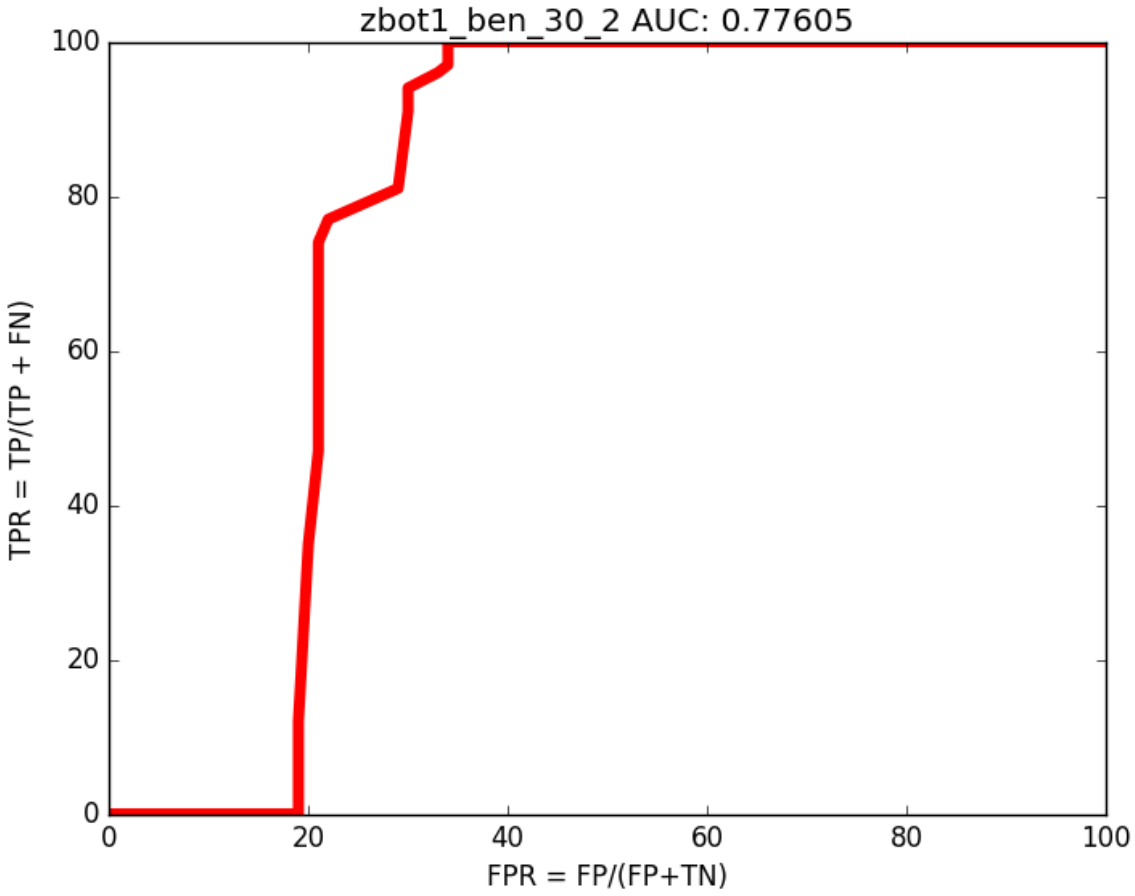

Figure A.56: Zbot ROC Curve and AUC with 30% Data Poisoning

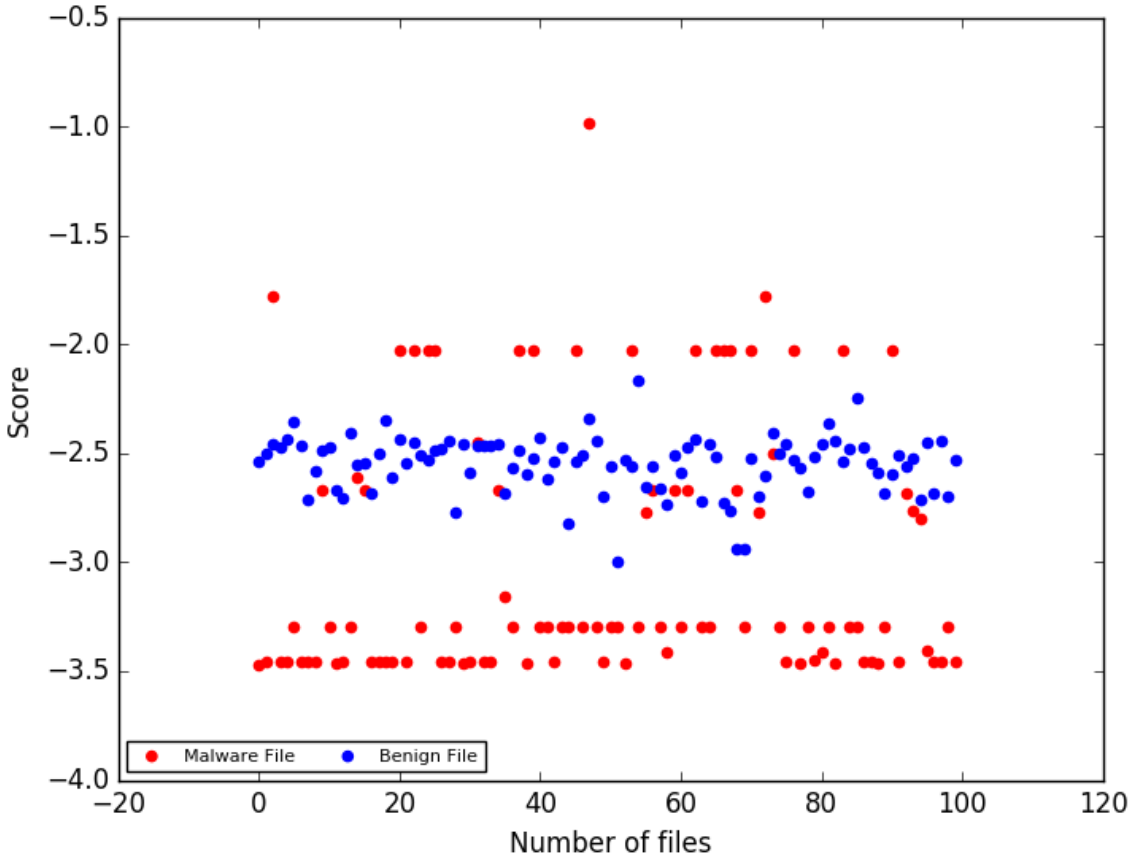

