

DEFENSE MECHANISMS AGAINST DISTRIBUTED DENIAL OF SERVICE (DDOS) ATTACKS

A THESIS

*Submitted in partial fulfilment of the
requirements for the award of the degree*

of

DOCTOR OF PHILOSOPHY

in

ELECTRONICS AND COMPUTER ENGINEERING

by

BRIJ BHOOSHAN GUPTA



DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE-247 667 (INDIA)

MARCH, 2011

**©INDIAN INSTITUTE OF TECHNOLOGY ROORKEE, ROORKEE- 2011
ALL RIGHTS RESERVED**



INDIAN INSTITUTE OF TECHNOLOGY ROORKEE ROORKEE

CANDIDATE'S DECLARATION

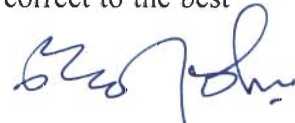
I hereby certify that the work which is being presented in the thesis entitled **DEFENSE MECHANISMS AGAINST DISTRIBUTED DENIAL OF SERVICE (DDOS) ATTACKS** in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy and submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee is an authentic record of my own work carried out during a period from January, 2006 to March, 2011 under the supervision of Prof. R. C. Joshi, and Prof. Manoj Misra, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.


(BRIJ BHOOSHAN GUPTA)

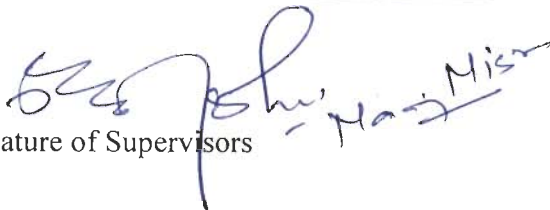
This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

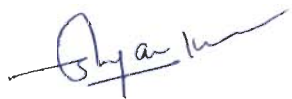

(Manoj Misra)
Supervisor


(R. C. Joshi)
Supervisor

Date: 8.3.11

The Ph.D. Viva-Voice Examination of Mr. BRIJ BHOOSHAN GUPTA, Research Scholar, has been held on 8.11.11


Signature of Supervisors


Signature of External Examiner

ABSTRACT

In past two decades, Internet has revolutionized almost every facet of our lives. Government, commercial, and educational organizations depend on Internet to such an extent that day-to-day operations are significantly hindered when the network is “down”. Almost all the important services such as banking, transportation, stock trade, medicine, education, etc are extended to Internet now. Everything is available on a click of a mouse. But unfortunately at the same time, the prosperity of the Internet also attracts abusers and malicious attackers. Since the original aim of Internet was to provide an open network for researchers to share their research resources, therefore openness and growth of the network were the design priorities while security issues were of less concern. Abusers and malicious attackers take advantage of this to launch attacks and intrusions to the Internet based services. Internet based attacks can be launched anywhere in the world, and unfortunately no Internet based service is immune to these attacks. These attacks lead to heavy financial losses, delays, and customer dissatisfaction. Trustworthiness and security of the Internet not only benefits on-line businesses, but is also an issue for national safety. Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks are currently amongst the most problematic Internet security threats. These attacks are critical as they aim at denying or degrading services for a legitimate user.

DDoS attacks can be defined as any form of attempt that forces some system component to limit, or even halt, normal services. The traditional purpose and impact of DDoS is to prevent or deny the legitimate use of computer or network resources. Regardless of significant advances that have been made in network management and security, Internet connected systems face a consistent threat from DDoS attacks. Over the recent years, several research works have proposed solutions for handling DDoS attacks. A lot of them claim to be best in absence of benchmarks, but none of them is able to withstand the advancing attack techniques. Researchers have come up with more and more specific solutions to the DDoS problem. However, existing DDoS attack tools also keep on improving using new attack techniques. Hence there is a critical need of addressing this issue to achieve a long lasting solution. Accordingly, the thesis focuses on the research towards developing a robust and effective solution to counteract DDoS attacks and is organized as follows.

In the first part of the thesis, a brief introduction of the research work, motivation, and problem formulation is given. Then it is followed by a state of the art literature review. Following that, we describe our proposed approach, ‘flow-volume based approach (FVBA)’, for detecting variety of DDoS attacks. In the proposed mechanism, attacks are detected by monitoring abrupt traffic changes inside ISP network. The flow-volume based approach (FVBA) constructs profile of the traffic normally seen in the network, and identifies anomalies whenever traffic goes out of profile. Tolerance factor which is a tunable parameter is used to make proposed detection system adaptable to the varying network conditions and attack loads in real time. Proposed scheme is evaluated through extensive simulations using NS-2 network simulator on Linux platform. Network topologies similar to Internet, used for simulation, are generated using Transit-Stub model of GT-ITM topology generator. Five performance metrics, i.e. detection rate, false positive rate and receiver operating characteristics (ROC), Goodput and NPSR are used to evaluate the performance of proposed scheme and it is compared with existing volume based approaches. The results show that proposed scheme gives 10-30% improvement in detection rate over earlier volume based schemes. For validating performance of proposed scheme, KDD 99, a publicly available benchmark dataset is used.

The flow-volume based approach (FVBA) though performs better than previous methods, it can be further improved by taking the heteroskedastic nature of DDoS attack traffic. Hence in subsequent section, the thesis deals with nonlinear statistical methods for fast and effective detection of flooding DDoS attacks. In this research work, the Generalized Autoregressive Conditional Heteroskedastic (GARCH) model, which is a commonly used statistical modeling technique for financial time series, is used as a new technique for detecting DDoS attacks. Our studies show that this non linear volatility model gives 4 to 5.5% improvement in detection performance from earlier models like linear prediction. The results reveal that time series modeling of DDoS attacks does show a lot of promise. Detection performance of GARCH model based detection scheme is also compared with FVBA scheme. Results show that GARCH model based detection scheme shows marginal improvement in detection rate over FVBA.

The thesis also deals with predicting number of zombies involved in a DDoS attack. A real time estimation of the number of zombies in DDoS scenario is helpful to suppress the effect of attack by choosing predicted number of most suspicious attack sources for either

filtering or rate limiting. We use various regression models i.e. linear, polynomial, exponential, power, logarithmic and multiple to predict number of zombies in a DDoS attack. Various statistical performance measures are used to evaluate the performance of various regression models. A comparative study of different regression models for predicting number of zombies is performed. Generally the method being promising, simulation results show that multiple regression model performs better than other regression models.

The other new proposal, which is a different method for predicting number of zombies involved in a DDoS attack, is presented next to the above section. The proposed method uses feed forward neural networks of different sizes to predict number of zombies. The sample data used to train and test the feed forward neural networks is generated using NS-2 network simulator running on Linux platform. Mean square error (MSE) is used to compare the performance of various feed forward neural networks. For the prediction of the number of zombies in a DDoS attack, three feed forward neural networks of different sizes have been tested. For the problem at hand, feed forward networks with 5, 10 and 15 neurons are used. Selected feed forward networks are compared for their prediction performance. The simulation results show that feed forward networks with 10 neurons perform better than the others, as it is able to predict number of zombies involved in a DDoS attack with very less error. Prediction performance of ANN based scheme has also been compared with regression based scheme and results show that ANN based scheme performs better than regression based scheme when attack is more severe.

The other main issue presented in the thesis is our approach for estimating strength of a DDoS attack. Estimating strength of attack is helpful to suppress the effect of attack, as it enables a security administrator to effectively equip his arsenal with proper defense mechanisms for fighting against DDoS threat according to the strength of attack. Hence in this research work, we use regression analysis to investigate suitability of various regression models i.e. linear, polynomial, exponential, power, logarithmic and multiple to estimate strength of a DDoS attack. A comparative study has also been performed using different regression models for estimating strength of DDoS attack. The simulation results show that multiple regression model performs better to estimate strength of a DDoS attack.

Lastly summary of the contributions made in the thesis and the future scope of the work are presented. All in all, the thesis expounds the various approaches we proposed for defending against variety of DDoS attacks.

Acknowledgements

I would like to express my deepest gratitude to my learned supervisors Dr. R. C. Joshi and Dr. Manoj Misra for their encouragement, painstaking supervision, innovative suggestions and invaluable help during the entire period of my Ph.D. Their insights and capability to judge things beyond text have strengthened this study significantly. Without their invaluable advices, guidance, and support on my research, I could not have achieved what I have done. They have been instrumental in shaping my approach towards accomplishing a task in hand and I learnt a great deal from them not only in research but also about aspects touching other aspects of life. I feel privileged to have worked under their supervision.

I would like to thank particularly Dr. Nadeem Jamali, my supervisor in University of Saskatchewan (UofS), Canada. I am grateful for all the guidance, comments, pieces of advice, and time he gave me since I started my research in the Agent Laboratory. It is my real pleasure to have a chance to work with him. I also thank to technical staff of the Computer Science Department, UofS for maintaining excellent working facilities during my stay at UofS, Canada.

The cooperation and help extended by the Head and faculty members, Department of Electronics and Computer Engineering, Indian Institute of technology Roorkee is gratefully acknowledged. I also want to thank my research committee members for providing insightful and constructive comments. The help rendered by technical staff of the department is heartily acknowledged.

Sincere thanks go to Government of Canada, for supporting my research through Canadian Commonwealth Scholarship and Government of Canada Awards. I am obliged for research assistantships received from the Indian Institute of Technology Roorkee through Ministry of Human Resource Development (MHRD) scholarship. I am thankful and acknowledge the help extended by Microsoft, Council of Scientific and Industrial Research (CSIR) and IITR Heritage Foundation (IITRHF) for sponsoring my conference participations.

I would like to thank everyone who supported me intellectually, socially, emotionally, and academically during my stay at IIT Roorkee. I am greatly indebted to all of them. I wish to convey my deep appreciation to my fellow research scholars whose company itself has

been a great pleasure and a real help. I express my indebted thankfulness especially to Dereje Shiferaw for providing unconditional support as needed. In addition, I would like to thank Dr. Krishan Kumar, Dr. T. P. Sharma, Dr. Kulbhushan, Dr. Santosh, Dr. Balwinder, Manish Goyal, Angad, Emmanuel, Ashwini, Govind, and Manoj. Discussion and debate on a cup of tea really helped in thinking innovatively. I thank all friends at IIT Roorkee who made me feel at home while away from my family for many years. The moments spent with friends playing volleyball and cricket are unforgettable. Thanks friends for sharing joyful moments with me. I would specially like to thank my friends in Saskatoon who made my Canada trip enjoyable and memorable one. In particular, I would like to thank Yue Zhang, Mohammad Hashemian, Mayya, Xinghui Zhao, Cam, Kurt, Simon, Linda, Gerry, Shirley, Peter and Arlene who made my stay easy and comfortable in Canada. The beautiful and happy moments, I spent in the company of them will always be cherished in my memories.

I have no words to express appreciation for my family for their understanding and support during these years. I could not reach the important milestone of my life without their support and encouragement.

I owe a debt of gratitude to my parents who brought me up to be a confident individual and showered unconditional love on me. The values you instilled in me give me strength and make me a better person. My father, P. C. Gupta, taught me that it was worth pursuing things I believed in. My mother, Vimla Gupta, was always there when I needed the type of support that only a mother can offer. I can never thank enough my parents and my brother for being a constant source of love and encouragement and strength. They deserve special thanks for just being who they are and taking me to this stage in life.

And above all, I am thankful to the Almighty whose divine grace gave me the required courage, strength and perseverance to overcome various obstacles that stood in my way.

Brij Bhooshan Gupta

Contents

| | |
|---|-------------|
| Candidate’s Declaration | i |
| Abstract | iii |
| Acknowledgements | vii |
| Contents | ix |
| List of Abbreviations | xv |
| List of Figures | xvii |
| List of Tables | xxi |
| | |
| Chapter 1 Introduction | 1 |
| 1.1 Introduction | 1 |
| 1.2 Motivation | 5 |
| 1.3 Statement of the Problem..... | 9 |
| 1.4 Organization of Thesis..... | 9 |
| | |
| Chapter 2 Background and Literature Survey | 13 |
| 2.1 Background and Overview..... | 13 |
| 2.1.1 Denial of Service Attack | 14 |
| 2.1.2 Distributed Denial of Service Attack..... | 14 |
| 2.1.3 Distributed Reflector Denial of Service Attacks..... | 16 |
| 2.2 Major Causes Responsible for DDoS Attacks | 16 |
| 2.3 Targeted Resources by Distributed Denial of Service attack..... | 17 |
| 2.4 DDoS attack: Modus Operandi..... | 18 |
| 2.5 DDoS Attack Tools..... | 20 |
| 2.6 Classification of Attack Mechanisms | 23 |
| 2.6.1 Based on Attacking Methods..... | 24 |
| 2.6.2 Based on Weaknesses Exploited..... | 25 |
| 2.6.3 Based on Connection Establishment..... | 29 |
| 2.6.4 Based on Attack Rate..... | 30 |

| | | |
|-------|--|----|
| 2.6.5 | Based on Attack Traffic Distribution..... | 30 |
| 2.6.6 | Based on Attack Packets Used..... | 31 |
| 2.6.7 | Based on Protocol Used..... | 31 |
| 2.7 | Defense Challenges and Principles | 31 |
| 2.8 | Classification of DDoS Defense Mechanisms | 33 |
| 2.8.1 | Based on Activity Deployed..... | 33 |
| 2.8.2 | Based on Degree of Deployment..... | 55 |
| 2.8.3 | Based on Deployment Point or Location..... | 56 |
| 2.8.4 | Based on Degree of Cooperation..... | 58 |
| 2.9 | Predicting Number of Zombies in a DDoS Attack..... | 59 |
| 2.10 | Research Gaps..... | 59 |
| 2.11 | Chapter Summary..... | 62 |

Chapter 3 Detecting Distributed Denial of Service Attacks using Flow-Volume 63

Based Approach

| | | |
|-------|---|----|
| 3.1 | Introduction..... | 63 |
| 3.2 | DDoS Attacks Detection Model | 64 |
| 3.2.1 | Choice of Traffic Parameter | 64 |
| 3.2.2 | Choice of Polling Interval | 67 |
| 3.3 | Flow-Volume based Attack Detection Scheme | 68 |
| 3.3.1 | System Model..... | 68 |
| 3.3.2 | DDoS Detection Scheme..... | 68 |
| 3.4 | Performance Evaluation..... | 72 |
| 3.4.1 | Simulation Model..... | 72 |
| 3.4.2 | Performance Metrics..... | 74 |
| 3.5 | Results and discussion | 75 |
| 3.5.1 | Degradation of Goodput with Attack..... | 75 |
| 3.5.2 | Degradation of NPSR with Attack..... | 76 |
| 3.5.3 | Detection of Attack..... | 76 |
| 3.5.4 | Results with KDD 99 Dataset..... | 78 |
| 3.5.5 | Comparison of Volume based Approaches..... | 80 |

| | | |
|--|--|------------|
| 2.5.6 | Comparison of Entropy based Approaches..... | 82 |
| 3.6 | Chapter Summary..... | 84 |
| Chapter 4 Detecting distributed Denial of Service Attacks using GARCH Model | | 85 |
| 4.1 | Introduction..... | 85 |
| 4.2 | Time Series Analysis..... | 85 |
| 4.2.1 | Stationarity..... | 87 |
| 4.2.2 | Autocorrelation..... | 87 |
| 4.3 | Non-linear Time Series Modeling..... | 88 |
| 4.4 | GARCH Model | 89 |
| 4.5 | Test for Heteroskedasticity | 90 |
| 4.5.1 | Engle's ARCH test | 91 |
| 4.5.2 | Ljung-Box-Pierce Q-Test..... | 92 |
| 4.6 | GARCH Model based DDoS Attack Detection..... | 92 |
| 4.6.1 | Choice of Parameter for Modeling Flooding Attacks..... | 92 |
| 4.6.2 | Detection Algorithm..... | 95 |
| 4.7 | Performance Evaluation..... | 96 |
| 4.7.1 | Experiment Setup..... | 96 |
| 4.7.2 | Performance Metrics..... | 96 |
| 4.7.3 | Prediction Error..... | 96 |
| 4.7.4 | Results and Discussion..... | 97 |
| 4.8 | Comparison between FVBA and GARCH based DDoS Attack Detection Schemes..... | 103 |
| 4.9 | Chapter Summary..... | 103 |
| Chapter 5 Predicting Number of Zombies in a DDoS Attack using Various Regression Models | | 105 |
| 5.1 | Introduction..... | 105 |
| 5.2 | Regression Models..... | 106 |
| 5.2.1 | Types of Regression model used | 106 |

| | | |
|---|---|------------|
| 5.2.2 | Estimating Number of Zombies | 109 |
| 5.3 | Statistical Performance Measures..... | 109 |
| 5.4 | Simulation Setup | 111 |
| 5.5 | Model Development and Experimental Analysis..... | 111 |
| 5.6 | Results and Discussion..... | 117 |
| 5.7 | Chapter Summary..... | 129 |
| Chapter 6 ANN Based Scheme to Predict Number of Zombies in a DDoS Attack | | 131 |
| 6.1 | Introduction..... | 131 |
| 6.2 | Artificial Neural Network (ANN)..... | 132 |
| 6.3 | Experimental Setup and Performance Analysis..... | 136 |
| 6.4 | Results and Discussion..... | 137 |
| 6.5 | Comparison between Regression and ANN based Schemes for Predicting number of Zombies in a DDoS attack..... | 144 |
| 6.6 | Chapter Summary..... | 145 |
| Chapter 7 Estimating Strength of a DDoS Attack using Regression Models | | 147 |
| 7.1 | Introduction..... | 147 |
| 7.2 | Regression Models..... | 148 |
| 7.3 | Statistical Performance Measures..... | 148 |
| 7.4 | Simulation Setup..... | 148 |
| 7.5 | Model Development and Experimental Analysis..... | 149 |
| 7.6 | Results and Discussion..... | 156 |
| 7.7 | Chapter Summary..... | 165 |
| Chapter 8 Conclusion and Future Work | | 167 |
| 8.1 | Contributions of the Thesis | 167 |
| 8.2 | Scope for Future Work..... | 169 |

| | |
|---------------------------------------|------------|
| Appendix-A | 171 |
| References | 177 |
| Author's research Publications | 195 |

List of Abbreviations

| | |
|--------|--|
| ACC | Aggregate based Congestion Control |
| AIMD | Additive Increase Multiplicative Decrease |
| AODV | Ad hoc On Demand Distance Vector |
| ARP | Address Resolution Protocol |
| AS | Autonomous System |
| ATA | Algebraic based Traceback |
| BA | Bandwidth Aggregate |
| BGP | Border Gateway Protocol |
| BSD | Berkeley Software Distribution |
| CBR | Constant Bit Rate |
| CBQ | Class Based Queuing |
| CDN | Content Distribution Network |
| CERT | Computer Emergency Response Team |
| CHOKe | CHOOse and Keep for Responsive Flows, CHOOse and kill for Unresponsive Flows |
| CNN | Cable News Network |
| CSI | Computer Security Institute |
| CUMSUM | Cumulative Sum |
| DDoS | Distributed Denial-of-Service |
| DNS | domain Name system |
| DoS | Denial-of-Service |
| DPM | Deterministic Packet Marking |
| DRDoS | Distributed Reflector Denial-of-Service |
| FBI | Federal Bureau of Investigation |
| FN | False Negatives |
| FP | False Positives |
| FQ | Fair Queuing |
| FTP | File Transfer Protocol |
| FRED | Flow random Early Drop |

| | |
|---------|---|
| Gb | Giga bits |
| GT-ITM | Georgia Tech Internetwork Topology Models |
| HRFD | High Rate Flooding DDoS attacks |
| ICMP | Internet Control Message Protocol |
| IDS | Intrusion Detection System |
| IntServ | Integrated Services |
| IP | Internet Protocol |
| ISP | Internet Service Provider |
| Mbps | Megabits per second |
| ms | millisecond |
| MULTOPS | Multi-Level Tree for Online Packet Statistics |
| Nam | Network Animator |
| NPSR | Normal Packet Survival Ratio |
| NS | Network Simulator |
| PD | Preferential Dropping |
| PKI | Public Key Infrastructure |
| PPM | Probabilistic Packet Marking |
| QoS | Quality of Service |
| RED | Random Early Detection |
| ROC | Receiver Operating Characteristic |
| RTT | Round Trip Time |
| SAVE | source Address Validity Enforcement |
| SFQ | Stochastic Fair Queuing |
| SOS | Secure Overlay System |
| SRED | Stabilized Random Early Detection |
| TCP | Transmission control Protocol |
| TN | True Negative |
| TP | True Positive |
| UDP | User Datagram Protocol |
| UIPF | Ubiquitous Ingress/Egress Packet Filtering |
| VBA | Volume Based Approach |