Figure A.57: Zbot Scatter Plot for 30% Data Poisoning

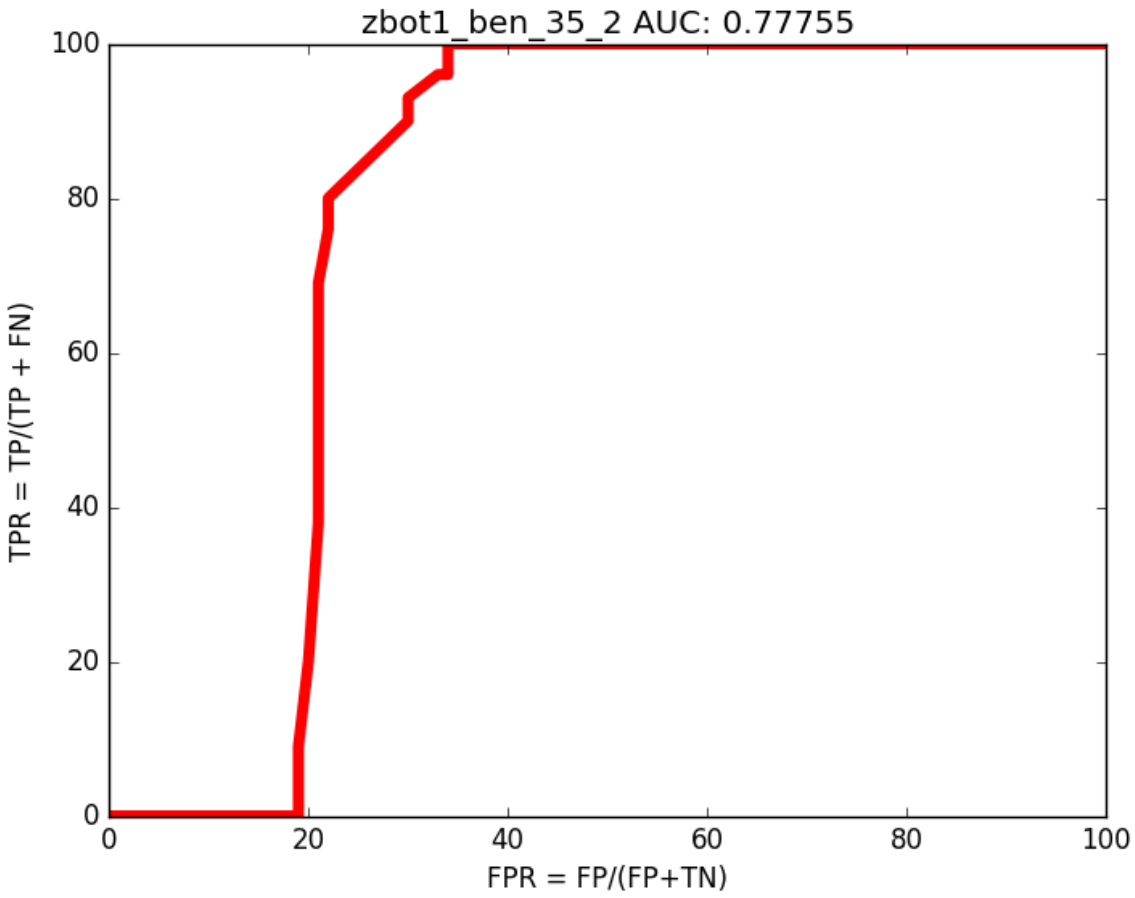

Figure A.58: Zbot ROC