List of Figures

| | | |
|---------|---|----|
| 1.1 | Internet Domain Survey Host Count..... | 2 |
| 1.2 | Estimated Internet Users in the World by Geographic regions according the survey of Mini Watts Marketing Group..... | 2 |
| 1.3 | Vulnerabilities reported since 1995..... | 3 |
| 1.4 | Increase in DDoS attack traffic..... | 4 |
| 2.1 | Denial of Service attack scenario..... | 14 |
| 2.2 | Distributed Denial of Service attack scenario..... | 15 |
| 2.3 | A hierarchical model of a DDoS attack..... | 19 |
| 2.4 | Classification of DDoS attack Mechanisms..... | 24 |
| 2.5 (a) | TCP 3-way handshaking..... | 26 |
| 2.5 (b) | TCP SYN attack..... | 26 |
| 2.6 | Smurf attack..... | 27 |
| 2.7 | Classification of DDoS Defense Mechanisms..... | 34 |
| 2.8 | DDoS defense deployment points | 57 |
| 3.1 | Temporal variation of volume measure when system is in normal condition, under low rate DDoS attack, and under high rate DDoS attack..... | 66 |
| 3.2 | Temporal variation of flow measure when system is in normal condition, under low rate DDoS attack, and under high rate DDoS attack..... | 66 |
| 3.3 | Variation of false alarm rate using varying polling intervals..... | 67 |
| 3.4 | Variation of Detection rate using varying polling intervals..... | 67 |
| 3.5 | Transit-Stub Network Model of Internet..... | 69 |
| 3.6 | FVBA architecture..... | 70 |
| 3.7 | DDoS attacks Detection Algorithm..... | 71 |
| 3.8 | Temporal variation of Goodput at different attack strengths..... | 75 |
| 3.9 | Temporal variation of NPSR at different attack strengths..... | 76 |
| 3.10 | Effect of detection tolerance factors on the detection and false positive rate..... | 77 |
| 3.11 | ROC curve showing the tradeoff between the detection rate and false positive rate of DDoS attacks..... | 78 |
| 3.12 | Overall detection results of testing for different protocol category..... | 79 |

| | | |
|------|--|-----|
| 3.13 | DoS Attacks detection summery in test dataset..... | 80 |
| 3.14 | Variation of detection rate of VBA and our detection system when attack with high rate is performed by varying number of zombie machines..... | 80 |
| 3.15 | Variation of detection rate of VBA and our detection system when attack with low rate is performed by varying number of zombie machines..... | 81 |
| 3.16 | Variation of detection rate of VBA and our detection system when attack with varying attack rate is performed using hundred zombie machines..... | 82 |
| 4.1 | Evidence of stationarity..... | 86 |
| 4.2 | Kurtosis for the sample dataset..... | 89 |
| 4.3 | Comparison of prediction error values..... | 97 |
| 4.4 | Detection delay in both GARCH(1,1) and LP model..... | 102 |
| 5.1 | Entropy variation with varied number of zombies..... | 112 |
| 5.2 | Flow variation with varied number of zombies..... | 112 |
| 5.3 | Volume variation with varied number of zombies..... | 113 |
| 5.4 | Regression equation and coefficient of determination for linear regression based model M1..... | 115 |
| 5.5 | Regression equation and coefficient of determination for polynomial regression based model M2..... | 115 |
| 5.6 | Regression equation and coefficient of determination for logarithmic regression based model M3..... | 116 |
| 5.7 | Regression equation and coefficient of determination for power regression based model M4..... | 116 |
| 5.8 | Regression equation and coefficient of determination for exponential regression based model M5..... | 117 |
| 5.9 | Comparison between actual number of zombies and predicted number of zombies using linear regression based model M1..... | 118 |
| 5.10 | Comparison between actual number of zombies and predicted number of zombies using polynomial regression based model M2..... | 118 |
| 5.11 | Comparison between actual number of zombies and predicted number of zombies using logarithmic regression based model M3..... | 119 |
| 5.12 | Comparison between actual number of zombies and predicted number of zombies using power regression based model M4..... | 119 |

| | | |
|------|--|-----|
| 5.13 | Comparison between actual number of zombies and predicted number of zombies using exponential regression based model M5..... | 120 |
| 5.14 | Comparison between actual number of zombies and predicted number of zombies using various regression models M1-M5..... | 120 |
| 5.15 | Summary of Residual error in various regression models..... | 122 |
| 5.16 | Value of coefficient of determination (R^2) for various regression models..... | 122 |
| 5.17 | Value of coefficient of correlation (CC) for various regression models..... | 123 |
| 5.18 | Value of sum of squared errors (SSE) for various regression models..... | 123 |
| 5.19 | Value of mean square error (MSE) for various regression models..... | 124 |
| 5.20 | Value of normalized mean square error (NMSE) for various regression models..... | 124 |
| 5.21 | Value of Nash–Sutcliffe efficiency index (η) for various regression models... | 125 |
| 5.22 | Comparison between actual number of zombies and predicted number of zombies using multiple regression model | 126 |
| 5.23 | Residual error in multiple regression model..... | 127 |
| 5.24 | Comparison between actual number of zombies and predicted number of zombies using polynomial and multiple regression model | 128 |
| 6.1 | Operation of single neuron..... | 133 |
| 6.2 | Sigmoid function..... | 134 |
| 6.3 | Fully connected, three layers feed forward network..... | 135 |
| 6.4 | Training performance of feed forward network (5-1) | 139 |
| 6.5 | Comparison between actual number of zombies and predicted number of zombies using feed forward neural network of size 5-1..... | 141 |
| 6.6 | Comparison between actual number of zombies and predicted number of zombies using Feed forward neural network of size 10-1..... | 141 |
| 6.7 | Comparison between actual number of zombies and predicted number of zombies using Feed forward neural network of size 15-1..... | 142 |
| 6.8 | Comparison of Absolute error using ANN and Regression based scheme..... | 145 |
| 7.1 | Entropy variation with varied attack strength..... | 150 |
| 7.2 | Flow variation with varied attack strength..... | 151 |
| 7.3 | Volume variation with varied attack strength..... | 151 |
| 7.4 | Regression equation and coefficient of determination for linear regression | 154 |

| | | |
|------|--|-----|
| | based model M1..... | |
| 7.5 | Regression equation and coefficient of determination for polynomial regression based model M2..... | 154 |
| 7.6 | Regression equation and coefficient of determination for logarithmic regression based model M3..... | 155 |
| 7.7 | Regression equation and coefficient of determination for power regression based model M4..... | 155 |
| 7.8 | Regression equation and coefficient of determination for exponential regression based model M5..... | 156 |
| 7.9 | Comparison between actual Strength of DDoS attack and predicted Strength of DDoS attack using linear regression based model M1..... | 157 |
| 7.10 | Comparison between actual Strength of DDoS attack and predicted Strength of DDoS attack using linear regression based model M2..... | 157 |
| 7.11 | Comparison between actual Strength of DDoS attack and predicted Strength of DDoS attack using linear regression based model M3..... | 158 |
| 7.12 | Comparison between actual Strength of DDoS attack and predicted Strength of DDoS attack using linear regression based model M4..... | 158 |
| 7.13 | Comparison between actual Strength of DDoS attack and predicted Strength of DDoS attack using linear regression based model M5..... | 159 |
| 7.14 | Comparison between actual Strength of DDoS attack and predicted Strength of DDoS attack using various regression models M1 to M5..... | 159 |
| 7.15 | Summary of residual error in various regression models..... | 161 |
| 7.16 | Comparison between actual DDoS attack strength and predicted DDoS attack strength using multiple regression model..... | 162 |
| 7.17 | Residual error in multiple regression model..... | 163 |
| 7.18 | Comparison between actual strength of a DDoS attack and predicted strength of a DDoS attack using polynomial and multiple regression model | 164 |
| A.1 | Distribution of TCP, UDP and ICMP connections in (a) training dataset, (b) testing dataset..... | 172 |
| A.2 | Distribution of total DoS and other attack connections in (a) training dataset, (b) testing dataset..... | 174 |

List of Tables

| | | |
|-----|--|-----|
| 2.1 | Summary of DDoS attack Tools..... | 22 |
| 2.2 | Comparison of various detection approaches classified based on detection method..... | 45 |
| 3.1 | Simulation parameters | 73 |
| 3.2 | Optimal values of tolerance factors to set normal profile..... | 79 |
| 4.1 | Engle ARCH Test..... | 91 |
| 4.2 | Ljung-Box-Pierce Q-Test..... | 92 |
| 4.3 | Prediction errors using GARCH(P,Q) for various value of P and Q..... | 94 |
| 4.4 | Mean prediction error for GARCH (1,1) and LP..... | 98 |
| 4.5 | Detection rate and false positive rate for GARCH (1,1) | 99 |
| 4.6 | Detection rate and false positive rate for LP..... | 100 |
| 4.7 | Detection Delay in seconds using GARCH (1,1) for flooding attacks..... | 101 |
| 4.8 | Detection Delay in seconds using LP for flooding attacks..... | 102 |
| 4.9 | Comparison between FVBA and GARCH model based scheme for DDoS attack detection..... | 103 |
| 5.1 | Deviation in entropy with actual number of zombies..... | 113 |
| 5.2 | Deviation in volume and flow with actual number of zombies..... | 114 |
| 5.3 | Summary of residual error for various regression models..... | 121 |
| 5.4 | Summary of various performance measures for simple regression models..... | 125 |
| 5.5 | Values of various performance measures for multiple regression model..... | 127 |
| 5.6 | Summary of various performance measures for polynomial and multiple regression..... | 128 |
| 6.1 | Training Data-Deviation in entropy with actual number of zombies..... | 138 |
| 6.2 | Testing Data-Deviation in entropy with actual number of zombies..... | 138 |
| 6.3 | Training results of various feed forward networks..... | 139 |
| 6.4 | Test results of various feed forward networks..... | 140 |
| 6.5 | Summary of test error for feed forward neural network for network size 5-1 | 143 |
| 6.6 | Summary of test error for feed forward neural network for network size 10-1 | 143 |
| 6.7 | Summary of test error for feed forward neural network for network size 15-1 | 143 |

| | | |
|-----|---|-----|
| 6.8 | Comparison between Regression and ANN based schemes for predicting number of zombies in a DDoS attack..... | 144 |
| 7.1 | Simulation parameters..... | 149 |
| 7.2 | Deviation in entropy with actual strength of DDoS attack..... | 152 |
| 7.3 | Deviation in volume and flow with DDoS attack strength..... | 153 |
| 7.4 | Summary of residual error for various regression models..... | 160 |
| 7.5 | Summary of various performance measures for simple regression models..... | 161 |
| 7.6 | Summary of residual error for multiple regression model..... | 163 |
| 7.7 | Summary of various performance measures for polynomial and multiple regression model..... | 164 |
| A.1 | Distribution of TCP, UDP and ICMP connections in normal connections in (a) training dataset, (b) testing dataset..... | 172 |
| A.2 | Distribution of TCP, UDP and ICMP connections in attack connections in (a) training dataset, (b) testing dataset..... | 173 |
| A.3 | Distribution of TCP, UDP and ICMP connections in DoS attack connections in (a) training dataset, (b) testing dataset..... | 173 |
| A.4 | Attacks Distribution in (a) training dataset, (b) testing dataset..... | 175 |

CHAPTER 1

INTRODUCTION

In the past two decades, Internet has revolutionized almost every facet of our lives. Government, commercial, and educational organizations depend on Internet to such an extent that day-to-day operations are significantly hindered when the network is “down”. Almost all the important services such as banking, transportation, stock trade, medicine, education, etc are extended to Internet now. Everything is available on a click of a mouse. But unfortunately at the same time, the prosperity of the Internet also attracts abusers and malicious attackers. Internet based attacks can be launched anywhere in the world, and unfortunately no Internet based services are immune to these attacks. These attacks lead to heavy financial losses, delays, and customer dissatisfaction. Trustworthiness and security of the Internet not only benefits on-line businesses, but it is also an issue for national safety. Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks are currently amongst the most problematic Internet security threats. These attacks are critical as they aim at denying or degrading services for a legitimate user. This chapter gives a brief introduction of the DDoS attacks, motivation for DDoS defense, problem statement and finally organization of the thesis.

1.1 Introduction

A revolution came into the world of computer and communication with the advent of Internet. Therefore, Internet has become increasingly important to the current society. It has changed our way of communication, business mode, and even everyday life [29]. The impact of Internet on society can be seen from figure 1.1 which shows exponential increase in number of hosts interconnected through Internet [198]. As, we can see from figure 1.1, there were only around 1 million Internet host in January 1993, which has increased to more than 775 million Internet hosts in October 2010. Though, it is not easy to manage few millions of Internet hosts, it is very difficult to manage 775 millions Internet hosts. Poorly managed machines tend to be easy to compromise. Figure 1.2 shows the size of the Internet users in the world by various geographic regions. This is the recent information according to the survey of

Mini Watts Marketing Group [95]. According to this survey, the estimated Internet users are 1,966,514,816 for June 30, 2010.

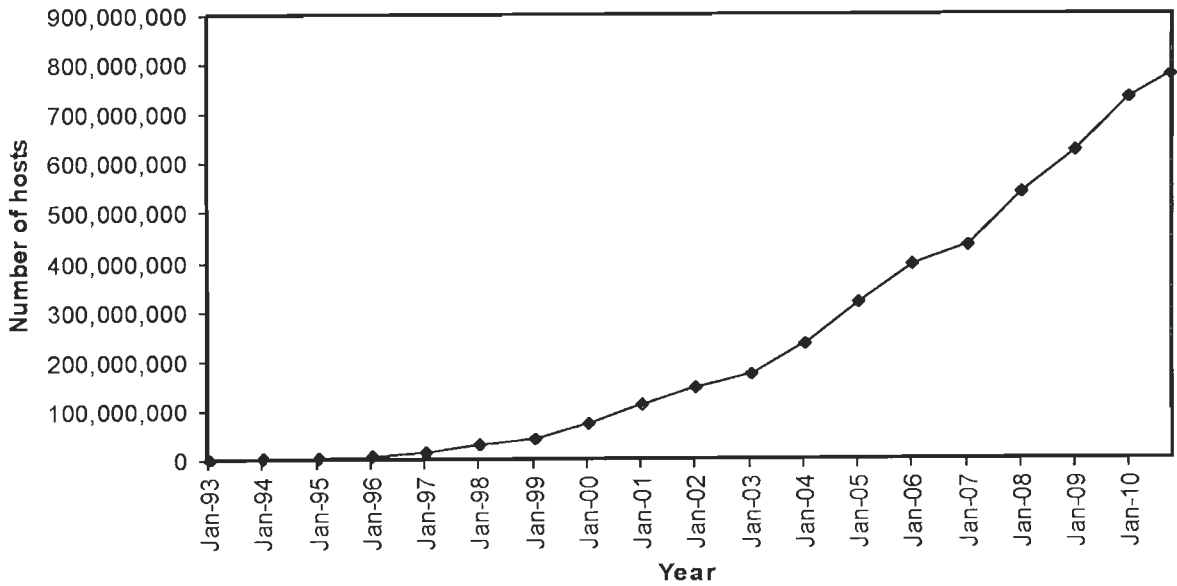


Figure 1.1. Internet Domain Survey Host Count

Internet users in the World by Geographic Regions-2010

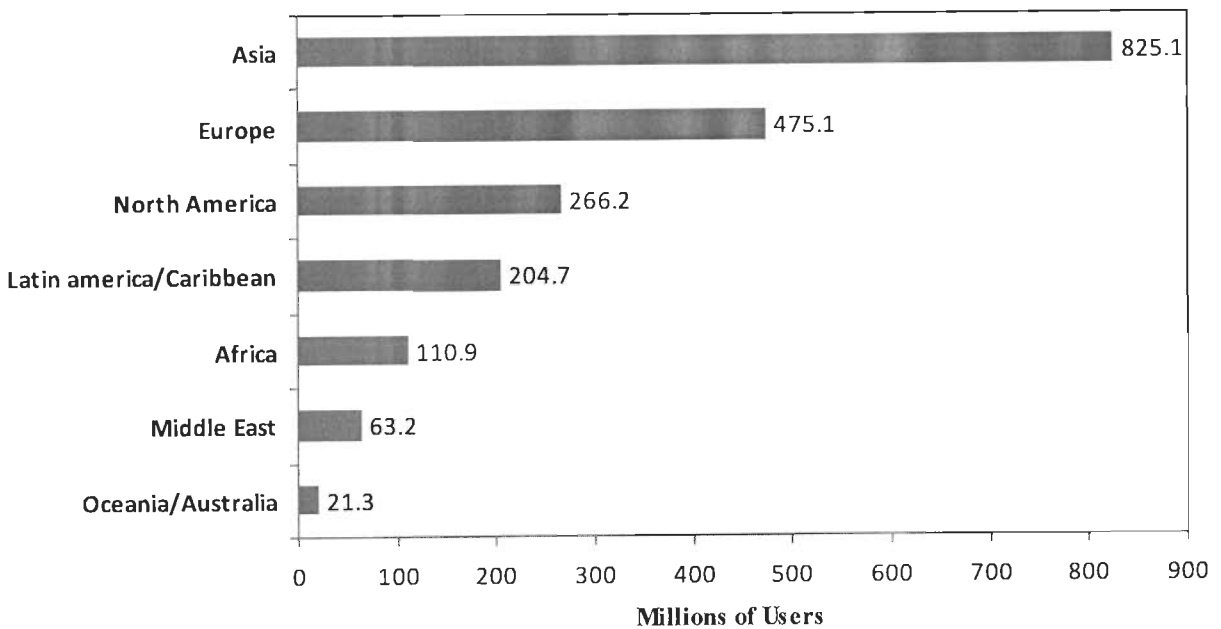


Figure 1.2. Estimated Internet Users in the World by Geographic regions according the survey of Mini Watts Marketing Group

Internet usage is growing at an exponential rate as organizations, governments and citizens continue to increase their reliance on this technology. Unfortunately with an increase in number of host, count of attacks on Internet has also increased incredibly fast. According to [49], a mere 171 vulnerabilities were reported in 1995 which boomed to 7236 in 2007. Vulnerabilities for third quarter of 2008 have gone up to 6058 as shown in figure 1.3. Computer emergency response team (CERT) stopped updating its website after 2008 due to increasing number of vulnerabilities reported every year. Apart from these, a large number of vulnerabilities go unreported every year. In particular, denial-of-service (DoS) attack is one of the most common and major threat to the Internet today. It reveals big loopholes not only in specific applications, but also in the entire TCP/IP protocol suite.

A DoS attack can be described as an attack designed to render a computer or network incapable of providing normal services. It is considered to take place only when access to a computer or network resource is intentionally blocked or degraded as a result of malicious action taken by another user [37]. Therefore, as defined by Weiler, it includes any of the following attempts [144]:

- to inhibit legitimate network traffic by flooding the network with useless traffic,
- to deny access to a service by disrupting connections between two parties,
- to block the access of a particular individual to a service, or
- to disrupt the specific system or service itself.

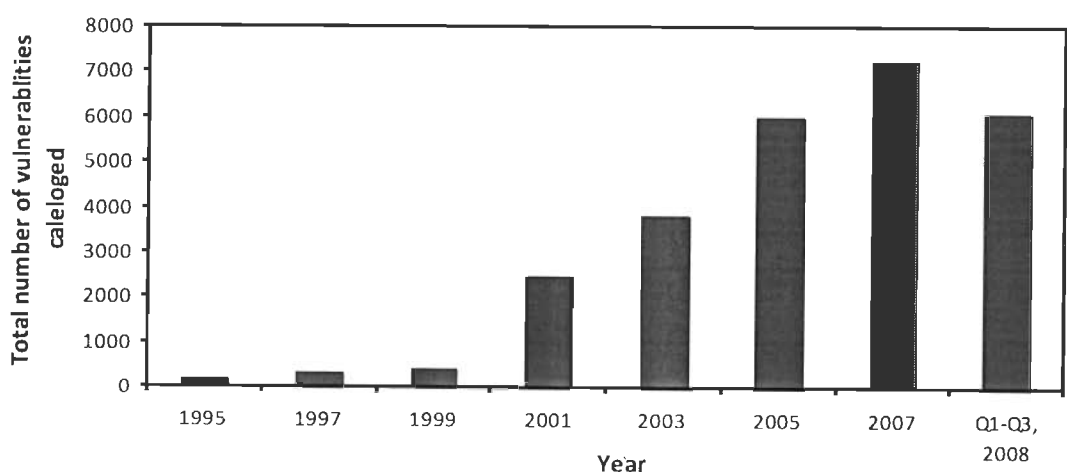


Figure 1.3. Vulnerabilities reported since 1995

The main aim of such attacks is to prevent the victim either from the benefit of a particular service (in case of client being victim) or from providing its services to others (in case of server being victim). A DDoS attacker uses many machines to launch a coordinated attack against one or more targets [38]. This attack is launched indirectly through many compromised computing systems by sending a stream of useless aggregate traffic meant to explode victim resources. As a side effect, these attacks frequently create network congestion on the way from the source to the target, thus disrupting normal Internet operations. The number of DDoS attack has been alarmingly increasing for the last few years [65]. Many of today’s DDoS attacks are carried out by organized criminals targeting financial institutions, e-commerce, gambling sites, etc.

Usually, DDoS attack can be launched in two forms [108, 109]. The first form is to exploit software vulnerabilities of a target by sending malformed packets and crash the system. The second form is to use massive volume of legitimate looking but garbled packets to clog up computational or communication resources on the target machine so that it can not serve its legitimate users. The resources consumed by attacks include network bandwidth, disk space, CPU time, data structures, network connections, etc [127]. While, it is possible to protect the first form of attack by patching known vulnerabilities, the second form of attack can not be so easily prevented. The targets can be attacked simply because they are connected to the public Internet.

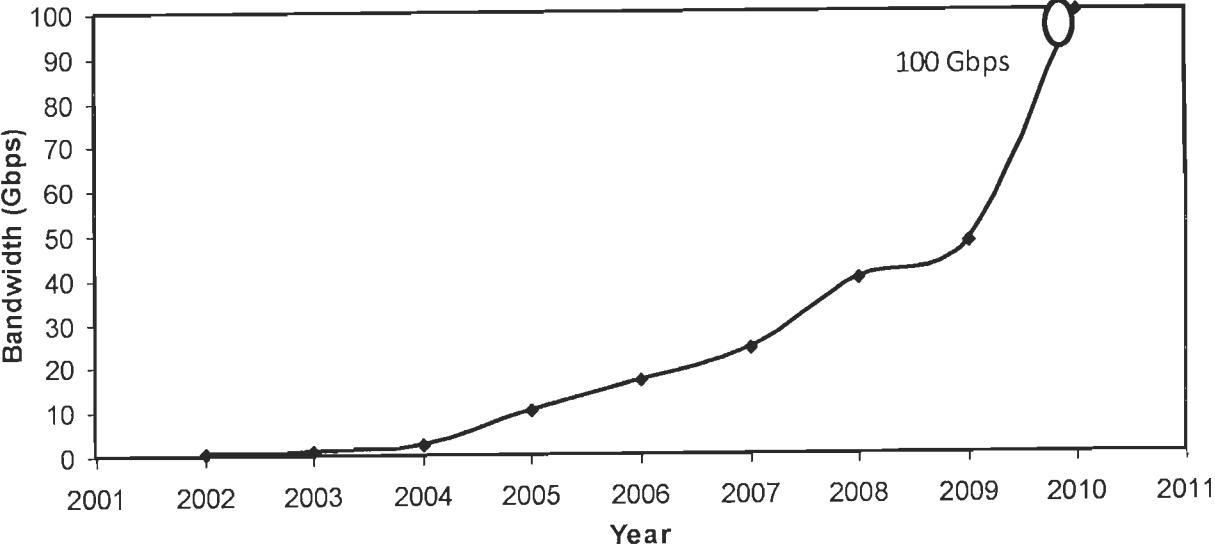


Figure1.4. Increase in DDoS attack traffic

Recent trends in the Internet [72, 164] show that the total amount of the DDoS attacks reached over 100 gigabit per second barrier. It also shows that the amount of DDoS attack traffic has been increasing in size year by year. A study conducted by Arbor Networks [164] shows the year by year increase of the DDoS attack traffic on the Internet, from the year 2001 to 2010 as shown in Figure 1.4.

It is immoral to start a DDoS attack on the Internet even if the purpose of it is research to defensive measures for the future attacks. Also, the results of a DDoS attack staged for research purposes on the Internet could be unpredictable due to the Internet complexity. Thus, for the research purposes one must consider either computer/simulation network models or analysis of various statistical data of real DDoS attacks that happened in the past. In this thesis, we focus our study on network models and use computer programs to simulate the DDoS attack on network models and statistical data of real DDoS attack that happened in the past for the validation. This thesis presents several techniques for defending against DDoS attacks, and evaluates their effectiveness against a wide range of DDoS attacks.

1.2 Motivation

DDoS attacks quickly became increasingly popular as communities of crackers developed and released extremely sophisticated, user friendly and automated toolkits [28, 36, 57, 58, 59, 60, 99, 136, 171] to carry them out. At present, even people with little knowledge can use them to carry out DDoS attacks. The impact of DoS attacks can vary from minor inconvenience to users of a website, to serious financial losses for companies that rely on their on-line availability to do business. Additionally, the problem is aggravated because of huge base of insecure machines on the Internet.

Various DDoS attacks against some high-profile websites such as Yahoo, Amazon, CNN news and E*Trade were performed in early 2000 [3, 77, 211], which caused them to go offline for several hours. In some cases these attacks have produced very high attack traffic (e.g. in Gbits/s) against a single victim. Series of attacks on GRC.Com in May, 2001 [175], a highly coordinated attack against CERT in May, 2001 [19], series of attacks against ISP's in UK in 2002 [103], distributed attack against name servers in Akamai's Content Distribution Network (CDN) in June 15, 2004 [21, 39], and the text-to-speech translation application running in the Sun Microsystem's Grid Computing system disabled with a DDoS attack in

March, 2006 [148] demonstrate how devastating DDoS attacks are and how defenseless the Internet is under such attacks. As a proof of these disturbing trends, a computer crime and security survey conducted by FBI/CSI in the United States for the year 2004 [125] on 251 organizations, DoS attack is the second most widely detected outsider attack type in computer networks immediately after virus infections. A computer crime and security survey conducted in Australia for the year 2004 [26] shows similar results. A study showed that the number of DDoS attacks increased by 50% per year [97], and the attacks also increased in sophistication and severity. The losses caused by DDoS attacks are remarkable particularly to e-commerce sites. According to Jupiter Communications, 46% of consumers report that the poor site performance drove them away from their preferred sites. Unacceptable download times often caused by DDoS attacks have been estimated to cause losses of up to \$4.35 billion in United States e-commerce sales and worldwide businesses experienced about 3.3% of unplanned downtime in 1999, translating to \$1.6 trillion in lost revenue [52, 186].

A DNS reflection attack on Register.com, which is first major attack involving DNS servers occurred in January 2001 [89]. This attack, which forged requests for the MX records of AOL.com lasted about a week before it could be traced back to all attacking hosts and shut off. It used a large list of DNS servers at least a year old at the time of the attack. Moore et. al. [65] used the backscatter analysis to assess the number, duration, and focus of DoS attacks in the Internet. Backscatter is called the unsolicited response traffic which the victim sends in response to attack packets with spoofed IP source addresses. The results indicate more than 12,000 attacks against more than 5,000 distinct victims during the 3-week period examined in February, 2001.

The DNS root servers are constant targets for DDoS attacks. On two occasions to date, attackers have performed DNS backbone DDoS attacks on the one or more of the thirteen DNS root servers. These attacks are extremely significant, because the root name servers translate text-based Internet hostnames into IP addresses and function as the Internet backbone. Therefore, these two DDoS attacks might be classified as attempts to take down the entire Internet, rather than specific websites. The first occurred in October 2002, lasted for approximately one hour and disrupted service at 9 of the 13 root servers [87]. The second occurred in February 2007, lasted for approximately five hour and caused disruptions at two of the root servers [91]. The botnet responsible for the attack has reportedly been traced to the Asia-Pacific region.

An increased attack on financial institutions and other organizations that keep financial records, auctions, e-commerce and gambling sites are observed and they are blackmailed before major events are due. For example, in August 2005 the Hamburg-based gambling site jaxx.de was blackmailed to pay 40,000 Euros to stop an ongoing DDoS attack [54, 113].

In February 2007, more than 10,000 online game servers that were hosting games such as Return to Castle Wolfenstein, Halo, Counter-Strike, and many others were attacked by 'RUS' hacker group. The Distributed denial of service attack was made from more than a thousand computer units located across the former republics of the erstwhile Soviet Union [90]. In September 2008, DDoS attacks on Sa-Mp servers started and became a very huge problem. All the servers in the official list were being attacked by DDoS attack by a hacker called Ryan Cleary/Savage [90]. On March 4, 2009 the home page of Game Rating Board was attacked by DDoS, lasted for approximately five days [88]. The incident was the first case for the public agency to be inflicted by a serious damage from DDoS.

Above examples of DDoS attacks show that the main motives behind these attacks are criminal, commercial or ideological nature. There exist few reasons, which make DDoS attacks inevitable: the Internet is designed to keep intermediate network as simple as possible to optimize it for packet forwarding [129]. This pushes the complexity to the end hosts and causes one unfortunate implication. If one party in two-way communication misbehaves, it can result in arbitrary damage to its peer. No one in the intermediate network will step in and stop it, because Internet is not designed to police traffic. Moreover, the Internet security is highly interdependent. Even though, we can use some traditional security mechanisms like firewall [140, 161], Intrusion detection system (IDS) [33, 70], access list [39], etc. to protect victim machine, its susceptibility to DDoS attacks also depends on the position of security in rest of the global Internet. For example, if an attacker is able to exploit an insecure legitimate machine which is authorized to communicate with the victim, that machine can be used to perform attack against the victim, as incoming attack traffic to the victim seems to be normal traffic. The limited availability of resources acts as an additional benefit for DDoS attackers. To add on, accountability is not enforced which leads to variety of reflector attacks [74, 204]. One of the most dangerous types of reflector attack that is very difficult to deal with, is Smurf attack [42, 53]. Thus, there exists no way out to enforce global deployment of a particular security mechanism [108].

Existing DDoS defense mechanisms are classified into four broad categories: prevention, detection, response, and tolerance & mitigation. *Attack prevention* methods try to stop all well known signature based and broadcast based DDoS attacks from being launched at the beginning of attack or at the edge routers and keep all the machines over Internet up to date with patches and fix security holes. This approach aims to improve the global security level and is the best solution to DDoS attacks in theory. However, the disadvantage is that it needs global cooperation to ensure its effectiveness, which is extremely difficult in reality. Attack prevention schemes are also vulnerable to novel and mixed attack types for which signatures and patches do not exist in the database. Therefore, these are considered forensic defense methods and the challenge is how to develop a scalable mechanism, which can be effective for preventing variety of attack types with low implementation cost.

Attack detection aims to detect an ongoing attack as soon as possible without misclassifying and disrupting legitimate traffic. Typical detection techniques fall into three categories: signature based attack detection, anomaly based attack detection and hybrid attack detection. The challenge in attack detection is how to detect variety of attack types quickly without misclassifying any legitimate traffic.

The goal of the *attack response* is to relieve the impact of the attack on the victim while imposing minimal collateral damage to legitimate clients. Typical response techniques fall into four categories: attack source/path identification, filtering, rate throttling and reconfiguration. The challenge for attack response is how to filter and rate throttle the attack traffic without disturbing legitimate traffic. Identification of attack sources/paths quickly and accurately without changing current Internet infrastructure at minimum space and time complexity is also an immense challenge.

Attack tolerance & mitigation is the final step in defending against DDoS attacks, and therefore focuses on minimizing the attack impact and tries to provide optimal level of service as per quality of its service requirement to legitimate users while service provider is under attack. The challenge for attack tolerance & mitigation is how to provide optimal level of services to legitimate users when attack is performed at very high rate.

Researchers have come up with more and more specific solutions to the DDoS problem. However, existing DDoS attack tools keep being improved and new attack techniques are developed. Therefore, cycle of attacking and defending is like a game. When someone finds a way to attack a system, someone else tries to defend against this attack. The

attacker then tries harder to defeat the protections. It becomes a cycle that never seems to end. This motivates us to investigate efficient solutions to current and future DDoS attacks rather than to react with specific countermeasures. Hence, research in this thesis focuses on developing robust and effective solutions to counteract variety of DDoS attacks.

1.3 Statement of the Problem

“The objective of this research is to develop efficient mechanisms to detect variety of DDoS attacks quickly and accurately to ensure reasonable performance of the network or system under attack”. To achieve the above objective, problem has been divided into following sub problems:

- To detect a variety of DDoS attacks using flow-volume based approach (FVBA).
- To detect flooding DDoS attacks using GARCH model based approach.
- To predict number zombies involved in a DDoS attack using various regression based schemes.
- To predict number zombies involved in a DDoS attack using ANN based scheme.
- To estimate strength of a DDoS attack using regression analysis.

1.4 Organization of the Thesis

This thesis comprises of eight chapters including this chapter that introduces the topic and states the problem. The rest of the thesis is organized as follows.

Chapter 2 gives an overview of the DoS and DDoS attacks and classifies them according to the various criteria. Also, a comprehensive study of a wide range of DDoS attacks and defense methods proposed to combat them are discussed. It also discusses the related work and research gaps in the various phases of the defense against DDoS attacks.

Chapter 3 presents a DDoS detection scheme named as FVBA. The flow and volume based approach (FVBA) utilizes number of flows and volume of incoming traffic to detect a wide range of DDoS attacks. The approach is scalable and can adapt to varying network conditions and attack loads using varying tolerance factors. To test the performance of the proposed approach, extensive simulations have been performed using NS-2 network simulator. The simulation results are promising and show the supremacy of the proposed approach over existing volume based approaches. Furthermore, KDD 99 dataset has also been utilized to validate the proposed approach.

Chapter 4 presents our efforts in applying Generalized Autoregressive Conditional Heteroskedastic (GARCH) model, which is a commonly used statistical modeling technique for financial time series, as a new technique for detecting DDoS attacks. The chapter gives a brief overview of need for nonlinear time series analysis in modeling DDoS attack and describes the heteroscedasticity property of DDoS attack traffic. The main part of the chapter focuses on the development of the model and algorithm for detection of DDoS attack. MATLAB routines were used for testing heteroskedastic nature of traffic and simulation of detection algorithm. In the last part of the chapter, the comparison between FVBA approach of chapter 3 with the nonlinear GARCH based approach is discussed. Results show that this non linear volatility model not only performs better than earlier models like linear prediction but also it shows slightly better performance than FVBA approach.

Chapter 5 presents use of various regression methods to predict number of zombies involved in a DDoS attack. Accurate prediction of number of zombies involved in a DDoS attack is very important to suppress the effect of attack by choosing predicted number of most suspicious attack sources for either filtering or rate limiting. To contribute to such an accurate and easier solution, we established a relationship between number of zombies and observed deviation in sample traffic using various regression models. A comparative study is performed among different regression models for effectiveness in predicting number of zombies. The simulation results are promising as we are able to predict number of zombies efficiently using various regression models.

Chapter 6 presents a more effective approach for predicting number of zombies presented in chapter 5. The chapter mainly deals with the use of ANN for the problem at hand. The underlying assumption for the use of Artificial Neural Network (ANN) in predicting number of zombies is its nonlinear estimation capacity. Hence network traffic data is used to train and test various sizes of feed forward networks and they are compared for their estimation performance. The generalization capacity of the trained network is promising and the network is able to predict number of zombies involved in a DDoS attack efficiently.

Chapter 7 presents suitability of various regression models to estimate strength of a DDoS attack. Estimating strength of attack is helpful to suppress the effect of attack, as it enables a security administrator to effectively equip his arsenal with proper defense mechanisms for fighting against DDoS threat according to the strength of attack. Regression

models are very much suitable for this and hence we established relationship between strength of attack and sample traffic to formulate regression equations. The regressions equations obtained using curve fitting are tested for their estimation performance and promising results are obtained.

Chapter 8 concludes our research work and gives directions for future work.

CHAPTER 2

BACKGROUND AND LITREATURE SURVEY

Today, denial of service (DoS) attacks and particularly the distributed ones (DDoS) are one of the latest threat and pose a grave danger to users, organizations and infrastructures of the Internet. In these attacks, goal of the attacker is to tie up chosen key resources at the victim, usually by sending a high volume of seemingly legitimate traffic requesting some services from the victim. The first publicly reported DDoS attack appeared in the late 1999 against a university [127]. These attacks quickly became increasingly popular as communities of crackers developed and released automated tools to carry them out. This made attack by even inexperienced crackers possible. Thus, these attacks are easiest to implement from an attacker's point of view and definitely one of the costliest from a business point of view. The main purpose of this chapter is to give an overview of DDoS attacks, its basic causes, targeted resources, its modus operandi and available DDoS attack tools. In addition, it presents a comprehensive study of a wide range of DDoS attacks and defense methods proposed to combat them. This provides better understanding of the problem, current solution space and future research scope to defend against DDoS attacks.

2.1 Background and Overview

Security is quite an old concern in the field of computer technology. In early 1950's, computers incorporated mechanisms to ensure that programs could not use someone else's memory. During 1960's, several security techniques such as protecting passwords by encryption or controlling access to files, whose principles are still in use today, were developed. The raising trend led computer security to be studied as a full discipline during the beginning of the 1970's. Since then, new security issues have appeared as fast as the old ones were solved. As research teams were developing new defense mechanisms, the underground attack field was also maturing and producing more and more sophisticated tools, raising new problems. The advent of the Internet particularly gave a boost to the importance of computer security. While often used as a business media, the Internet is a highly non-secure, non-trustworthy environment from a security point of view [109].

2.1.1 Denial of Service Attacks

DoS attack is a computer or network security issue which can affect the availability of computer or network services by degrading or disrupting their resources. It is commonly characterized as an event in which a legitimate user or organization is deprived of certain services, like web, email or network connectivity, that they would normally expect to have [108].

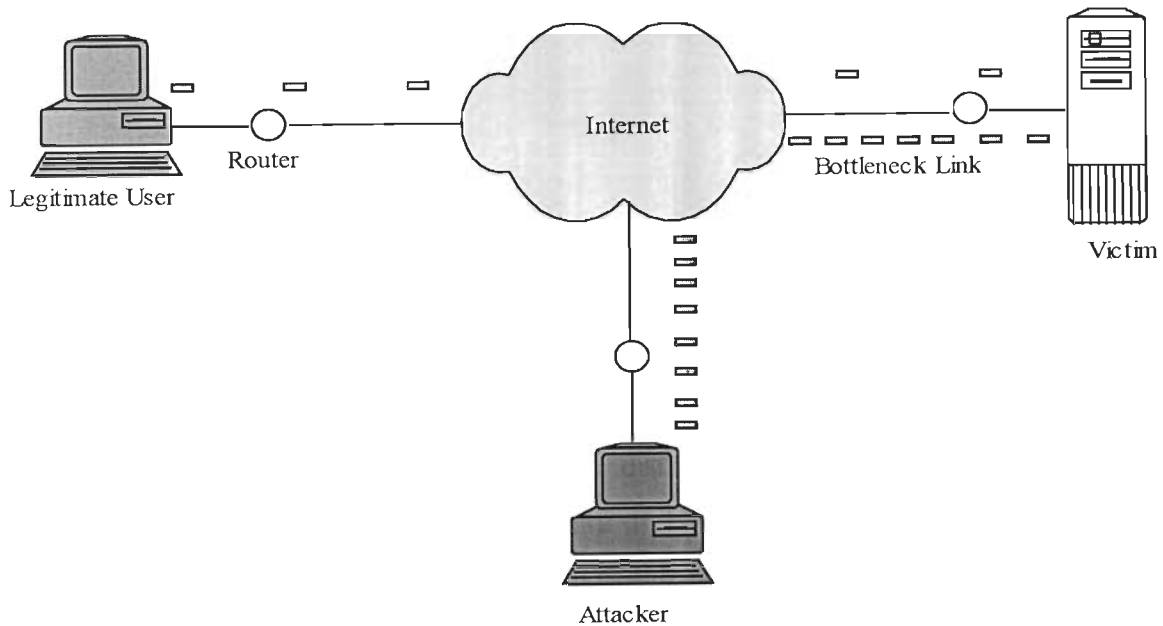


Figure 2.1. Denial of Service attack scenario

A DoS attack is considered to take place only when access to a computer or network resource is intentionally blocked or degraded as a result of malicious action taken by another user [37]. A wide range of DoS attacks exist and can target individual user, group of users, organizations and infrastructures of the Internet. In denial of service attacks, only one machine is used by attacker to perform attack. Figure 2.1 depicts a typical denial-of-service attack scenario in which an attacker sends a stream of malicious packets to a victim, denying its service to legitimate user.

2.1.2 Distributed Denial of Service Attacks

Today, the most common attack type is the distributed denial-of-service (DDoS). It can usually cause more significant damage than DoS attack by performing attack from many

zombie machines. A distributed approach makes the attack prevention more difficult. Figure 2.2 depicts a simple distributed denial-of-service attack scenario in which attacking machines A1, A2, A3 send streams of malicious packets to victim, denying its services to legitimate user. A DDoS attack has two phases: a deployment and an attack phase [109]. A DDoS program must first be deployed on one or more compromised hosts before an attack is possible. Mitigation of DDoS attacks thus requires defense mechanisms for both phases [108].