Curve and AUC with 35% Data Poisoning

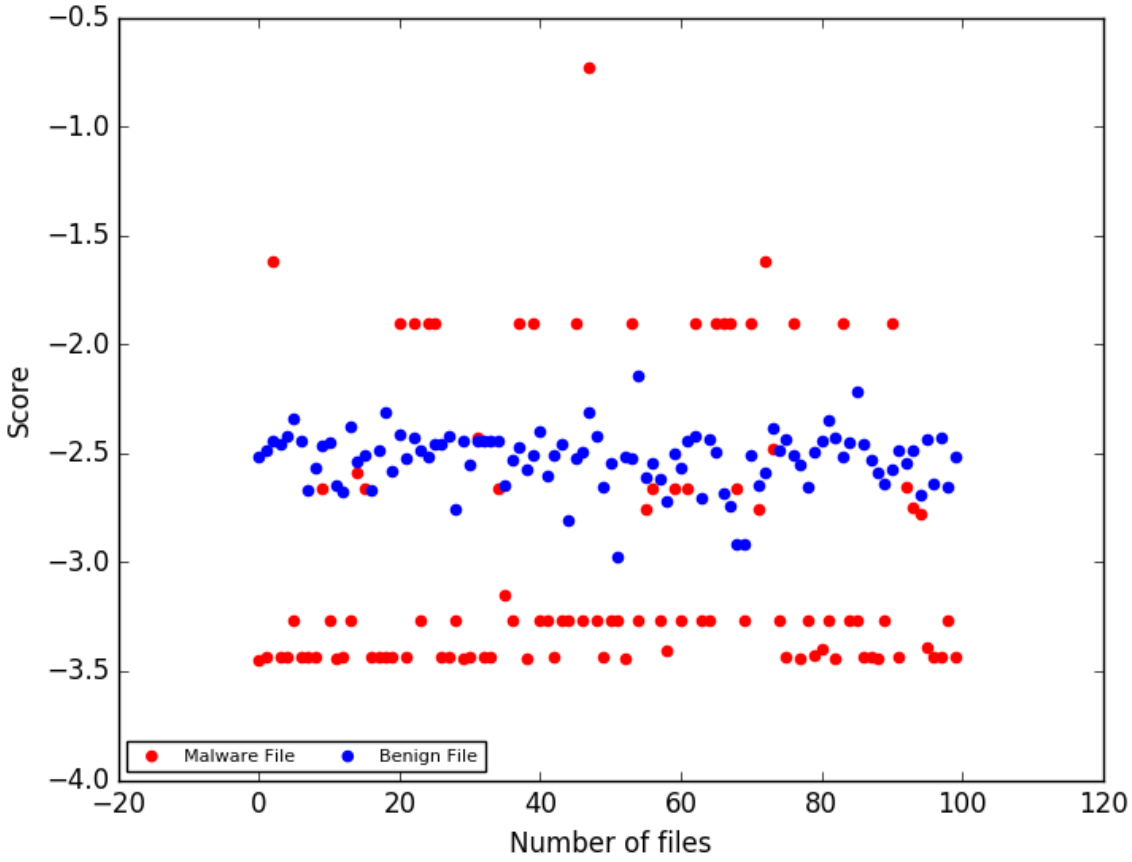

Figure A.59: Zbot Scatter Plot for 35% Data Poisoning