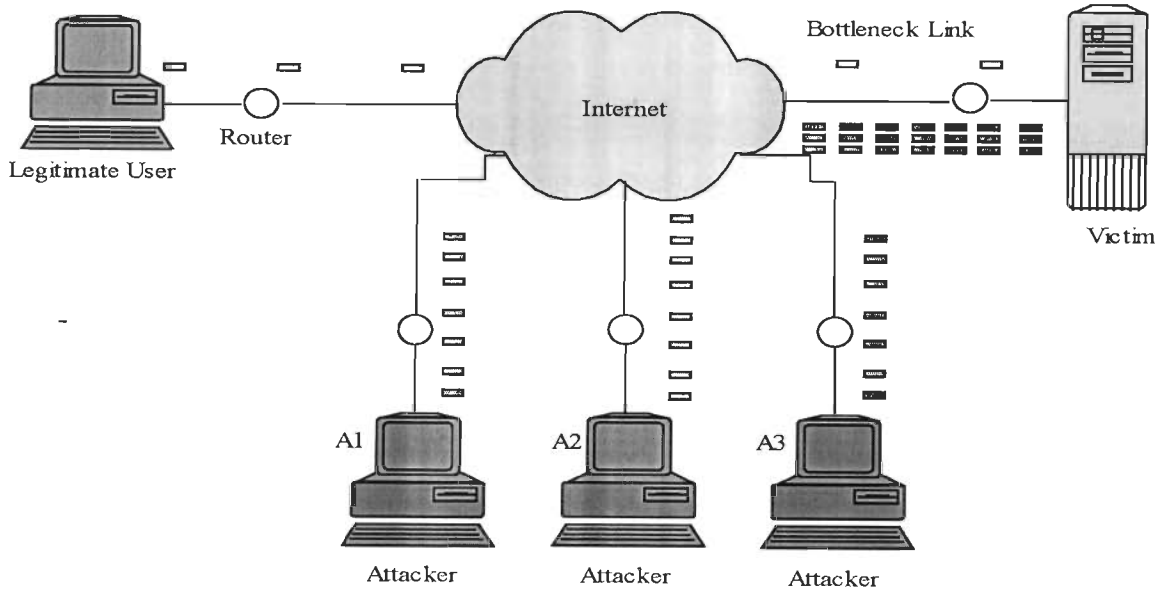


Figure 2.2. Distributed Denial of Service attack scenario

The distributed DDoS attack model provides the attacker with the following advantages:

Attack Effect: A well-coordinated attack that originates from multiple locations will have a devastating effect on the target. Since the attack effect is directly proportional to the number of zombie machines, the distributed denial of service attack model typically delivers the desired results from an attacker's point of view.

Anonymity: Since the actual attack is originated from multiple locations, the distributed model provides the attacker a high ground for covering its tracks.

Hard-to-stop Attacks: The distributed attacks are sometimes referred as hard-to-stop Attack. The level of anonymity involved and the dimensions at which the attack is carried out makes the distributed attack very difficult to stop without bringing down or disconnecting the target system from the network.

2.1.3 Distributed Reflector Denial of Service (DRDoS) Attacks

In the DRDoS attacks [204], attacker does not use zombie machines to flood the victim directly, but uses third-party (e.g. router, web servers, DNS server etc.) called reflectors. Reflector attacks make distributed denial of service attacks more difficult to defend because of IP spoofing. IP spoofing refers to forging of source address fields in the packet. Source address spoofing hides location of an attacker which means packets are sent with a false source IP address. Attacker spoofs requests containing the address of the victim to a large set of Internet servers that will in turn send their combined replies to the victim. The combination with the legitimate stream complicates the victim's abilities both to isolate the attack traffic in order to block it, and to use traceback techniques for locating the source. A DRDoS attack is more detrimental than a typical DDoS attack as it creates a larger volume of attack traffic.

2.2 Major Causes Responsible for DDoS Attacks

Internet's design goal is functionality, not security [78]. The network fabric tries to provide fast, simple and cheap communication at the network level. More complicated functionalities are assigned to end hosts. Under such an end-to-end paradigm and the so called best efforts principle, end users are allowed to manage their communication as they wish, and add complexities while leaving the intermediate network fabric simple and efficient. However, such design opens several security issues that provide opportunities for the attackers. Following section summarizes the major causes responsible for DDoS attacks to occur [108]:

i). Dependency and Lack of centralized control: Internet security is highly interdependent and has lack of central control. Placing all networks and users under the same control is infeasible due to its anarchic culture. Hence, it is impossible to guarantee all end hosts to have the latest security software installed and suitable policies applied. Secondly, DDoS attacks are commonly launched from systems that are subverted by an intruder via a security-related compromise rather than from the intruder's own system or systems. Thus, no matter how well secured the victim system may be, its susceptibility depends on the state of security of rest of the global environment. It is easy for attackers to hide their identities from tracing back in different networks [48].

ii). Each individual Internet host owns limited resources: Each Internet host only has limited and consumable resources such as bandwidth, processing power, buffer size, memory size, etc that attracts variety of attacks. When the amount of requests is beyond the capability of the host, newer requests will not receive the services.

iii). More the number of zombie machines, more power attacker has: As long as there are enough vulnerable machines on the Internet, an attacker is always able to recruit enough zombies. Considering that the botnet consists of millions of machines [100], theoretically speaking, it is capable of overpowering any end host in the Internet (if there is no defense).

iv). Dummy intermediate network fabric: The intermediate network fabric will do nothing to stop attacking traffic flows. On the contrary, the network passively forwards packets to the destination under best effort principle. In fact, the network helps attackers to reach their goals.

v). Accountability is not enforced: The Internet is not equipped with any mechanism for making accountability against forged IP addresses in packet header. This creates the opportunity for source address spoofing. Setting the victim's address in the source field makes it look like the packet is originally send from the victim. Setting an unused IP address in the source field forces the victim to try to contact with a non-existing host.

2.3 Targeted Resources by Distributed Denial of Service Attack

Although, DoS attacking strategies may differ, studies show that attackers mainly target the following resources to cause damage on victims [30, 35, 108].

- **Network bandwidth resources:** This is related with the capacity of the network links connecting servers to the wider Internet or connectivity between the clients and their Internet Service Providers (ISP). The traffic that comes from the Internet to the client may consume the entire bandwidth of the client's network. As a result, a legitimate request will not be able to get service from the targeted network. In a DoS attack, the vast majority of traffic directed at the target network is malicious; generated either directly or indirectly by an attacker. These attacks prevented 13,000 Bank of America ATMs from providing

withdraw services and paralyzed such large ISPs as Freetel, SK Telecom, and KoreaTelecom on January 25, 2003.

- **System memory resources:** An attack targeting system memory resources typically aims to crash its network handling software rather than consuming bandwidth with large volume of traffic. Specific packets are sent to confuse the operating system or other resources of the victim's machine. These include temporary buffer used to store arriving packets, tables of open connections, and similar memory data structures. Another system resource attack uses packets whose structures trigger a bug in the network software, overloading the target machine or disabling its communication mechanism, or making a host crash, freeze or reboot which means the system can no longer communicate over the network until the software is reloaded.
- **System CPU resources:** An attack targeting system CPU resources typically aims to employ a sequence of queries to execute complex commands and then overwhelm the CPU. The Internet key Exchange protocol (IKE) is the current IETF standard for key establishment and secure association (SA) parameter negotiation of IPSec. However, IKE's aggregate mode is still very susceptible to DoS attacks against both computational and memory resources because the server has to create states for SA and compute Diffie-Hellman exponential generation [167].

2.4 DDoS Attack: Modus Operandi

Here, we describe a typical DDoS attack scenario, its core elements and strategy. A DDoS attack is composed of four elements [76]: attack source, control masters, agents, and victim.

- **Attack source:** Attack source is machine, handled by attacker who is the mastermind behind the attack. It is the one who sets every plan about the attack.
- **Control masters:** Control masters coordinate and control multiple agents and exploit further agent machines on behalf of attack source. Control masters are deployed on the hosts on the Internet.

- **Agents:** Agents, also known as slaves or attack daemons, are programs that actually conduct the attack on the victim. Attack daemons are usually deployed on host computers. These daemons affect both the target and the host computers. The task of deploying these attack daemons requires the attacker to gain access and infiltrate the host computers.
- **Victim:** A victim is a target host that has been chosen to receive the impact of the attack.

Figure 2.3 shows how these elements are coordinated to inflict DDoS attack on a targeted victim machine. DDoS attack is carried out from multiple sources to aim at a single target, in several phases [48, 75,108]:

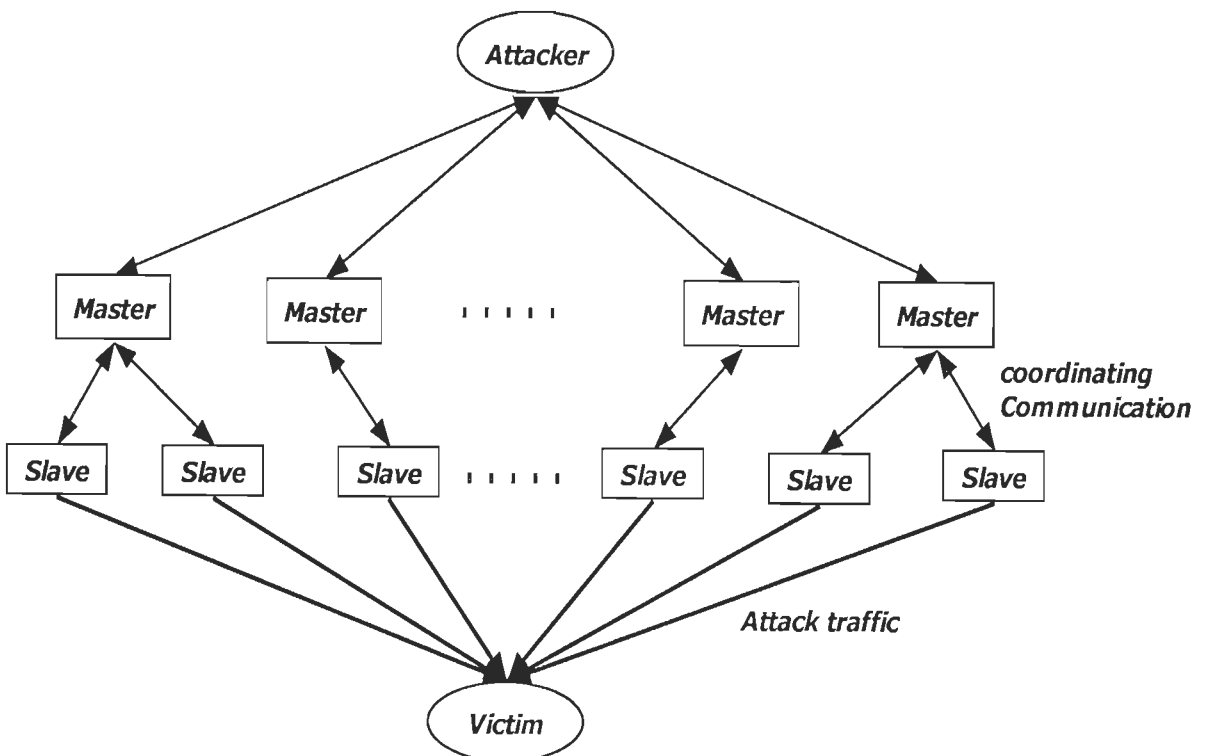


Figure 2.3. A hierarchical model of a DDoS attack

i). Recruiting phase:

In order to launch a DDoS attack, the attacker first scans millions of machines for vulnerable services and other weakness on the Internet through high-bandwidth, always-on connections that permit penetrations. The scanning can be manual or automatic. The attacker

uses different host address scanning strategies such as Random, Hitlist, Signpost, Permutation, or local subnet scanning techniques [66, 108, 143].

ii). Exploiting phase:

In this phase, the discovered vulnerabilities are then exploited to gain access on these machines, known as handlers, or masters.

iii). Infecting phase:

Here, the attacker installs malicious scripts or Trojans that can be used to create back doors for future communication with the host. After being installed the malicious scripts, these infected machines can repeat the same procedure to recruit more machines, known as zombies or slaves. These all exploited machines used as attack army, are collectively called bots and the attack network is known as botnet in the hacker's community. During infecting phase, the attacker may use different attack propagation mechanisms like central source propagation, back-chaining propagation, autonomous propagation [48, 108], etc.

iv). Attack phase:

Once the attacker or control master (by delegation) setup the botnet with the type of attack, the actual flood may be initiated manually or automatically to the victim by the attack source. The real attacker sends a command to the masters to initiate a coordinated attack. When the masters receive the command, they transfer it to the slaves under their control. Upon receiving attack commands, the zombies or slaves begin the attack on the victim [76]. The real attacker tries to hide himself from detection, for example, by providing spoofed IP addresses [216].

2.5 DDoS Attack Tools

One of the major reason that make the DDoS attacks wide spread and easy to implement in the Internet is the availability of attacking tools and the powerfulness of these tools to generate attacking traffic. There are a variety of different DDoS attack tools on the Internet that allow attackers to execute attacks on the target system. Some of the most common tools are discussed below:

- *Trinoo* [57, 108] can be used to launch a coordinated UDP flooding attack against target system. Trinoo uses master/slave architecture and attacker controls a number of Trinoo master machines. Communication between attacker and master and between master and slave is performed through TCP and UDP protocol, respectively. Both master and slaves are password protected to prevent them from being taken over by another attacker. *Wintrinoo* is a Windows version of trinoo that was first reported to CERT on February 16, 2000.
- *TFN* [59] uses a command line interface to communicate between the attack source and the control master program. Communication between the control masters and slaves is done via ICMP echo reply packets. But, it does not offer any kind of encryption between attack source and masters or between masters and slaves. It can implement Smurf, SYN flood, UDP flood, and ICMP flood attacks. Detailed discussion about the various types of attack i.e. Smurf, SYN flood, UDP flood, TCP RST, TCP ACK and ICMP flood attack is given in the next section.
- *TFN2K* [38, 45, 99] is a more advanced version of the primitive TFN network. It uses TCP, UDP, ICMP or all three to communicate between the control master program and the slave machines. TFN2K can implement Smurf, SYN flood, UDP, and ICMP flood attacks. Communication between the real attacker and control master is encrypted using a key-based CAST-256 algorithm. In addition to flooding, TFN2K can also perform some vulnerability attacks by sending malformed or invalid packets.
- *Stacheldraht* [58] combines best features of both Trinoo and TFN. It also has the ability to perform updates on the slave machines automatically. It uses an encrypted TCP connection for communication between the attacker and master control program. Communication between the master control program and attack daemons is conducted using TCP and ICMP. Stacheldraht can implement Smurf, SYN flood, UDP flood, and ICMP flood attacks.

- *Shaft* [171] has been modeled on Trinoo network. Other than the port numbers being used for communication purpose, working of it is very similar to that of Trinoo. Thus, distinctive feature of Shaft is the ability to switch control master servers and ports in real time, hence making detection by intrusion detection tools difficult. Communication between the control masters and slave machines is achieved using UDP packets. The control masters and the attacker communicate via a simple TCP telnet connection. Shaft can implement UDP, ICMP, and TCP flooding attack.

Table 2.1. Summery of DDoS attack Tools

| DDoS attack tool | Encrypted/Unencrypted communication | Types of Attacks Generated | Communication Protocols |
|-------------------------|--|---|---|
| Trinoo | Not encrypted | UDP flooding | Attacker to handler- TCP Handler to agent- UDP Agent to handler - UDP |
| TFN | Numeric code and not encrypted | ICMP flooding TCP flooding UDP flooding SMURF | Attacker to handler-required third-party program Handler to agent- ICMP Agent to handler - none |
| TFN2K | Encrypted | ICMP flooding TCP flooding UDP flooding SMURF Mix flood | Handler to agent- can be mixture of TCP, UDP and ICMP Agent to handler – none |
| Stacheldraht | Encrypted | ICMP flooding TCP flooding UDP flooding SMURF | Attacker to handler- TCP Handler to agent- UDP Agent to handler – none |
| Shaft | Not encrypted | ICMP flooding TCP flooding UDP flooding Mix flood | Attacker to handler- TCP Handler to agent- TCP or ICMP Agent to handler – UDP |
| Mstream | Not encrypted | TCP flooding | Attacker to handler- TCP Handler to agent- UDP Agent to handler - UDP |
| Knight | Not encrypted | TCP flooding UDP Flood attacks an urgent pointer flooder | Uses IRC as its communication method |
| Trinity | Not encrypted | TCP flooding UDP flooding | Uses IRC as its communication method |

- *Mstream* [60] is more primitive than any of the other DDoS tools. It attacks target machine with a TCP ACK flood. Communication is not encrypted and is performed through TCP and UDP packets and the master connects via telnet to zombies. Masters can be controlled remotely by one or more attackers using a password protected interactive login. Source addresses in attack packets are spoofed at random. Unlike other DDoS attack tools, here, masters are informed of access, successful or not, by competing parties.
- *Knight* [36, 46] uses Internet relay chat (IRC) as a control channel. It has been reported that this tool is commonly installed on machines that were previously compromised by the BackOrifice Trojan horse program. Knight can implement SYN attacks, UDP flood attacks, and an urgent pointer flooder [36]. It is designed to run on Windows operating systems and has features such as an automatic updater via http or ftp, a checksum generator and more.
- *Trinity* [28, 136] is also IRC based DDoS attack tool. It can implement UDP flood, TCP SYN, TCP RST, TCP ACK, and other flooding attacks. Each trinity compromised machine joins a specified IRC channel and waits for commands. Use of legitimate IRC service for communication between attacker and agents eliminates the need for a master machine and elevates the level of the threat [38].

Source code of these attack tools can be easily downloaded from the Internet. Even though these attack tools differ in the commands used, types of attacks performed, communication techniques, and the presence of backdoors or self-upgrade capability, all share the common objective of attempting to overwhelm a victim with an abundant amount of traffic that is difficult to detect or filter. Table 2.1 shows a summary of different attack tools.

2.6 Classification of Attack Mechanisms

Here, a classification of a wide range of DDoS attacks, that users and Internet service providers need to be aware of, is presented. The classification is illustrated in figure 2.4 and is described below in detail:

2.6.1 Based on Attacking Methods

2.6.1.1 Flooding

Currently, most of the DDoS attacks performed are flooding type. In flooding DDoS attack, also known as brute force attack [108], legitimate looking but garbled packets are sent to victim machine to clog up computational or communication resources on the target machine so that it can not serve its legitimates users. The resources consumed by attacks include network bandwidth, disk space, CPU time, buffers, data structures, etc. The flood packets can be any of the following protocol types: TCP, UDP, ICMP or other protocol. Here, attackers need large number of compromised machines which can generate sufficient volume of traffic that can overload the victim's resources. In addition to compromised machines, there must be a control master that can synchronize the attacks.

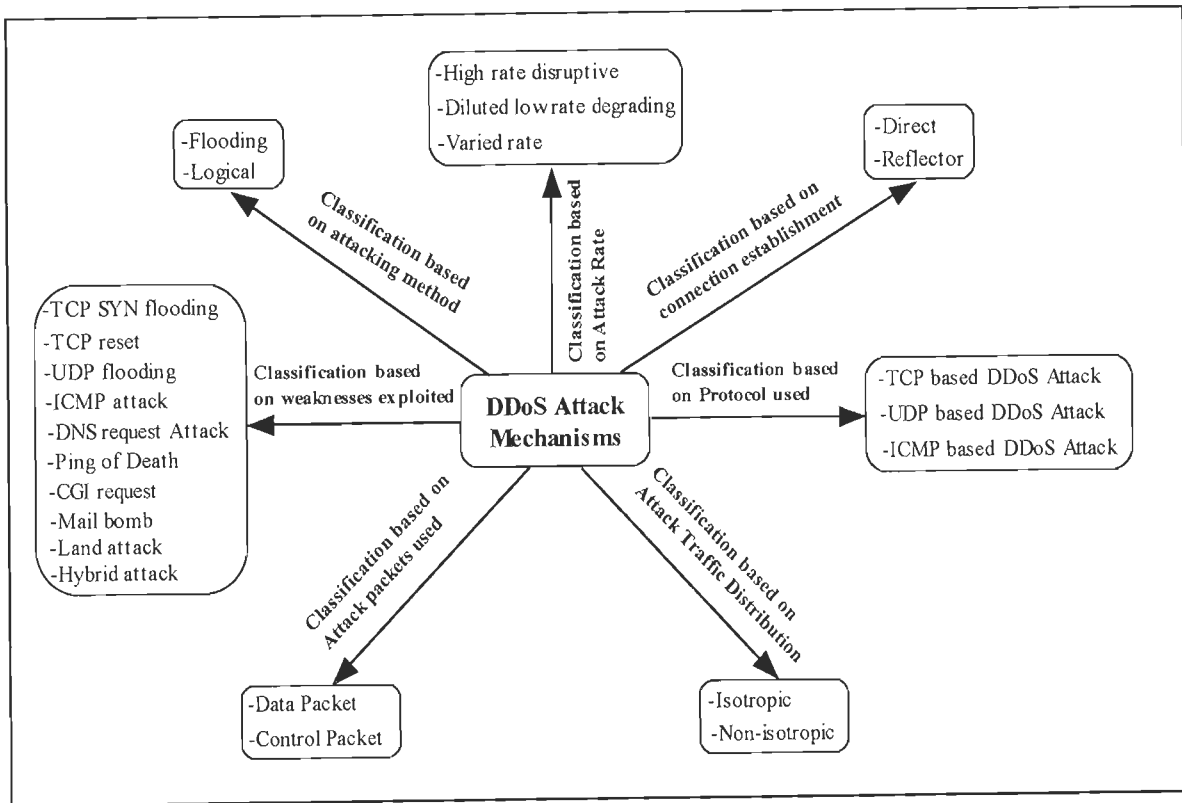


Figure 2.4. Classification of DDoS attack Mechanisms

2.6.1.2 Logical

Logical attacks exploit a specific feature or implementation bug of some protocol or application installed at the target machine in order to consume excess amount of its resources [109]. For example, in the TCP SYN attack, the exploited feature is the allocation of substantial space in a connection queue immediately upon receipt of a TCP SYN request. The attacker initiates multiple connections that are never completed, thus filling up the connection queue.

The attacker designs abnormal or bogus packets that can confuse the victim's system by exploiting the "natural" weakness of a system. Such weakness can be in protocol design, in operating system, in specific application, or in service of the victim's system. It is sufficient to use few packets (not necessarily a flood) to cause damage on a victim machine.

2.6.2 Based on Weaknesses Exploited

2.6.2.1 TCP SYN flooding

Any system providing TCP-based network services is potentially subject to this attack. In normal case, TCP 3-way handshaking is performed as shown in figure 2.5 (a). First the client sends a SYN request to the server. After receiving such request, server replies with a packet, which contains both the acknowledgement ACK and the synchronization request SYN (denoted as ACK/SYN). Then the client sends ACK back to establish the connection. The attacker sends a flood of TCP/SYN packets, often with a forged sender address. Each of these packets are handled like a connection request, causing the server to spawn a half-open connection, by sending back a TCP/SYN-ACK packet, and waiting for a TCP/ACK packet in response from the sender address.

However, because the sender address is forged, the response never comes. These half-open connections consume resources on the server and limit the number of connections the server is able to make, reducing the server's ability to respond to legitimate requests until after the attack ends. The result would be system crash or system inoperative. As shown in figure 2.5 (b) an attacker B initiates a SYN flooding attack by sending many connection requests with spoofed source addresses to the victim machine D. That causes D to allocate resources and, once the limit of half-open connections is reached, it refuses all successive connection establishment attempts [44, 157, 50].

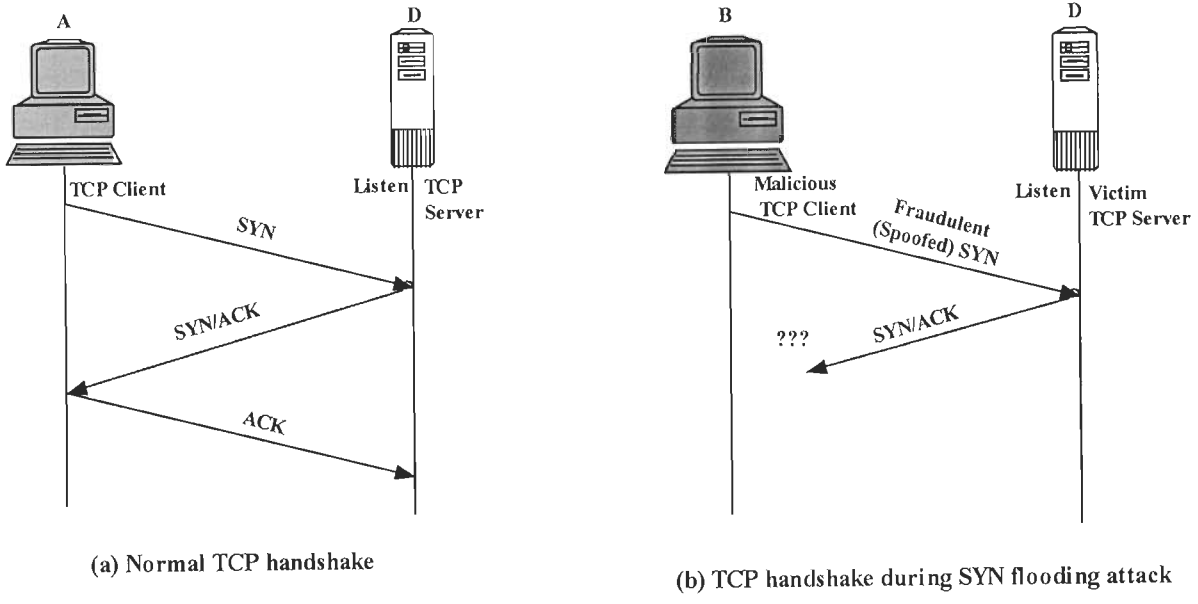


Figure 2.5. (a) TCP 3-way handshaking (b) TCP SYN attack

2.6.2.2 TCP reset

TCP reset also exploit the characteristics of TCP protocol. The main idea behind a TCP reset attack is to falsely terminate an established TCP connection without the consent of the two parties which own the endpoints [153, 196]. Let's imagine an established TCP connection from host A to host D. Now, a third host, B, spoofs a packet that matches the source port and IP address of host A, the destination port and IP address of host D, and the current sequence number of the active TCP connection between host A and host D. Then, host B sets the RST bit on the spoofed packet and when this packet is received by host D, host D immediately terminates the connection. This results in a denial of service, until the connection is reestablished.

2.6.2.3 UDP flooding

A UDP flood attack is possible when an attacker sends a UDP packet to a random port on the victim system. When the victim system receives a UDP packet, it will determine what application is waiting on the destination port. When it realizes that there is no application waiting on the port, it will generate an ICMP packet of destination unreachable to the forged

source address. If enough UDP packets are delivered to ports on victim, the system will go down.

This type of attack, most commonly exploits the chargen or echo services, creating an infinite loop between two UDP services [114, 153].

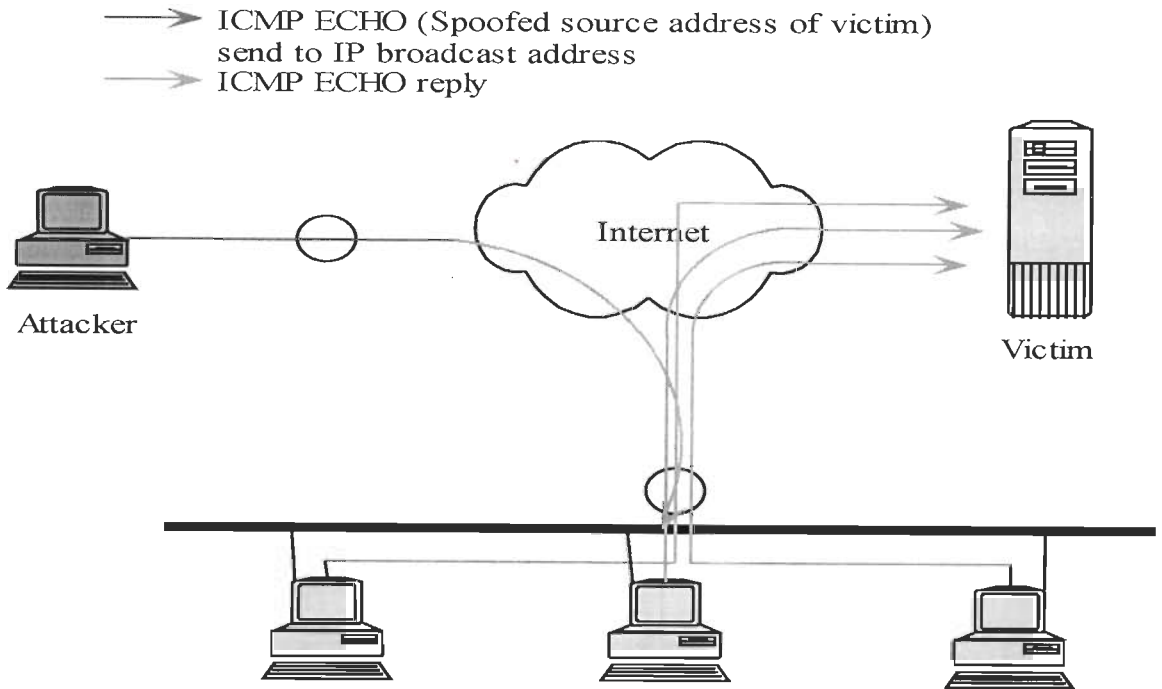


Figure 2.6. Smurf attack

2.6.2.4 ICMP attack

Smurf attack, as shown in figure 2.6, is ICMP flooding attack. The attacker directs a stream of ICMP ECHO requests to broadcast addresses in intermediary networks, spoofing the victim's IP address in their source address fields. A multitude of machines then reply to the victim, overwhelming its network [42, 53, 153].

2.6.2.5 DNS request Attack

In this attack scenario, the attacker sends a large number of UDP-based DNS requests to a name server using a spoofed source IP address i.e. victim address. Then, the name server, acting as an intermediate party in the attack, responds by sending back replies to the victim. In a DNS request, attack small queries can generate larger UDP packets in response, which is

known as amplification effect of DNS response. Because of this amplification effect of DNS response, it can cause serious bandwidth attack [170]. For example, in the initial DNS specification, UDP packets were limited to 512 bytes. At most, a 60 byte query could generate a 512 byte response for an amplification factor of 8.5. This amplification effect has been used in DNS based attacks for some time [111].

2.6.2.6 Ping of Death

The Ping of Death is a typical TCP/IP implementation attack. In this assault, the DDoS attacker creates an IP packet that exceeds the IP standard's maximum 65,536 byte size. When this fat packet arrives, it crashes systems that are using a vulnerable TCP/IP stack. No modern operating system or stack is vulnerable to the simple Ping of Death attack [134].

2.6.2.7 CGI request

By simply sending multiple CGI request to the target server, the attacker consumes the CPU resource of the victim. Finally, the server is forced to terminate its services [51].

2.6.2.8 Mail bomb

A mail bomb is the sending of a massive amount of e-mails to a specific system. A huge amount of mails may simply fill up the recipient's disk space on the server or, in some cases, may be too much for a server to handle and may cause the server to stop functioning. This attack is also a kind of flood attack [47].

2.6.2.9 Land Attacks

A Land attack is similar to a SYN attack, the only difference being that, instead of a bad IP Address, the IP address of the target system itself is used. What this means is that, in a land attack, the attacker sends SYN packets to a particular port of the target system with the source address and source port number of these SYN packets, being same as the destination IP address and port number. This creates an infinite loop between the target system and the target system itself and hangs or crashes it.

2.6.2.10 Teardrop attack

Teardrop attack exploits the vulnerability present in the reassembling of data packets. It involves sending invalid or garbage IP fragments with overlapping, over-sized, payloads to the target machine. A bug in the TCP/IP fragmentation re-assembly code of various operating systems causes the fragments to be improperly handled and forced them to crash, hang or reboot [197].

2.6.2.11 Hybrid attack

With the large number of countermeasures being employed by a number of organizations on the Internet, recently there has been an emergence of hybrid forms of DDoS attacks. In such attacks, the attacker combines two or more attack types to form a hybrid variety of DDoS attack. Example: teardrop spoofing attack, overlapping land attack etc.

Teardrop spoofing attack involves spoofed mangled IP fragments with overlapping, over-sized, payloads to the target machine to crash, hang or reboot it. Similarly, overlapping land attack involves mangled IP fragments with overlapping, over-sized payloads and with the source address and source port number of these mangled IP fragments, being same as the destination IP Address and port number to the target machine to crash, hang or reboot it.

2.6.3 Based on Connection Establishment

2.6.3.1 Direct

In this case, zombies send huge amount of packets directly targeting victim machine. To serve this purpose, attackers often compromise and gain control over thousands or even millions of vulnerable machines. The attacking packets are routed to the victim from zombies distributed widely on the Internet.

2.6.3.2 Reflector

It is more complicated and harder to trace back compared to direct attacks. Instead of sending packets to victims directly, the zombies take advantage of the TCP three-way handshake mechanism. Zombies are instructed to continuously send TCP connection-requesting SYN packets to other innocent IP hosts. Those SYN packets carry a spoofed source IP belonging to the victim. As the second phase of the TCP connection handshake,

these innocent hosts reply to the victim with SYN/ACK packets according to the source IP address in the requesting packets they received. In this manner, malicious SYN packets are being “reflected” off innocent nodes and their SYN/ACK responses are being used to flood and attack the victim [204].

2.6.4 Based on Attack Rate

2.6.4.1 High rate disruptive

In high rate disruptive attacks, sheer volume of packets at very high rate are sent from distributed locations in a coordinated manner to completely disrupt the availability of Internet services. As these attacks have direct impact on ISP networks, they are easy to detect and characterize.

2.6.4.2 Diluted low rate degrading

In diluted low rate degrading attacks, packets are sent from a large number of infected machines i.e. zombie machines, at low rate in a coordinated manner to gracefully degrade network performance. As these attacks degrade Quality of Service (QoS) of the network slowly, thus they are very difficult to detect and characterize.

2.6.4.3 Varied rate

To make detection of attacks more difficult, attackers can use some sophisticated attack tools to generate varied rate attacks in which they use some of the zombie machines to generate packets at high rate while remaining machines to generate packets at low rate.

2.6.5 Based on Attack Traffic Distribution

In order to defeat aggregate based defense, attackers try to distribute attack traffic uniformly throughout all ingress points of attacked autonomous system. This is called isotropic distribution of attack traffic whereas if attack traffic is aggregated more in certain parts of Internet, then it is called non-isotropic distribution of attack traffic [120, 170].

2.6.6 Based on Attack Packets Used

Logical DDoS attacks are normally launched with control packets like TCP SYN, TCP FIN, ICMP echo packets whereas for launching flooding DDoS attacks control as well as data packets like HTTP, FTP (involving TCP), UDP, and ICMP bogus packets can be used [120].

2.6.7 Based on Protocol Used

Network protocols based classification of DDoS attacks basically divides DDoS attacks into TCP, UDP and ICMP protocol based attacks, as either of these protocol's packets can be used for flooding and logical attacks [120].

2.7 Defense Challenges and Principles

Launching DDoS attacks on the victim machine is only a matter of few keystrokes for the attacker. The victim can prevent these attacks at its network boundary by configuring some sort of traditional security tools like access list [166], firewall [140, 161], or intrusion detection system [33, 70] at its end. But the traffic coming from legitimate user, which is under control of the attacker, looks normal and can not be detected using these traditional methods.

With the present technology, many challenges are involved in designing and implementing an effective DDoS defense mechanism. Some of them are as follows [120]:

- (a) Large number of unwitting participants,
- (b) No common characteristics of DDoS streams,
- (c) Use of legitimate traffic models by attackers,
- (d) No administrative domain cooperation,
- (e) Automated tools,
- (f) Hidden identity of participants,
- (g) Persistent security holes on the Internet,
- (h) Lack of attack information and
- (i) Absence of standardized evaluation and testing approaches.

Following five principles are recommended by robinson et al. [137] in order to build an effective solution:

- Since, DDoS is a distributed attack and because of high volume and rate of attack packets, distributed instead of centralized defense is the first principle of DDoS defense.
- Secondly, High Normal Packet Survival Ratio (NPSR) hence less collateral damage is the prime requirement for a DDoS defense.
- Third, a DDoS defense method should provide secure communication for control messages in terms of confidentiality, authentication of sources, integrity and freshness of exchanged messages between defense nodes.
- Fourth, a partially and incrementally deployable defense model is successful as there is no centralized control for autonomous systems (AS) in Internet.
- Fifth, a defense system must take into account future compatibility issues such as interfacing with other systems and negotiating different defense policies.

Similarly, Tupakula et. al. [199] presented following characteristics that an ideal effective model against DDoS attacks should have:

- It should be invoked only during the attack times and at other times, it must allow the system to work normally. So, it should readily integrate with existing architecture with minimum modifications.
- It must provide simple, easy and effective solution to counteract the attacking sources in preventing the attack.
- It should identify the attack at the victim and prevent the attack near to the attacking source.
- It should prevent only the attack traffic from reaching victim. That is, the model should be able to differentiate a malicious traffic flow from a regular benign flow by incorporating different attack signatures for different attacking sources.
- It should have fast response time and should respond quickly to any changes in attack traffic pattern.
- It should provide mechanisms for retaining the attack evidence for any future legal proceedings.

2.8 Classification of DDoS Defense Mechanisms

Based on the above defense principles and existing DDoS attacks discussed in section 2.6, large numbers of defense methods have been proposed to combat DDoS attacks in the literature. Figure 2.7 summarizes a classification of various defense mechanisms used by researchers. Detailed description of the classification is as below:

2.8.1 Based on Activity Deployed:-

Classification based on activity deployed categorizes the DDoS defense mechanisms in the following four categories:

2.8.1.1 DDoS attack prevention

Attack prevention methods try to stop all well known signature based and broadcast based DDoS attacks from being launched at edge routers, keeps all the machines over Internet up to date with patches and fix security holes. Attack prevention schemes are not enough to stop DDoS attacks, because they are always vulnerable to novel and mixed attack types for which signatures and patches do not exist in the database. So, these are considered forensic defense methods.

Techniques for preventing against DDoS can be broadly divided into two categories: (i) General techniques, which are some common preventive measures [213] i.e. system protection, replication of resources etc. that individual servers and ISPs should follow so that they do not become part of DDoS attack process. (ii) Filtering techniques, which include ingress filtering, egress filtering, router based packet filtering, history based IP filtering, SAVE protocol, etc.

A. General Techniques

i). Disabling unused services:

The less there are applications and open ports in hosts, less there are chances to exploit vulnerabilities by attackers. Therefore, if network services are not needed or are unused, the services should be disabled to prevent attacks, e.g. UDP echo, character generation services [213].

ii). Install latest security patches

Today, many DDoS attacks exploit vulnerabilities in target system. So removing known security holes by installing all relevant latest security patches prevents re-exploitation of vulnerabilities in the target system [213].

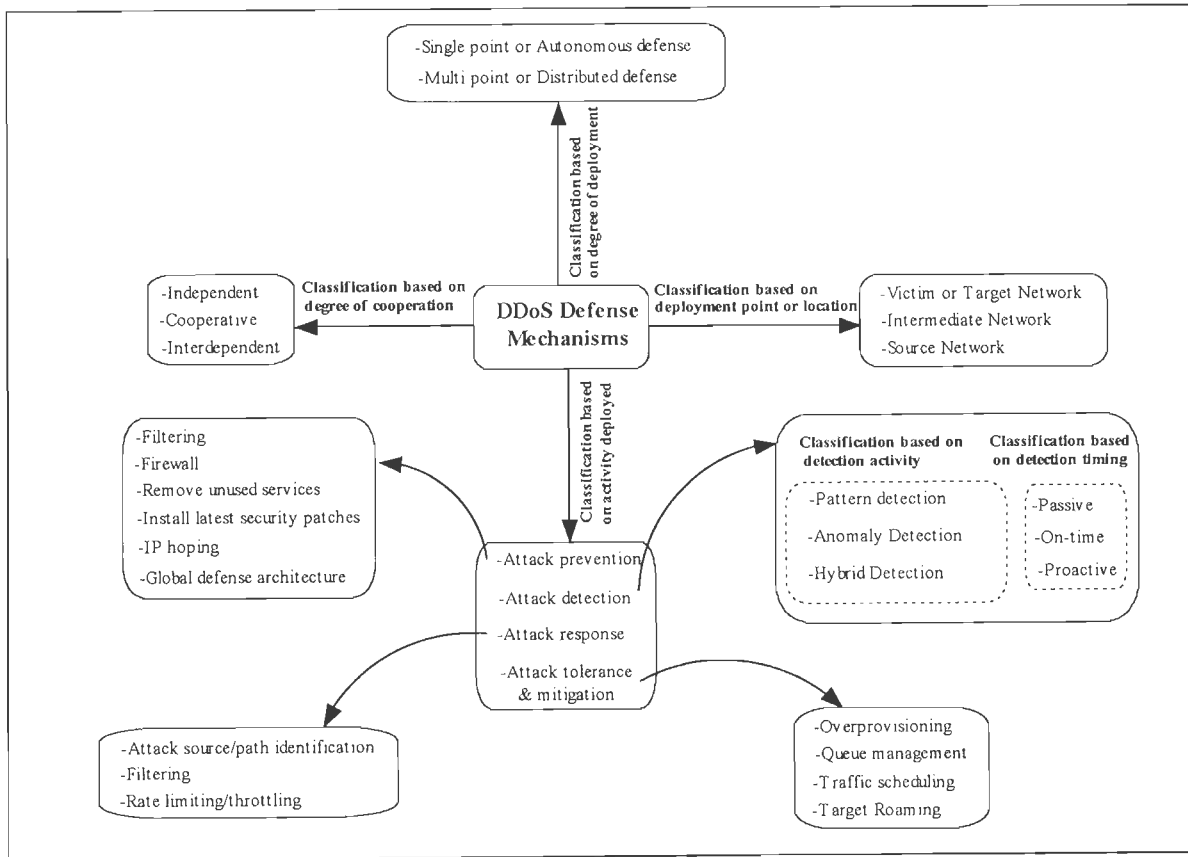


Figure 2.7. Classification of DDoS Defense Mechanisms

iii). Disabling IP broadcast

Defense against attacks that use intermediate broadcasting nodes e.g. ICMP flood attacks, Smurf attacks etc. will be successful only if host computers and all the neighboring networks disable IP broadcast [76].

iv). Firewalls

Firewalls can effectively prevent users from launching simple flooding type attacks from machines behind the firewall. Firewalls have simple rules such as to allow or deny

protocols, ports or IP addresses. But, some complex attack e.g. if there is an attack on port 80 (web service), firewalls can not prevent that attack because they can not distinguish good traffic from DoS attack traffic [140, 161].

v). Global defense infrastructure

A global deployable defense infrastructure can prevent from many DDoS attacks by installing filtering rules in the most important routers of the Internet. As Internet is administered by various autonomous systems according to their own local security policies, such type of global defense architecture is possible only in theory [213].

vi). IP hopping

DDoS attacks can be prevented by changing location or IP address of the active server proactively within a pool of homogeneous servers or within a pre-specified set of IP address ranges [213]. The victim computer's IP address is invalidated by changing it with a new one. Once the IP address change is completed all internet routers will be informed and edge routers will drop the attacking packets. Although, this action leaves the computer vulnerable because the attacker can launch the attack at the new IP address, this option is practical for DDoS attacks that are based on IP addresses. On the other hand, attackers can make this technique useless by adding a domain name service tracing function to the DDoS attack tools.