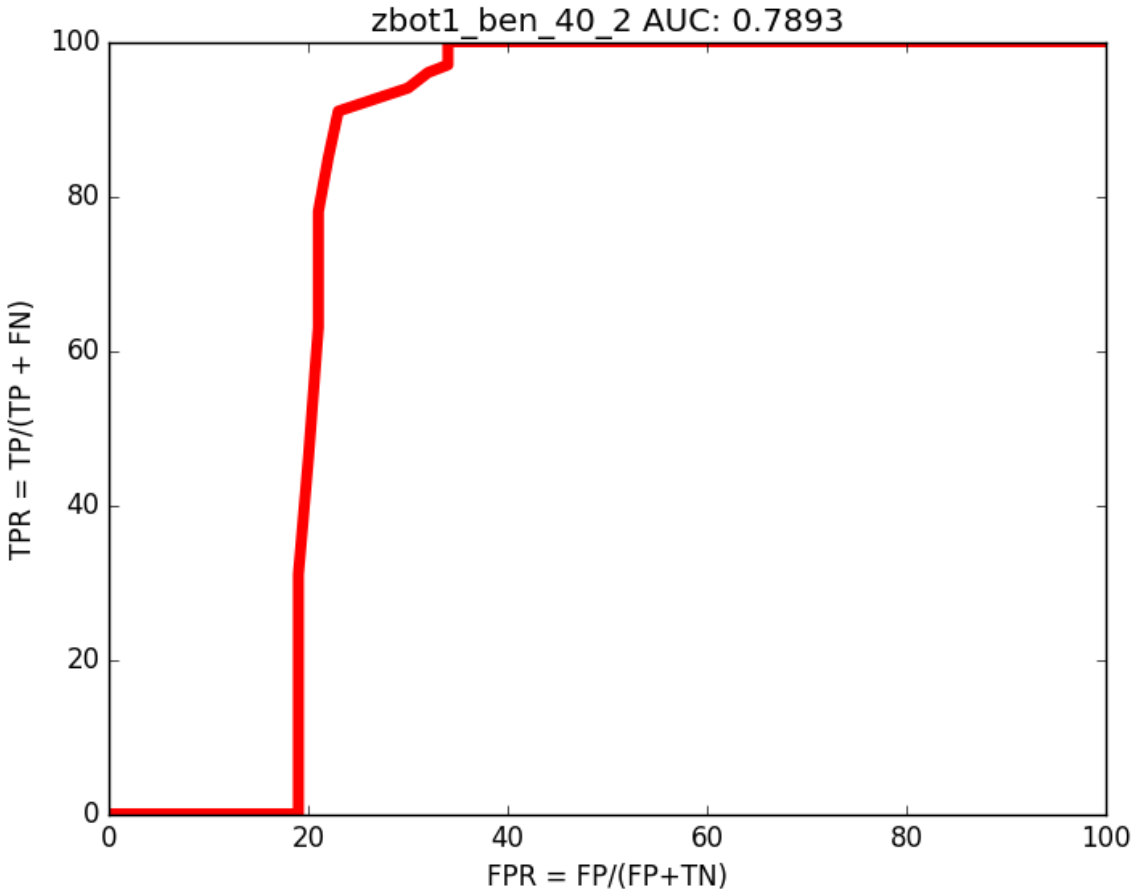

Figure A.60: Zbot ROC Curve and AUC with 40% Data Poisoning

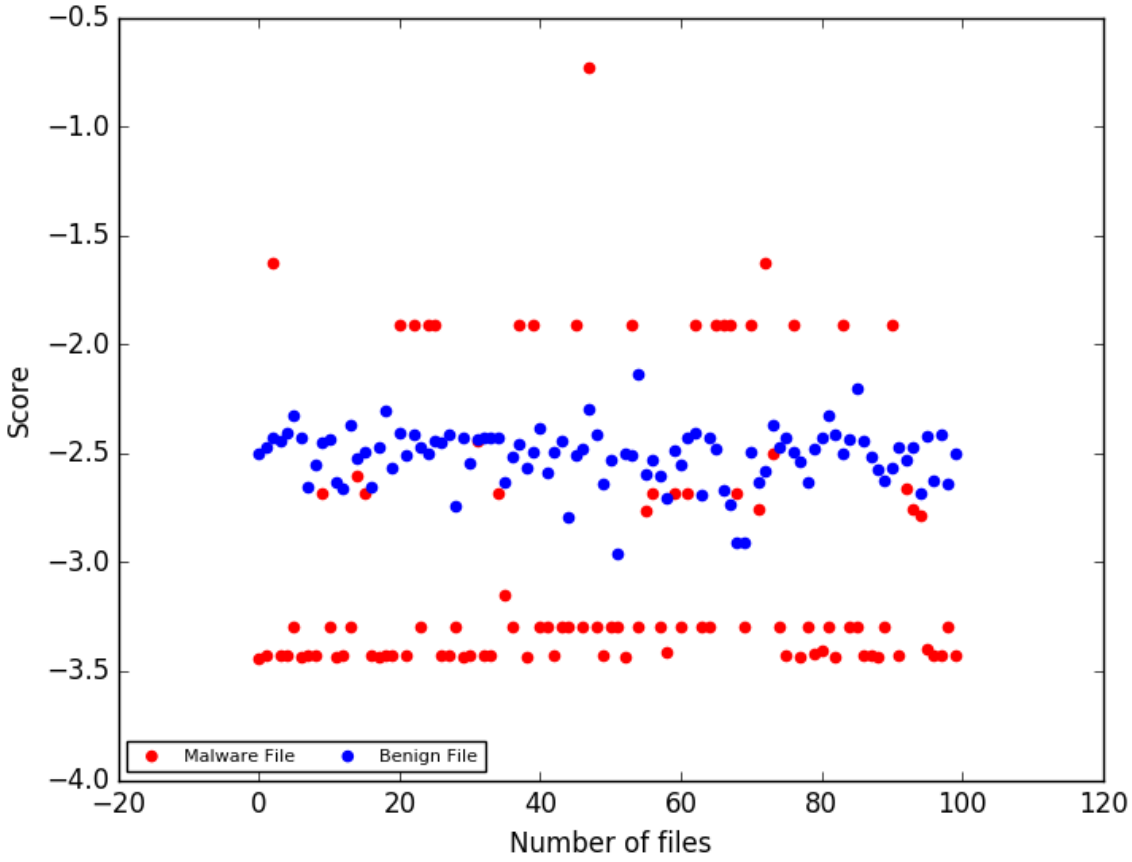