B. Filtering Techniques

i). Ingress/Egress filtering

Ingress Filtering, proposed by Ferguson et al. [147], is a restrictive mechanism to drop traffic with IP addresses that do not match a domain prefix connected to the ingress router. Egress filter is an outbound filter, which ensures that only assigned or allocated IP address space leaves the network. A key requirement for ingress or egress filtering is knowledge of the expected IP addresses at a particular port. For some networks with complicated topologies, it is not easy to obtain this knowledge.

One technique known as reverse path filtering [34] can help to build this knowledge. This technique works as follows. Generally, a router always knows which networks are reachable via any of its interfaces. By looking up source addresses of the incoming traffic, it

is possible to check whether the return path to that address would flow out of the same interface as the packet arrived upon. If they do, these packets are allowed. Otherwise, they are dropped.

Unfortunately, this technique can not operate effectively in real networks, where asymmetric Internet routes are not uncommon. More importantly, both ingress and egress filtering can be applied not only to IP addresses, but also to protocol type, port number, or any other criteria of importance. Both ingress and egress filtering provide some opportunities to throttle the attack power of DoS attacks. However, it is difficult to deploy ingress/egress filtering universally. If the attacker carefully chooses a network without ingress/egress filtering to launch a spoofed DoS attack, the attack can go undetected. Moreover, if an attacker spoofs IP addresses from within the subnet, the attack can go undetected as well. Nowadays DDoS attacks do not need to use source address spoofing to be effective. By exploiting a large number of compromised hosts, attackers do not need to use spoofing to take advantage of protocol vulnerabilities or to hide their locations. For example, each legitimate HTTP Web page requests from 10,000 compromised hosts can bypass any ingress/egress filtering, but in combination they can constitute a powerful attack. Hence, ingress and egress filtering are ineffective to stop DDoS attacks.

ii). Router based Packet Filtering (RPF)

Route based filtering, proposed by Park and Lee [122], extends ingress filtering and uses the route information to filter out spoofed IP packets. It is based on the principle that for each link in the core of the Internet, there is only a limited set of source addresses from which traffic on the link could have originated.

If an unexpected source address appears in an IP packet on a link, then it is assumed that the source address has been spoofed, and hence the packet can be filtered. RPF uses information about the BGP routing topology to filter traffic with spoofed source addresses. Simulation results show that a significant fraction of spoofed IP addresses can be filtered if RPF is implemented in at least 18% of Autonomous systems (ASs) in the Internet. However, there are several limitations of this scheme. The first limitation relates to the implementation of RPF in practice. Given that the Internet contains more than 10,000 ASs, RPF would need to be implemented in at least 1800 ASs in order to be effective, which is an onerous task to accomplish. The second limitation is that RPF may drop legitimate packets if there has

recently been a route change. The third potential limitation is that RPF relies on valid BGP messages to configure the filter. If an attacker can hijack a BGP session and disseminate bogus BGP messages, then it is possible to mislead border routers to update filtering rules in favor of the attacker. RPF is effective against randomly spoofed DoS attacks. However, the filtering granularity of RPF is low. This means that the attack traffic can still bypass the RPF filters by carefully choosing the range of IP addresses to spoof. Hence, RPF is ineffective against DDoS attacks. The router-based packet filtering is vulnerable to asymmetrical and dynamic Internet routing as it does not provide a scheme to update the routing information.

iii). History based IP filtering

Generally, the set of source IP addresses that is seen during normal operation tends to remain stable. In contrast, during DoS attacks, most of the source IP addresses have not been seen before. Peng et al. [193] relies on the above idea and use IP address database (IAD) to keep frequent source IP addresses. During an attack, if the source address of a packet is not in IAD, the packet is dropped. Hash based/Bloom filter techniques are used for fast searching of IP in IAD. This scheme is robust, and does not need the cooperation of the whole Internet community [193].

However, history based packet filtering scheme is ineffective when the attacks come from real IP addresses. In addition, it requires an offline database to keep track of IP addresses. Therefore, cost of storage and information sharing is very high.

iv). Capability based method

Capability based mechanisms provides destination a way to control the traffic directed towards itself. In this approach, source first sends request packets to its destination. Router marks (pre-capabilities) are added to request packet while passing through the router. The destination may or may not grant permission to the source to send packets. If permission is granted then destination returns the capabilities, if not then it does not supply the capabilities in the returned packet. The data packets carrying the capabilities are then send to the destination via router. The main advantage achieved in this architecture is that the destination can now control the traffic according to its own policy, thereby reducing the chances of DDoS attack, as packets without capabilities are treated as legacy and might get dropped at the router when congestion happens [187].

However, these systems offer strong protection for established network flows, but responsible to generate a new attack type known as DOC (Denial of Capability), which prevents new capability-setup packets from reaching the destination, limits the value of these systems. In addition, these systems have high computational complexity and space requirements.

v). Secure Overlay Service (SOS)

Secure overlay service proposed by Keromytis et al. [11] defines an architecture to secure the communication between the confirmed users and the victim. All the traffic from a source point is verified by a secure overlay access point (SOAP). Authenticated traffic will be routed to a special overlay node called a beacon in an anonymous manner by consistent hash mapping. The beacon then forwards traffic to another special overlay node called a secret servlet for further authentication, and the secret servlet forwards verified traffic to the victim. The identity of the secret servlet is revealed to the beacon via a secure protocol, and remains a secret to the attacker. Finally, only traffic forwarded by the secret servlet chosen by the victim can pass through its perimeter routers.

SOS addresses the problem of how to guarantee the communication between legitimate users and a victim during DoS attacks. SOS can greatly reduce the likelihood of a successful attack. The power of SOS is based on the number and distribution level of SOAPs. However, wide deployment of SOAPs is a challenge.

Moreover, the power of SOS is also based on the anonymous routing protocol within the overlay nodes. Unfortunately, the introduction of a new routing protocol is in itself another security issue. If an attacker is able to breach the security protection of some overlay node, then it can launch the attack from inside the overlay network. Moreover, if attackers can gain massive attack power, for example, via worm spread, all the SOAPs can be paralyzed, and the target's services will be disrupted.

vi). SAVE: Source Address Validity Enforcement

Li et al. [104] have proposed a new protocol called the source address validity enforcement (SAVE) protocol, which enables routers to update the information of expected source IP addresses on each link and block any IP packet with an unexpected source IP address. The aim of the SAVE protocol is to provide routers with information about the range

of source IP addresses that should be expected at each interface. Similarly to the existing routing protocols, SAVE constantly propagates messages containing valid source address information from the source location to all destinations. Hence, each router along the way is able to build an incoming table that associates each link of the router with a set of valid source address blocks. SAVE is a protocol that enables the router to filter packets with spoofed source addresses using incoming tables. It overcomes the asymmetries of Internet routing by updating the incoming tables on each router periodically.

However, SAVE needs to change the routing protocol, which will take a long time to accomplish. If SAVE is not universally deployed, attackers can always spoof the IP addresses within networks that do not implement SAVE. Moreover, even if SAVE were universally deployed, attackers could still launch DDoS attacks using non spoofed source addresses.

To conclude, attack prevention aims to solve IP spoofing, a fundamental weakness of the Internet. However, as attackers gain control of larger number of compromised computers, attackers can direct these “zombies” to attack using valid source addresses. Since the communication between attackers and “zombies” is encrypted, only “zombies” can be exposed instead of attackers. According to the Internet Architecture Working Group [131], the percentage of spoofed attacks is declining. Only four out of 1127 customer-impacting DDoS attacks on a large network used spoofed sources in 2004. Moreover, security awareness is still not enough, so expecting installation of security technologies and patches in large base of Internet seems to be an ambitious goal in near future. To add on, there exists no way out to enforce global deployment of a particular security mechanism. Therefore, relying on attack prevention schemes is not enough to stop DDoS attacks.

2.8.1.2 DDoS attack detection

To defend against DDoS attacks efficiently, a real-time detection of network anomalies is preferred. Attack detection aims to detect an ongoing attack as soon as possible without disrupting legitimate traffic. We may classify DDoS detection mechanisms using following different criteria:

A. Based on detection timing

Based on detection timing, DDoS detection approaches can be classified as follows:

i). Passive detection

Detection is passive if logs are analyzed after attacker fulfills its desire and attack is over.

ii). On-time detection

Detection is on time, if attack can be detected when attack is going on.

iii). Proactive detection

Detection is proactive, if attack can be detected either before it reaches target machine or before appreciable degradation of service.

B. Based on detection method

Based on detection method, DDoS detection approaches can be classified as follows:

i). Pattern based attack detection

Signature based approach employs a priori knowledge of attack signatures. The signatures are manually constructed by security experts after analyzing previous attacks and used to match with incoming traffic to detect intrusions. SNORT [138] and Bro [205] are the two widely used signature based detection approaches. Signature based techniques are only effective in detecting traffic of known DDoS attacks whereas new attacks or even slight variations of old attacks go unnoticed.

ii). Anomaly based attack detection

Anomaly-based system uses a different philosophy. It treats any network connection violating the normal profile of traffic as an anomaly. A network anomaly is revealed if the incoming traffic pattern deviates from the normal profiles significantly. Detecting DDoS attacks involves first knowing normal behavior of the system and then to find deviations from that behavior. Anomaly based techniques can detect novel attacks; however, it may result in higher false alarms. The common challenge for all anomaly-based intrusion detection systems is to accurately train data to provide all types of normal traffic behavior. As a result, legitimate traffic can be classified as attack traffic, causing false positive. To minimize the false positive rate, a larger number of parameters are used to provide more accurate normal

profiles. However, with increase in number of parameters, the computational overhead to detect the attack increases. Some of anomaly detection schemes proposed in the literature are discussed below:

Gil and Poletto [192] proposed a scheme called MULTOPS (*MUti-Level Tree for Online Packet Statistics*) to detect denial of service attacks by monitoring the packet rate in both the up and down links. MULTOPS assumes that packet rates between two hosts are proportional during normal operation. A significant, disproportional difference between the packet rate going to and from a host or subnet is a strong indication of a DoS attack. MULTOPS assumes that the incoming packet rate is proportional to the outgoing packet rate, which is not always the case. For example, real audio/video streams are highly disproportional, and with the widespread use of online movies and online news, where the packet rate from the server is much higher than from the client, false positive rates will become a serious concern for this scheme. Another countermeasure is to connect to the target from a large number of attack sources in a legitimate manner (e.g., downloading a file from a ftp server). Therefore, the packet rate ratio between in flows and out flows during the attack will appear to be normal and will be undetected by MULTOPS. Thus, this method is also ineffective when an attack is launched through multiple distributed sources or the source spoofing is used.

Normally, an attacker performs a DDoS attack using large number of similar packets (in terms of their destination address, protocol type, execution pattern etc.) generated from various locations but intended for the same destination. Thus, there is a lot of similarity in the traffic pattern. On the other hand, legitimate traffic flows tend to have many different traffic types. Hence, traffic flows are not highly correlated and appear to be random. Based on this assumption, Kulkarni et al. [7] proposed a Kolmogorov complexity based detection algorithm to identify attack traffic. The assumption of the Kolmogorov test is based on the fact that multiple attack sources use the same DDoS attack tool. Therefore, the resulted traffic is highly correlated.

Unfortunately, there is no theoretical analysis to support this assumption. Attack sources can be organized to break the correlation by sending attack traffic at different times, with different traffic types, packet sizes, and sending rates. This is easy to achieve. For example, attackers can use the IP address of a compromised computer as the random seed to

generate a set of parameters for configuring attack traffic. By doing this, attack traffic will appear random, which can bypass detection.

Based on the strong correlation between traffic behavior at the target and at the attack source, Cabrera et al. [116] proposed a scheme to proactively detect DDoS attacks using time series analysis. There are three steps in to this scheme. The first step is to extract the key variables from the target. For example, the number of ICMP echo packets is the key variable for Ping Flood attacks. The second step is to use statistical tools (e.g., Auto Regressive Model) to find the variables from the potential attackers that are highly related to the key variable. For example, the number of ICMP echo reply packets at the potential attackers is highly correlated with the key variable for Ping flood attacks. The third step is to build a normal profile using the found variables from the potential attackers. Any anomalies from potential attackers compared with the normal profile are regarded as strong indications of an attack. Step one and two are completed during the off-line training period and step three is done on-line.

The vulnerability of this scheme is that the efficacy of training is based on the features of known attacks. The attacker can disturb or disable the detection scheme by inventing new attacks. As DDoS attacks do not necessarily need to use any particular type of traffic, it is easy for the attacker to create a new type of attack just by combining different types of attack traffics. This causes multiple key variables from the target, and the correlations between the variables from the potential attackers and the target will become extremely complex, which complicates the process of building a normal profile and makes the detection less effective.

In Pushback [174], flow belonging to DDoS attacks is identified by considering high traffic volume to the victim. Then right drop probability for such traffic is calculated by detection system that conveys this information to the upstream routers, which in turn could drop packets belonging to the attack traffic themselves. It is effective in countering high rate disrupting flooding attacks only.

Cheng et al. [41] proposed to use spectral analysis to identify DoS attack flows. Generally, DoS attack flows are not regulated by TCP flow control protocols as normal flows are. Hence, DoS attack flows have different statistical features compared with normal flows. Cheng et al. [41] use this assumption for DDoS detection. In this approach, the number of packet arrivals in a fixed interval is used as the signal. In the power spectral density of the signal, a normal TCP flow will exhibit strong periodicity around its round-trip time in both

flow directions, whereas an attack flow usually does not. Spectral analysis techniques are only valid for TCP flows. As UDP and ICMP are connectionless protocols, the periodic traffic behavior is unexpected. Attackers can use UDP or ICMP traffic to confuse the detection scheme. Moreover, the attacker can mimic the periodicity of normal TCP flows by sending packets periodically.

Peng et al. [195] have proposed a new DoS attack detection scheme using source IP address monitoring. Generally, the set of source IP addresses that is seen during normal operation tends to remain stable. In contrast, during DoS attacks, most of the source IP addresses have not been seen before. By using a carefully prebuilt IP Address Database, it is possible to sequentially monitor the proportion of new source IP addresses seen by the target, and detect any abrupt change using a statistical test. An abrupt change of the proportion of new source IP addresses is a strong indication of a DoS attack.

Feinstein et al. [126] proposed entropy based DDoS detection scheme in which they calculated randomness in a particular feature of packet (e. g. source IP address) for normal flows and whenever entropy crosses threshold in actual scenario, it is termed as an anomaly and hence attack. It can detect high rate flooding attacks. This approach fail against varied rate attacks wherein intelligent attacker mixes low and high rate zombie machines to generate attack traffic in such a manner that overall entropy remains unchanged.

Wang et al. [210] proposed SYN detection to detect SYN flooding attack, by monitoring statistical changes. The ratio of SYN packets to FIN and RST packets was used. The attack detection is based on the following assumptions. First, the random sequence is statistically homogeneous. Second, there will be a statistical change when an attack happens. This approach is based on the fact that a normal TCP connection starts with a SYN packet and ends with a FIN or RST packet. When the SYN flood starts, there will be more SYN packets than FIN and RST packets. The attacker can avoid detection by sending the FIN or RST packet in conjunction with the SYN packets. Another limitation of the proposed approach is that it is not applicable for other attacks i.e UDP flooding, ICMP flooding, etc.

Bencsath et al. [27] have given a traffic level measurement based approach, in which incoming traffic i.e. number of packets or bytes count per unit time is monitored continuously and dangerous traffic intensity rises are detected. This approach is better suited for isolating large traffic changes (such as bandwidth flooding attacks), but low rate attacks can not be

detected and characterized because these attacks do not cause detectable disruptions in traffic volume.

Mirkovic et al. [106] proposed a system called D-WARD that does DDoS attack detection at the source based on the idea that DDoS attacks should be stopped as close to the sources as possible. D-WARD is installed at the edge routers of a network and monitors the traffic being sent to and from the hosts in its interior. If an asymmetry in the packet rates generated by an internal host is noticed, D-WARD rate limits the packet rate.

The drawback of this approach is that there is a possibility of numerous false positives while detecting DDoS conditions near the source, because of the asymmetry that there might be in the packet rates for a short duration. Furthermore, some legitimate flows like real time UDP flows do exhibit asymmetry. Moreover, at source why all clients use and bear the expense where the benefit is meant for others?

Blazek et al. [154] proposed batch detection to detect DoS attacks. DoS attack detection is performed by monitoring statistical changes. The first step for this method is to choose a parameter for incoming traffic and model it as a random sequence during normal operation. In this method, a variety of parameters, such as TCP and UDP traffic volume, were used. The attack detection is based on the following assumptions. First, the random sequence is statistically homogeneous. Second, there will be a statistical change when an attack happens. To beat the detection scheme of Blazek et al. [154], the attacker can carefully mix different types of traffic to ensure that the proportion of each traffic is the same as it is in normal traffic. Therefore, separating different types of traffic cannot make the attack behavior more conspicuous.

Chen et al. [214] used distributed change-point detection (DCD) architecture using change aggregation trees (CAT) to detect DDoS attack over multiple network domains. The idea is to detect abrupt traffic changes across multiple network domains at the earliest time. Early detection of DDoS attacks minimizes the flooding damages to the victim systems serviced by the provider.

Entropy based approach [119] treats DDoS anomalies as events that disturb the distribution of traffic features and entropy is used as metric to measure distribution of the traffic features. Although, by carefully calculating upper and lower threshold values of entropy, these approaches are successful in countering high rate disruptive and diluted low rate degrading flooding attacks but varied rate attacks are unbeaten for them wherein

intelligent attacker mixes low and high rate zombie machines to generate attack traffic in such a manner that overall entropy remains unchanged.

Lee and Stolfo [207] use data mining techniques to discover patterns of systems features that describe program and user behavior and used a classifier that can recognize anomalies and intrusions. A mechanism called congestion triggered packet sampling and filtering is proposed by Huang et al. [215]. According to this approach, a subset of dropped packets due to congestion for statistical analysis is selected. If anomaly is indicated by the statistical results, a signal is sent to the router to filter the malicious packets.

Table 2.2. Comparison of various detection approaches classified based on detection method

| Detection Category | NPSR | Complexity | Detection Accuracy | Limitations |
|--------------------|-------------|------------|--------------------|--|
| Pattern Detection | High | Low | High | Detection of the novel attacks are not possible |
| Anomaly Detection | Medium | Medium | Medium | False positives and negatives rate is very high, since defining normal system behavior and setting threshold values is difficult |
| Hybrid Detection | High-Medium | High | High | Complexity and cost of implementation is very high to be deployed in practice |

iii). Hybrid attack detection

Hybrid attack detection combines the positive features of both pattern and anomaly based attack detection models to achieve high detection accuracy, low false positives and negatives, and, thus, a raised level of cyber trust. Though hybrid attack detection approach decreases false positive rate but complexity and cost of implementation is high [117]. Table 2.2 shows the comparison of various detection approaches based on detection method.

2.8.1.3 DDoS attack response

The goal of the attack response is to relieve the impact of the attack on the victim while imposing minimal collateral damage to legitimate clients. We classify attack response mechanisms as follows:

A. Attack source/ path identification

Once an attack has been detected, an ideal response would be to block the attack traffic at its source. Unfortunately, there is no easy way to track IP traffic to its source due to the stateless of the IP protocol. The attacker can easily spoof the IP source address and send it to any destination without notice. In order to address this limitation, several enhancements have been proposed to support IP traceability [128, 183]. Attack source identification mechanisms provide the victim with information about the identity and path taken by the machines that are responsible for performing the attack. A brute force solution to traceback can be obtained by having every router mark every packet as it passes through it. An alternative brute force solution requires every router to keep a record of every packet that passes through it. Such solutions are infeasible because: (1) they require a large and unbounded space in each packet (or router); (2) they require a large overhead at every router. Thus, most existing approaches to traceback attempt to reduce the above two effects. Some of traceback schemes proposed in the literature are discussed below:

Burch and Cheswick [85] propose a controlled flooding of links to determine how this flooding affects the attack stream. Flooding a link will cause all packets, including packets from the attacker, to be dropped with the same probability. We can conclude from this that if a given link was flooded, and packets from the attacker were slowed, then this link must be part of the attack path. Then recursively upstream routers will perform the same test until the attack path is discovered. This scheme requires considerable knowledge of network topology and the ability to generate huge traffic in any network link. The most important problem with this approach is that it is resource intensive and highly intrusive. In fact, this approach may be viewed itself as a DoS attack.

ICMP traceback has been proposed by Bellovin [159]. According to this mechanism every router samples the forwarding packets with a low probability and sends an ICMP traceback message to the destination. If enough traceback messages are gathered at the victim, the source of traffic can be found by constructing a chain of traceback messages. This mechanism have several limitations, e.g. ICMP traffic is increasingly differentiated and may be filtered or rate-limited differently from normal traffic, ICMP messages are transmitted over already congested channel.

Savage et al [180] suggest probabilistically marking packets as they traverse routers in the Internet. More specifically, they proposed that the router mark the packet, with low

probability (say, $1/20,000$), with either the router's IP address or the edges of the path that the packet traversed to reach the router. Song et al. [67] propose an enhanced scheme of probabilistic packet marking and also set up a scheme for router authentication. However, the authentication scheme is complex to implement.

Initially, Belenky and Ansari [8] outlined deterministic packet marking. Their idea is to put, with random probability of 0.5, the upper or lower half of the IP address of the ingress interface into the fragment id field of the packet, and then set a reserve bit indicating which portion of the address is contained in the fragment field. By using this approach they claim to be able to obtain 0 false positives. Unfortunately, their approach is extremely light on details and fails to address how upstream routers are to account for their marking if they too are also marked. Also, their approach does not take into account the non-unique status of an IP address that NAT confers on network topologies. Stone [162] proposes routing suspicious packets on an overlay network using ISP edge routers. By simplifying the topology, suspicious packets can easily be re-routed to a specialized network for further analysis.

Snoeren et al. [10] proposed a scheme to let routers store a record of every packet passing through them and then trace back the origin of the packet by using the history stored in the routers. Although they describe a smart scheme to compress the storage, it is still a huge overhead for the router to implement this scheme, especially with the increasing network speed.

Some other traceback schemes are Source Path Isolation Engine (SPIE, also called hash-based traceback) [9], Algebraic-Based Traceback Approach (ATA) [56], deterministic edge router marking (DERM) [183], control-agent model for single ISP domain [199] and multiple ISP domains [202]. A survey and analysis of traceback schemes is presented in [201, 220].

In summary, existing solutions to the traceback problems attempt to reduce the state and processing overheads by a combination of probabilistically generating traceback information and/or using hash function to reduce the marking state. As such, for these to be effective traceback requires a colossal number of packets and vast computing resources for reconstructing the attack graph.

B. Filtering

Segregating and filtering off malicious flows without hurting legitimate traffic is the main goal for defenders. Filtering techniques are used to filter out incoming traffic completely that has been characterized as malicious by the detection mechanism. Examples include dynamically deployed firewalls [190], and also a commercial system, TrafficMaster [139].

Response schemes based on filtering is much more common. Packet filtering can be done according to classification rules [104, 122, 193, 213] or can be based on link testing schemes [85, 96, 162] and pushback schemes [85, 98, 158, 159, 160, 163] that traceback to the source and drop the attack traffic.

Many routers include a feature called input debugging [96, 162] that allows an operator to filter particular packets on some egress port and determine which ingress port they arrived on. This capability is used to implement a trace as follows: first, the victim must recognize that it is being attacked and develop an attack signature. The network operator installs a corresponding input debugging filter on the victim's upstream egress port. This filter reveals the associated input port, hence identify which upstream router originated the traffic. The process is then repeated recursively on the upstream router, until the originating site is reached. Once this reroute is complete, network operator can then use input debugging at the tracking router to investigate where the attack enters the ISP network. The most obvious problem with the input debugging approach is that it requires considerable management overhead time, attention and commitment of both the victim and the remote personnel and approximate technical skills.

Roshan Thomas proposed a legitimacy-based DDoS filtering scheme, NetBouncer [163]. It maintains a legitimacy list to differentiate malicious packets and legitimate packets. If the packets are not on the list, NetBouncer will proceed to administer a variety of legitimacy tests to challenge the client to prove its legitimacy. However, this scheme has not been tested in real network environment.

Another filtering mechanism that has been proposed is Hop-Count Filtering [40]. The idea is that although the attacker can forge any field in the IP header, the number of hops an IP packet takes to reach its destination cannot be falsified. So Hop-Count Filtering (HCF) could be mainly applied to filter the spoofed IP packets. It extracts the TTL information from the IP head to compute the hop-count, then by comparing the computed hop-count with the stored hop-count, the likely spoofed packets are identified. Because this method still has a

false positive rate, it takes no action to defend attacks until in the action state. Steven J. Templeton also found that the final TTL values from an IP address are generally clustered around a single value [182], but no solution has been provided yet.

An aggregation detection algorithm, pushback was proposed by Mahajan et. al [158, 159, 160] which is based on the IP destination addresses of the packets. In the pushback scheme, a router notifies its upstream routers when it detects an attack, then the upstream routers drop such packets so that the legitimate traffic will be affected less. When the congestion level of a network is high, such as the total incoming bandwidth is 1.2 times of the output bandwidth, the pushback algorithm begins to match the destination address of each dropped packet against the routing table and selects the longest matching prefix, which constitutes the congestion signature. In the router, a rate limiter is added to decide whether a packet is dropped or forwarded. It takes effect when the incoming traffic exceeds a threshold and drops the packets matching the congestion signature. The excess packets are also dropped by the output queue.

A pushback daemon receives dropped packets from both the rate limiter and the output queue. It analyzes the number of dropped packets, determines whether there is an attack going on and how to react against it. Then it updates the parameters of the rate limiter and informs upstream routers periodically. It also listens to the requests from downstream routers. If dropping the packets matching this congestion signature can't effectively reduce the incoming traffic, it is likely that more than one attack is happening. This algorithm needs to be performed repeatedly to find out more prefixes. If no such prefix can be found, it means the traffic is not caused by an attack, but by increase in general traffic.

C. Rate throttling

Rate-throttling [25, 55, 106, 159] is a lenient response technique that imposes a rate throttle on the incoming traffic that has been characterized as malicious by the detection mechanism, usually deployed when the detection mechanism has a high level of false positives or can not precisely characterize the malicious traffic. The disadvantage is that such an approach will allow some attack traffic through, so extremely high-scale attacks might still be effective even if all traffic streams are rate-limited.

Pushback [158, 159, 160] was proposed as a mean to relieve the Internet from the congestion induced by bandwidth-flooding attacks or flash crowds. To this end, a receiver

identifies the last-hop routers that forward to it above a certain rate and ask them to rate-limit traffic addressed to it; each of these routers can then repeat the same process, i.e. identify the upstream routers that forward traffic above a certain rate to the receiver and ask them to rate-limiting traffic addressed to the receiver. Peering domains establish bilateral agreements that allow routers from one domain to send pushback requests to adjacent routers in the other domain. In this way, rate limiting of the bandwidth flood or flash crowd is pushed away from its target and closer to its sources. Note that the target of a bandwidth flood does not have to identify the source of undesired traffic.

Yau et. al. [68] used router throttles to combat DDoS attacks against Internet servers. A proactive approach is followed in the sense that before aggressive packets can converge to overwhelm a server, routers along forwarding paths, regulate the contributing packet rate to more moderate levels, thus averting an impending attack. The basic mechanism is for a server under stress, to install a router throttle at an upstream router several hop away. The throttle limits the rate at which packets can either be dropped or rerouted to alternative server. However, attackers can exploit communication part as no secure ways are used to send throttle messages in same and different domain. In case of low rate attack, collateral damage is more as normal packet survival ratio is very low.

Xiong et. al. [217] also took the defense of DDoS attack as a congestion control problem. They propose to use backward pressure propagation, feedback control scheme to defend DDoS attack. They used rate-based and queue-length based algorithms to create the feedback signal accordingly. Once the input traffic rate or the output queue length has exceed the desired threshold, a feedback signal is sent to adjust the admitted portion of traffic in different input and output ports to put the rate and queue length below the threshold. The method is effective to make sure that the network traffic works in a tolerable level during DDoS attack. However, they don't set up a scheme to discriminate good traffic from bad traffic.

D. Reconfiguration

Reconfiguration mechanisms change the topology of the victim or the intermediate network to either add more resources to the victim or to isolate the attack machines. Examples include reconfigurable overlay networks [62], attack isolation strategies [110], etc.

2.8.1.4 DDoS attack tolerance and mitigation

Attack tolerance & mitigation focuses on minimizing the attack impact and tries to provide optimal level of service as per quality of service requirement to legitimate users while service provider is under attack. This is not a comprehensive solution in any way, however it can complement other approaches to work in parallel and achieve their goals by providing sufficient assurance in terms of time to provider so that the legitimate clients are being served. Moreover this approach can itself get complemented from other approaches and then synergistic effort is best performance for the clients. We classify attack tolerance and mitigation mechanisms as follows:

A. Overprovisioning

An abundance of resources e.g. a pool of servers with load balancer, high bandwidth link between victim machine and upstream routers are used to tolerate these attacks [108, 155, 156].

Kargl et al [73] have proposed a load balancer based technique in which a cluster of web servers are sheltered by firewall and load balancer. Firewall applies traditional prevention measures and filters suggested traffic by load balancer time to time. Load balancer as per load works as translator and also allocates requests to appropriate web server. Moreover traffic monitors at web servers and load balancer in consultation with manager, deduce classification for packets to be treated by load balancer using class based queuing. The same CBQ is also used at web servers for sending response to various classes. Sairam et al [16] also worked for fair bandwidth allocation using load balancing. Overprovisioning [156] works as tolerance based scheme for node based defense. Nevertheless, due to the absence of attacks most of the time or at low attack loads, especially when client load is low, it has cost associated with it.

B. Router's queue management

Router's queue management schemes aim to reduce attack impact or congestion simply without providing fairness between the traffic flows. Therefore, false positive rate is very high [172, 173].

Random Early Detection (RED) [173] represents this class of algorithms. A router only maintains a simple FIFO queue for all traffic flow and drops the arriving packet randomly during congestion. The packet drop probability increases with growth in queue.



RED can reduce the delay time for the most of traffic flow, by keeping the output queue size small,

Misbehaving traffic flows can not be penalized by RED. Floyd et al [172] have proposed a technique to use a lightweight detection algorithm to identify unresponsive flows and then explicitly manage the bandwidth of these flows, in order to improve the RED's probability to penalize the misbehaving traffic flows. Their technique performs tests that identifies flows that are unresponsive, TCP-friendly, or high-bandwidth and regulates them. Lin et al. [63] propose a technique called Flow Random Early Drop (FRED) to maintain the fairness between the traffic flows. It only keeps the states for the flows that have packets buffered in the router. The packet for this flow is dropped randomly once queue length of one flow in the buffer is between minimum (min) and maximum (max). Once the queue length is larger than max, the incoming packet is dropped. To count the number of times the flow has failed to respond to congestion notification, the router also keeps the information. Penalties are taken to the unresponsive flows. These variants of RED incur extra implementation overhead since they collect certain type of state information. Ott et al. [191] set up another interesting variant called stabilized RED (SRED).

SRED stabilizes the FIFO buffer occupancy independently of the number of active flows. It maintains a data structure called Zombie list, which serves as a proxy for information about the recent flows. By doing this, it can estimate the number of active flows and identify the candidates for the misbehaving flows. Although SRED can identify misbehaving flows, it hasn't set up a scheme to penalize them. To improve this scheme, a stateless active queue management scheme called CHOKe (CHOOSE and Keep for responsive flows, CHOOSE and kill for unresponsive flows) [24] was proposed to approximate fair bandwidth allocation. CHOKe draws a packet from the FIFO buffer at random and compares it with the arriving packet if they both belong to the same flow, they are both dropped: else the randomly chosen packet is left intact and the arriving packet is dropped with a probability p which depends on the congestion level.

Although this scheme is effective in defending the unresponsive traffic flow, it performs poorly for a large number of small traffic flows. So it is still vulnerable in defending flash crowds [101] and DDoS attacks. A new variant of RED called RED with Preferential Dropping (RED-PD) to identify high bandwidth flows and control the bandwidth obtained by these flows is proposed by Mahajan et al [160]. However, it controls the high bandwidth

flows by estimating their arriving rate, which is not very accurate. Furthermore, the test for unresponsive traffic flows need to be more accurate to maintain fairness.

Performance of various queuing algorithm implemented in a network router under flooding DDoS attack are investigated by Lau et al. [76]. They tried to find whether legitimate users can obtain desired service or not. The simulation results illustrates that RED and class based queuing (CBQ) are successful in providing a part of bandwidth requested by the legitimate user during high rate flooding DDoS attack. The topology, traffic generation model and applications used in the simulation are very simple compared to the realistic network topology.

It was suggested that DDoS attacks could be countered by applying resource allocation techniques on network bandwidth to guarantee fairness. Two approaches namely Integrated services (IntServ) [209] and Differentiated Service (DiffServ) [169] aimed at isolating flows with specific QoS requirements from lower-priority traffic. IntServ uses the resource reservation protocol (RSVP) to coordinate the allocation of resources along the path that a specific traffic flow will pass. The link bandwidth and buffer space are assured for the specific traffic flow. In [218], taxonomy of approaches to per-class QoS differentiation is presented.

C. router's traffic scheduling

Router's traffic scheduling algorithm can reduce congestion or attack impact with the fairness between the traffic flows, but they are too expensive in terms of delays and state monitoring [12, 15, 149]. Fair queuing algorithm (FQ) is a classic example for scheduling algorithm. FQ requires the router to partition the input traffic into separate queues and use a separate buffer space for each queue. State for each flow is kept and managed individually by router. One flow can not degrade the quality of another.

Nevertheless FQ needs, complex per-flow-state, which makes it too expensive to be widely implemented. Stoica et al. [93] proposed a new scheduling algorithm called core stateless FQ to categorize the routers into edge and core routers and to reduce the cost of keeping per flow state in every router. Per flow state information and estimation of the arriving rate for per flow is maintain by an edge router. These estimates are inserted into the packet headers and passed on to the core routers. The core routers keep a simple stateless FIFO queue and drops packet according to the estimates in the packet header during the

congestion. Although the scheme simplifies FQ, it is still very expensive to keep per flow state information. However because of lesser metering in core routers, it is better than previous one. Still extracting packet information from the packet in core, adds to the complexity of this scheme. To approximate FQ at a smaller implementation cost, Mckenny [149] proposed stochastic fair queuing (SFQ). SFQ classifies packets into smaller number of queues than FQ using a hash function. Although it reduces the complexity of FQ, it still needs 1,000 to 2,000 queues in a typical router to approximate FQ performance.

Fairness between the traffic flows can be ensured using Scheduling algorithms [12, 93, 149], but they are too expensive in terms of delays, state monitoring and don't scale well to a large number of users. Moreover large number of slow rate DDoS traffic can still prove lethal to victims.

D. Target roaming

Active server changes its location within distributed homogeneous servers proactively to eliminate or curtail DDoS attacks impact [178].

Khattab et al [178] proposed proactive server roaming based approach to defend DDoS attacks, which was further extended by sangpachatanaruk et al [43]. In proactive server roaming based approach, one server from cluster of servers is made active server at a particular time. The timing and actual address of server is calculated by legitimate clients with the help of preloaded client module. The incomplete connections and sessions are replicated on the roamed servers also using secure migration protocols. So by this way, only legitimate clients can access the server whereas all others are filtered either through dynamically configured router access lists or firewall. Moreover attackers' packets are logged for further analysis.

The firewall does not give proper protection from high volume of packets. During roaming and replication, even legitimate packets suffer. This methodology should be tested on real Internet like topology using Internet like traffic models running various types of services. Moreover, various secure communication methods and roaming strategies can also be explored in simulations to get better results.

Changing victim IP address [108] described earlier methods for node based DDoS defense.

E. Others

Based on the system load, dynamic resource pricing [64] imposes dynamically changing prices on resources. This cost has to be distributed from the requesting client before the resource is allocated. A special case of such a pricing mechanism is Client puzzles [13, 188], where the client has to solve a cryptographic problem with varying complexity before the server allocates resources to the request and starts servicing it. Puzzle auctions [212] is based on similar concepts.

To combat DDoS attacks against Internet servers, Yau et al [68] have used router throttles. The defense of DDoS attack as a congestion control problem is also obtained by Xiong et al [217]. They propose to use backward pressure propagation, feedback control scheme to defend DDoS attack. To create the feedback signal accordingly, they used rate-based and queue-length-based algorithms. Once the input traffic rate or the output queue length has exceeded the desired threshold, a feedback signal is sent to adjust the admitted portion of traffic in different input and output ports to put the rate and queue length below the threshold. To make sure that the network traffic works in a tolerable level during DDoS attack, the method is effective.

2.8.2 Based on Degree of Deployment

A typical DDoS defense system consists of detection of attack, characterization of attack sources, and response to attack traffic. It can be deployed using various ways. Classification based on degree of deployment categorizes the DDoS defense mechanisms in the following two categories:

2.8.2.1 Single point or Autonomous defense

Single point or autonomous defense mechanisms [14, 27, 118, 126] consist of a single defense node that observes the attack, analyses the traffic and applies response.

2.8.2.2 Multipoint or Distributed defense

Multipoint or distributed defense mechanisms [8, 10, 11, 56, 107, 174, 180, 181, 214] consist of multiple defense nodes, generally with the same functionalities that are deployed at various locations and organized into network. Nodes communicate through the network and coordinate their actions to achieve a better overall defense.

2.8.3 Based on Deployment Point or Location

DDoS attack streams on the Internet originated from geographically distributed machines, are forwarded by core routers and converge at the victim network. There is interaction of three types of networks: source networks containing the host attack machines, several intermediate networks that forward attack traffic to the victim, and the victim network that host the target. Figure 2.8 illustrates this interaction. Each of the involved networks can host DDoS defense systems. Classification based on deployment points or locations categorizes the DDoS defense mechanisms in three categories, namely victim network, intermediate network, and source network.

2.8.3.1 Victim or target Network

Most of the existing DDoS defenses systems have been designed to work at the victim's network [14, 27, 118, 126]. This is understandable, because it can closely observe the victim system's behavior, model its normal behavior that can be used to find variety of anomalies. So it is best placed to detect DDoS attack, however these systems may themselves become targets of DoS attacks, by sending a sheer amount of traffic from various distributed attack sources that can overwhelm it. Storage and processing power requirement to store and examine various statistical measures are very high in these systems.

2.8.3.2 Intermediate Network

These mechanisms [8, 10, 11, 56, 174, 180, 181, 214] are deployed at core routers. Since core routers can handle large volume, highly aggregated traffic, they are likely to overlook all but large scale attacks. However, response to attacks is likely to inflict collateral damages, as core routers can only accommodate simple rate-limiting requests and cannot dedicate memory or processing cycle to traffic profiling.

At the intermediate network i.e. in the core of Internet, many solutions, such as pushback, SOS, and traceback are deployed. They all put load on core routers, which are meant for forwarding packets at high speed as per Internet Design. Besides, intermediate network is not owned by single administrative domain. Therefore, establishing cooperation and trust relationship between different domains, such that request originating from one domain will be honored by the other or not, or the module to be installed in other domain will be allowed or not, are the concerns that have practically no answers.