Figure A.61: Zbot Scatter Plot for 40% Data Poisoning

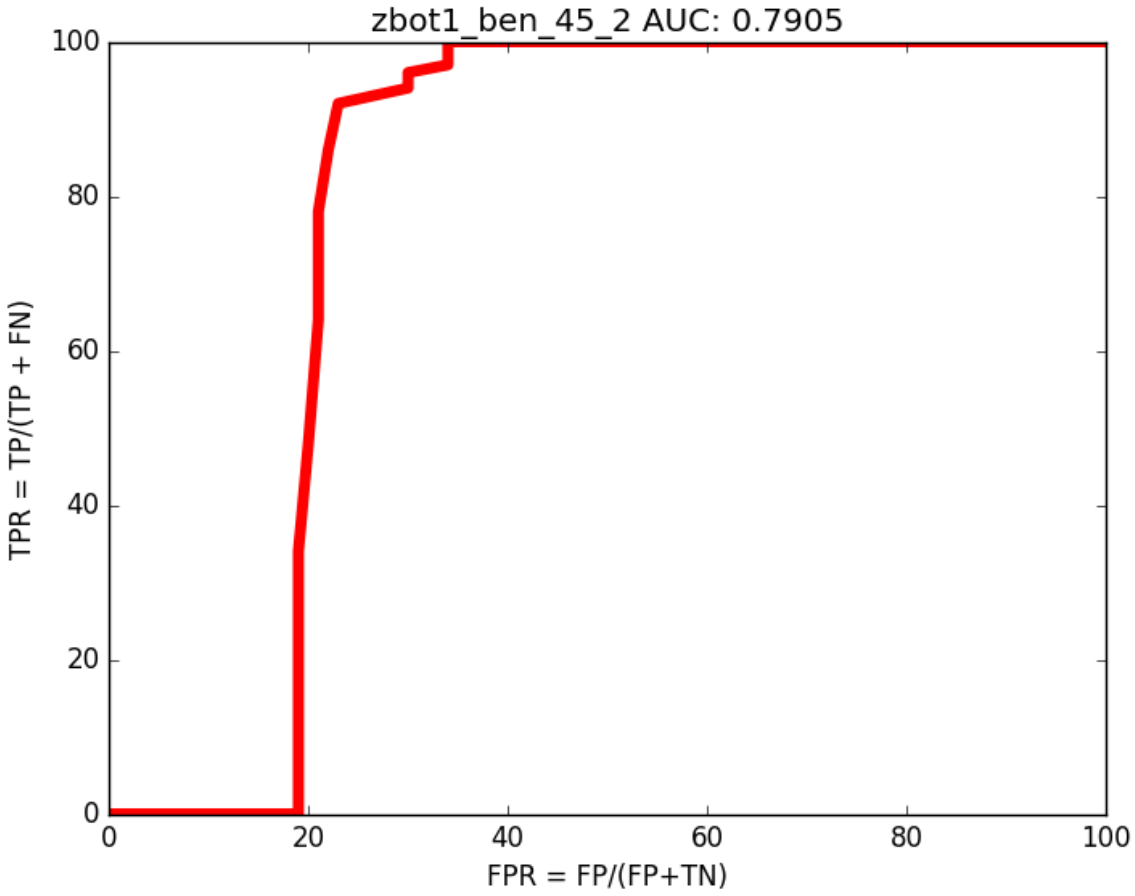

Figure A.62: Zbot ROC Curve and AUC with 45% Data Poisoning

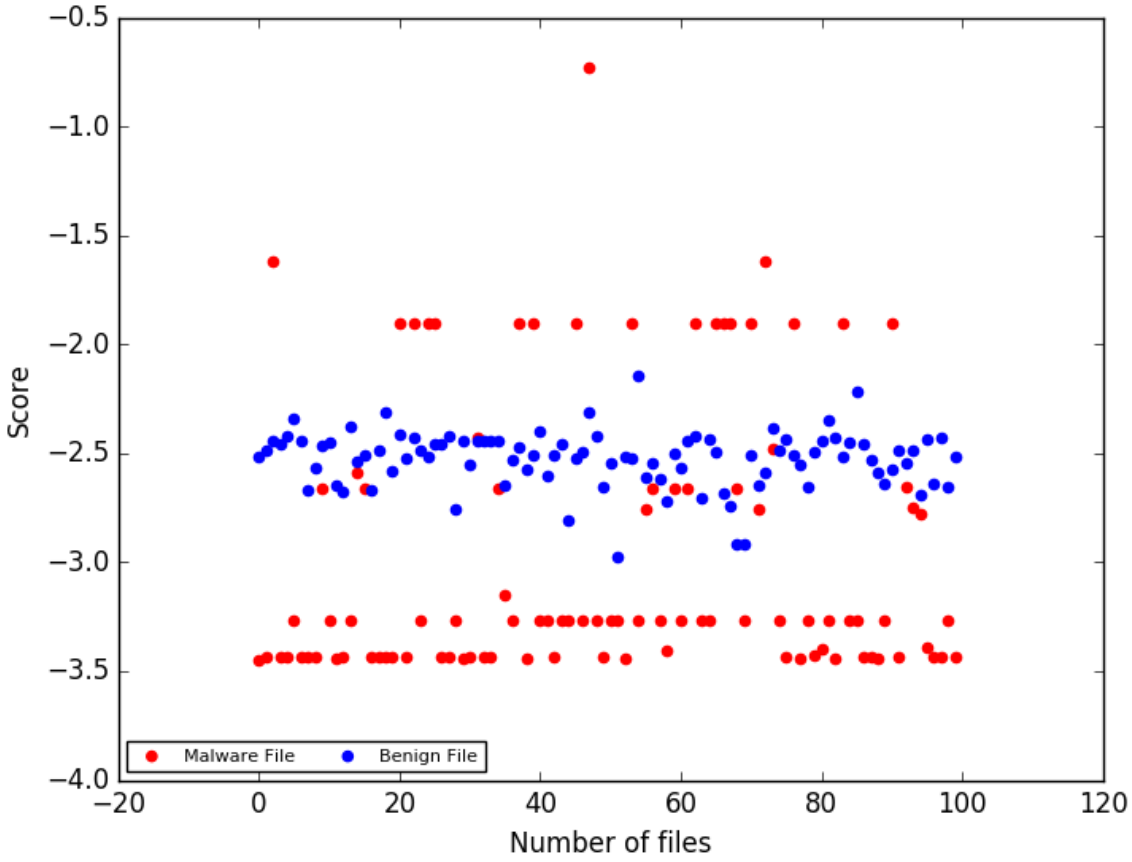

Figure A.63: Zbot Scatter Plot for 35% Data Poisoning