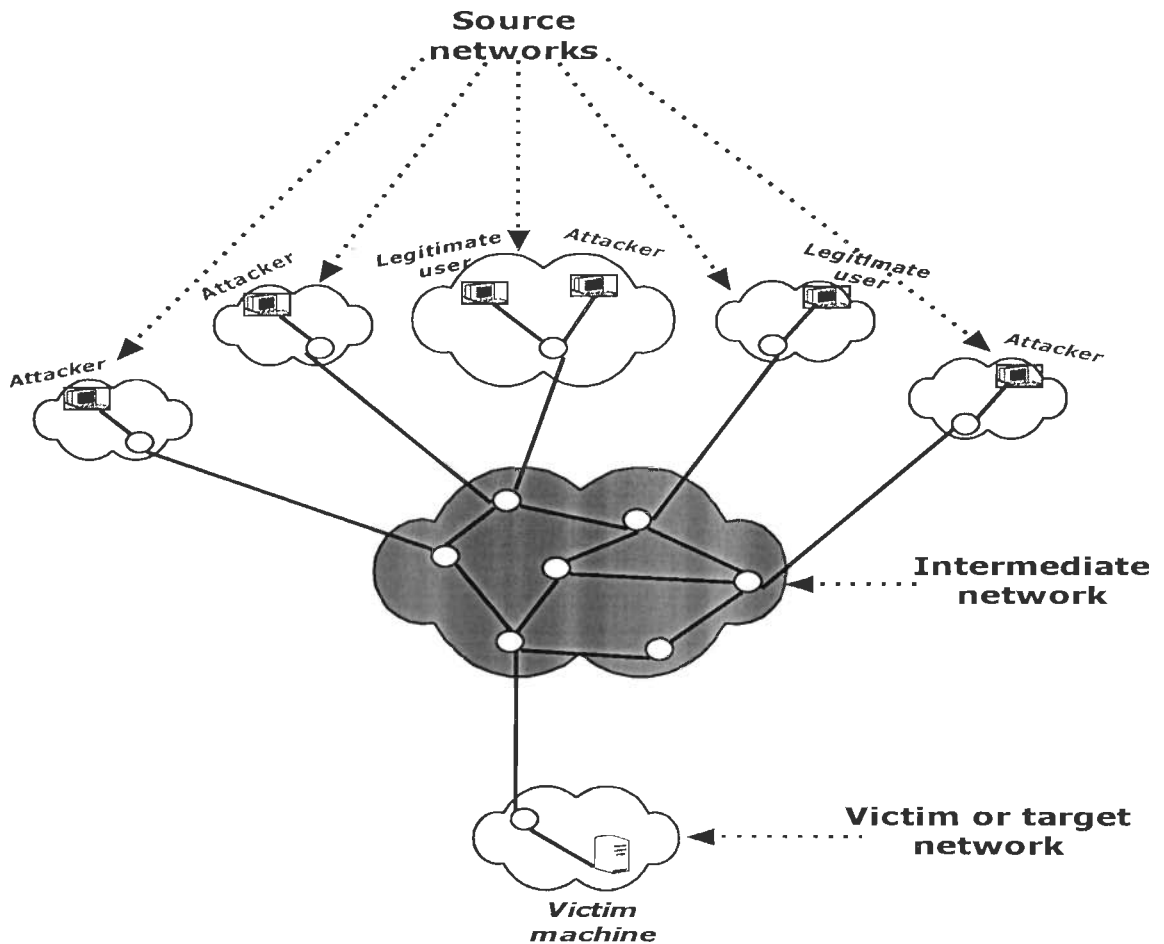


Figure 2.8. DDoS defense deployment points

2.8.3.3 Source Network

These mechanisms are deployed at source end i.e. edge routers and detect DDoS attacks at the source, based on the idea that DDoS attacks should be stopped as close to the source as possible. At this place attack flows are not so aggregated yet, so it would put less burden on the defense systems to analyze them. And since they would be cut off at the source, it would save transit networks from transporting malicious traffic. This approach however requires a very large scale deployment in order to be effective. And since attack streams in the source network usually are small in volume, they may be more difficult to detect and a large number of false positives and negatives are there [106, 192].

To stop origin of DDoS traffic at source network, prevention methods, such as ingress/egress filtering and repairing security holes are implemented. Absence of incentives,

per packet filtering overheads, and security measures awareness stand in the way of DDoS defense deployed at the source network. D-WARD is an example of source–end defense scheme. Two hard challenges are faced by it. First, in a highly distributed attack, each source network is responsible for only a small fraction of the attack traffic, which is unlikely to generate anomalous statistic. Secondly, a witty DDoS attacker can also control the attack traffic from each source network to be within normal range because ultimately it is the aggregation of attack traffic and not individual source traffic which is going to inflict damage to the victim. Moreover, the biggest problem in source-end defense is requirement of global deployment, which is impossible to achieve as Internet has no central control.

2.8.4 Based on Degree of Cooperation

DDoS defense mechanisms can perform defensive measures either alone or in cooperation with other entities in the Internet. Classification based on degree of cooperation categorizes the DDoS defense mechanisms in the following categories:

2.8.4.1 Independent

As name suggest, independent defense mechanisms [127, 138, 154, 161, 192, 205, 210] work independently at the location where they are deployed. Firewalls and intrusion detection systems provide easy examples of autonomous mechanisms. Even if a defense system performs its function in a distributed manner, it would still be considered autonomous if it can be completely deployed within the network it protects.

2.8.4.2 Cooperative

Cooperative defense mechanisms [11, 107, 174, 214] are capable to work independently, but can cooperate with other entity to increase performance significantly. The aggregate congestion control (ACC) system [159] deploying a pushback mechanism [98] is an example of cooperative scheme.

2.8.4.3 Interdependent

Interdependence defense mechanisms [8, 10, 56, 180, 181, 214] can not operate independently at a single deployment point. They either require deployment at multiple networks, or rely on other entities for attack prevention, detection or response. Traceback

mechanisms [8, 9, 10, 56, 67, 162, 180] provide examples of interdependent mechanisms. Secure overlay services [11] are another example of an interdependent mechanism.

2.9 Predicting Number of Zombies in a DDoS Attack

Anomaly based DDoS detection systems construct profile of the traffic normally seen in the network, and identify anomalies whenever traffic deviate from normal profile beyond a threshold. This deviation in traffic beyond threshold is used in the past for DDoS detection but not for finding number of zombies. This deviation in traffic can be used to predict number of zombies. A real time estimation of the number of zombies in DDoS scenario is helpful to suppress the effect of attack by choosing predicted number of most suspicious attack sources for either filtering or rate limiting. Moore et. al [65] have already made a similar kind of attempt, in which they have used backscatter analysis to estimate number of spoofed addresses involved in DDoS attack. This was an offline analysis based on unsolicited responses. In another approach [121], authors have used linear regression and correlation analysis to predict number of zombies. But due to the nonlinear nature of DDoS attack traffic, this method is unable to predict the number of zombies accurately.

2.10 Research Gaps

One of the primary goal of attack prevention schemes [11, 104, 122, 147, 187, 193] is to handle IP spoofing, a fundamental weakness of the Internet. However, as attackers gain control on large number of compromised computers, attackers can direct these “zombies” to attack using valid source addresses. Since the communication between attackers and “zombies” is encrypted, only “zombies” can be exposed instead of attackers.

To stop IP spoofing, to repair security holes by patches, and to stop intrusion, prevention approaches have lots of hurdles in terms of host based incentives, global deployment, overheads to check extra packet headers, installation of patches as soon as they are developed and released, and inability to detect new attacks. Moreover, in prevention techniques, non-spoofing, subnet spoofing, en-route spoofing and DRDoS based attacks have no reliable solution. In addition, on an average security awareness is still not enough, so expecting installation of security technologies and patches in large base of Internet looks an ambitious goal in near future. Therefore, relying only on attack prevention schemes is not enough to stop DDoS attacks.

Most of the existing schemes [7, 27, 41, 79, 106, 116, 117, 126, 138, 154, 174, 177, 192, 195, 199, 205, 207, 210, 214, 215] proposed in the literature for detecting DDoS attacks have certain limitation. These techniques are used to detect attacks either at source network or at victim network. The majority of the existing DDoS defense systems have been designed to work at the victim network [7, 27, 116, 126, 174, 192, 207]. This is understandable, because victim network can closely observe the victim system's behavior, model its normal system profile that can be used to find variety of anomalies. Therefore, it is best place to detect DDoS attack. Nevertheless, victim system may itself become target of DDoS attacks, by sending a sheer amount of traffic from various distributed attack sources that can overwhelm it. Storage and processing power requirement to store and examine various statistical measures are very high in these systems.

In contrast, schemes which detect DDoS attack at source network [106, 195, 199] are not effective, as they require a very large scale deployment in order to be effective. In addition, since attack streams in the source network usually are small in volume, they may be more difficult to detect and a large number of false positives and negatives are there.

Detection schemes based on monitoring the volume of traffic [27, 79, 154, 192, 199] are better suited to detect high rate disrupting attacks (HRD), which completely disrupt the services to legitimate client. Low rate degrading (LRD) attacks consume a small portion of victim's resources are not detected using these schemes. However, the accurate detection of these low rate flooding attacks is very important, as detection closer to source is possible, which is otherwise very difficult because of lesser volume of attack traffic source. Entropy based approaches [14, 119, 126] can detect low rate degrading attacks, but fail against varied rate attacks wherein intelligent attacker mixes low and high rate zombie machines to generate attack traffic in such a manner that overall entropy remains unchanged.

Signature based schemes [138, 205] employ a priori knowledge of attack signatures. Signature based techniques are only effective in detecting traffic of known DDoS attacks whereas new attacks or even slight variations of old attacks go unnoticed. Anomaly based techniques [20, 41, 102, 106, 116, 174, 195, 199] can detect novel attacks; however, it may result in higher false alarms. Availability of user friendly attack tool kits and their source codes give flexibility to attacker to create a variety of new attacks by error and trial. Most of the detection schemes can easily be defeated by developing attacks through this error and trial method. Even existing variety of attacks are sufficient to disguise most of prevailing detection

methods. Detection models normally have tunable parameters like clustering level (traffic aggregation for monitoring), sample window size, and thresholds etc. In most cases, researchers offer no guidance on parameter variations or their effects on detection performance. Ad hoc training is typically required to tune parameters as per desired detection performance. But, actually researchers often optimize parameters of their own experimental test cases so as to show better results. Overall, in all of the detection techniques, high computational and memory overheads are involved and they are very complex in nature.

Once an attack has been detected, an ideal response would be to block the attack traffic at its source. Unfortunately, there is no easy way to track IP traffic to its source due to the statelessness of the IP protocol. The attacker can easily spoof the IP source address and send it to any destination without notice. Most of the existing solutions to the traceback problems attempt to reduce the state and processing overheads by a combination of probabilistically generating traceback information and/or using hash function to reduce the marking state. As such, for these to be effective, traceback requires a colossal number of packets and vast computing resources for reconstructing the attack graph. In addition, to achieve IP traceback, co-operation between ISP's is always difficult to achieve.

Filtering techniques are used to filter out incoming traffic completely that has been characterized as malicious by the detection mechanism. However, it is always very difficult to distinguish malicious packets from legitimate packets therefore; these techniques cause high number of false positives. The disadvantage of rate limiting scheme is that it allows some attack traffic through, so extremely high-scale attacks might still be effective even if all traffic streams are rate-limited.

Attack tolerance & mitigation focuses on minimizing the attack impact and tries to provide optimal level of service as per quality of service requirement to legitimate users while service provider is under attack. This is not a comprehensive solution in any way, however it can complement other approaches to work in parallel and achieve their goals by providing sufficient assurance in terms of time to provider so that the legitimate clients are being served.

For predicting number of zombies, previous approaches [121] have used linear regression and correlation analysis. But due to the nonlinear nature of DDoS attack traffic, these methods are unable to predict the number of zombies accurately. Some of the research gaps mentioned above have been investigated in this thesis.

2.11 Chapter Summary

DoS attack causes either disruption or degradation on victim's shared resources, as a result preventing legitimate users from their access right on those resources. DoS attack may target on a specific component of a computer, entire computer system, certain networking infrastructure, or even entire Internet infrastructure. Attack can be performed either by exploiting the natural weakness of a system, which is known as logical attack or overloading the victim with high volume of traffic, which is called flooding attack. A distributed form of DoS attack is called DDoS attack, which is generated by many compromised machines to coordinately hit a victim.

In this chapter, we have covered an overview of the DDoS problem, its basic causes, targeted resources, attack modus operandi and available DDoS attack tools. However, DDoS attacks are adversarial and constantly evolving. By the time a particular kind of attack is successfully countered, a slight variation is designed that bypasses the defense and still performs an effective attack. In addition, a comprehensive study of a wide range of DDoS attacks and defense methods proposed to combat them is presented. This provides a better understanding of the problem, current solution space and future research scope to defend against DDoS attacks.

CHAPTER 3

DETECTING DISTRIBUTED DENIAL OF SERVICE ATTACKS USING FLOW-VOLUME BASED APPROACH

3.1 Introduction

In the previous chapter, we have seen that DDoS attacks are major threat to the Internet and there is no efficient approach to detect these attacks. Attack detection should aim to detect an ongoing attack quickly when a network senses any occurrence of these attacks. Timely detection of DDoS attacks is the only key to respond to these attacks and to protect system from failure. Most of the existing schemes [7, 27, 41, 79, 106, 116, 117, 126, 138, 154, 174, 177, 192, 195, 199, 205, 207, 210, 214, 215] proposed in the literature for detecting DDoS attacks have certain limitations. Availability of user friendly attack tool kits [28, 36, 45, 46, 57, 58, 59, 60, 99, 136, 171] and their source codes gives flexibility to attacker to create a variety of new attacks by error and trial method. Most of detection schemes can easily be defeated by developing attacks through this error and trial method. Existing variety of attacks are also sufficient to disguise most of prevailing detection methods. In addition to this, in all of the detection techniques, high computational and memory overheads are involved and they are very complex in nature.

In this chapter, we introduce a new scheme that deals with the detection of flooding DDoS attacks by constant monitoring of abrupt traffic changes inside ISP network. Two traffic parameters namely, volume and flow are used to detect DDoS attacks. For detecting DDoS attacks, proposed scheme constructs profile of the traffic normally seen in the network, and identifies anomalies whenever traffic goes out of profile. Consideration of varying tolerance factors, as described in section 3.3, make proposed detection scheme adaptable to varying network conditions and attack loads in real time. Different attack scenarios are implemented by varying total number of zombie machines and attack strengths. Proposed scheme has been extensively evaluated through simulation. Detection thresholds and efficiency are justified using receiver operating characteristics (ROC) curve [141]. For validation, KDD 99 [142], a

publicly available benchmark dataset is used. The comparison with existing volume and entropy based approaches clearly indicates the supremacy of our proposed scheme.

Our scheme uses anomaly based detection in ISP domain for detecting flooding DDoS attacks. It interprets flooding DDoS attacks as events that disturb distribution of traffic flows. Here the traffic flow is a set of packets satisfying a 5-tuple (source address, destination address, source port, destination port and protocol type) qualifier, monitored in a polling interval. We model Internet as transit-stub network. The NS-2 network simulator is used as testbed for implementation and evaluation of our approach.

3.2 DDoS Attack Detection Model

3.2.1 Choice of Traffic Parameter

The main factor that governs the effectiveness of a detection model is the parameters used in modeling. In literature, many parameters have been proposed and studied. One of the most obvious parameter of choice is volume and most of existing solutions use volume based metrics [27, 79, 154, 192, 199] to detect DDoS attacks. These suffer in the form of large number of false positives/negatives and hence more collateral damage when attack is carried at slow rate or when volume per attack flow is not so high as compared to legitimate flow. Entropy based approaches [14, 119, 126] can detect low rate degrading attacks, but fail against varied rate attacks wherein intelligent attacker mixes low and high rate zombie machines to generate attack traffic in such a manner that overall entropy remains unchanged.

Lakhina et al [14] observed that most of traffic anomalies despite their diversity share a common characteristic: they induce a change in distributional aspects of packet header fields (i.e. source address, source port, destination address, and destination port etc called traffic features). Our scheme to detect attacks treats DDoS anomalies as events those disturb the distribution of traffic features. For example, a DDoS attack, regardless of its volume, will cause the distribution of the destination address to be concentrated on the victim address. Similarly, a scan for vulnerable port will have a dispersed distribution for destination addresses, and a skewed distribution for destination ports that is concentrated on the vulnerable port being scanned. The key question here is to decide which parameter to be used for measuring distribution of traffic features. We have chosen two parameters namely, volume

and flow together to detect variety of flooding DDoS attacks efficiently. Before presenting illustrations to substantiate our assumptions, let us give formal definition of volume and flow.

Let $X = \{n_i, i = 1, \dots, N\}$ is the frequency distribution consisting of N flows where flow i contains n_i packets. Let $S = \sum_{i=1}^N n_i$ be the total number of incoming packets. Then, volume metric and flow metric can be defined as follows:

Volume metric: Volume metric is used to count the total number of bytes occurring during the polling interval Δ .

$$\text{Volume} = S|_{\Delta} = \sum_{i=1}^N n_i|_{\Delta}$$

Flow metric: Flow metric is used to measure the total number of distinct flows during the polling interval Δ .

$$\text{Flow} = \text{Total number of distinct flows in polling interval } \Delta = N|_{\Delta}$$

The underlying idea is to exploit changes observed in the distribution of addresses or ports under attack, as this is directly proportional to volume and flow, to characterize important traffic anomalies. Two parameters are used together, with the intention that they can detect variety of flooding DDoS attacks efficiently. Whenever incoming attack traffic load is high i.e. in high rate attacks, attack alarm is triggered as volume metric shows the anomaly in the system i.e. total incoming volume is above the threshold. Similarly, when incoming attack traffic load is low per attacking host using a large number of distributed zombies i.e. in low rate attacks, attack alarm is triggered, as flow metric shows the anomaly in the system i.e. total incoming flows are above the threshold. This can be clearly seen from the illustration below:

Figures 3.1 and figure 3.2 show value of volume and flow in subsequent polling intervals during normal and different attack scenarios. Figure 3.1 shows temporal variation of volume measure when system is in normal condition, under low rate DDoS attack and under high rate DDoS attack. DDoS attack starts at 25th second and ends at 50th second. Total 400 client machines are used to send legitimate traffic. High rate attack is performed using 100 zombie machines with mean attack rate 3Mbps per attacker. To perform low rate attack 100 zombie machines are used with mean attack rate 0.1Mbps per attacker. As shown in figure 3.1, it is clear that low rate attacks are nearly undetectable when using only volume measure.

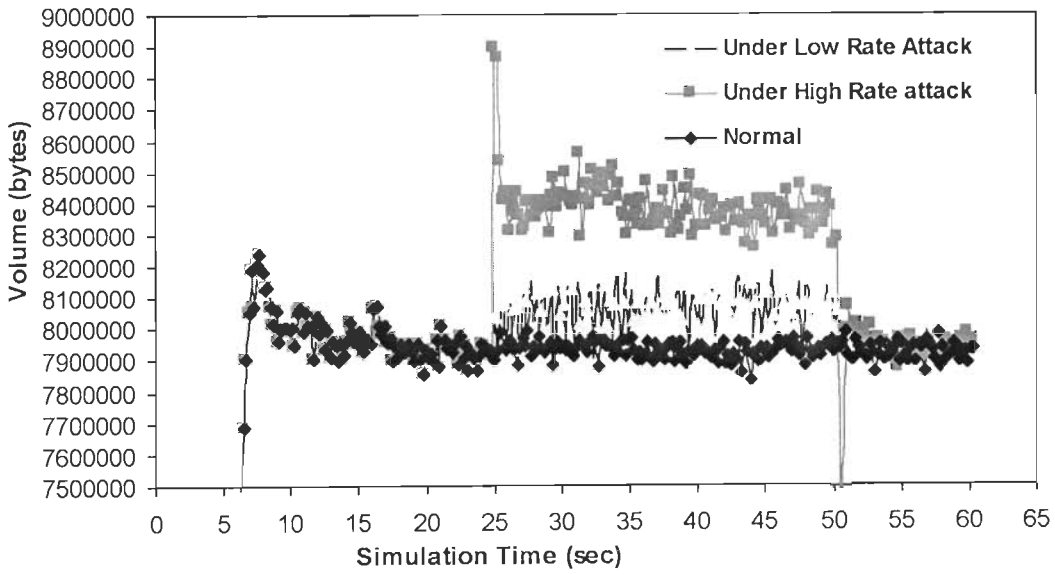


Figure 3.1. Temporal variation of volume measure when system is in normal condition, under low rate DDoS attack, and under high rate DDoS attack

For detection of low rate DDoS attack correctly with low false positive rate, flow measure should also be considered along with volume measure. Figure 3.2 shows temporal variation of flow measure when system is in normal condition, under low rate DDoS attack, and under high rate DDoS attack.

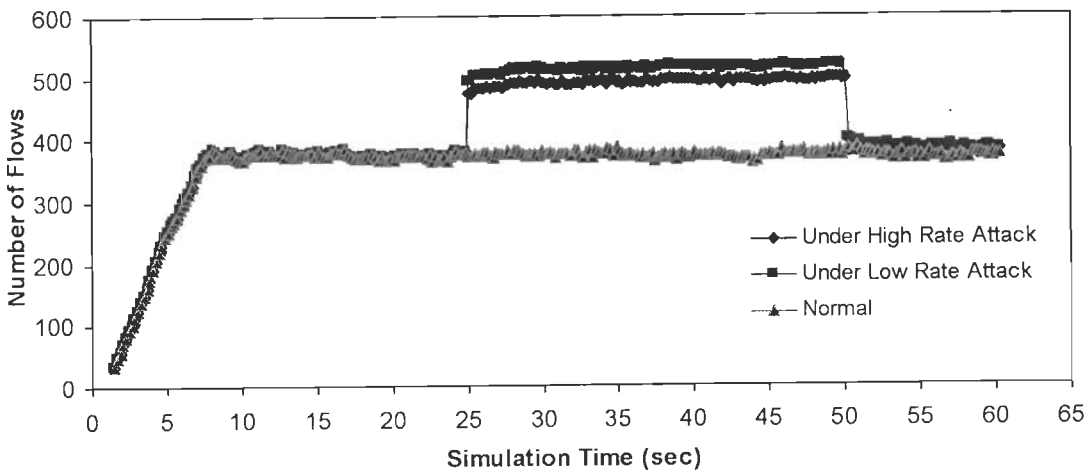


Figure 3.2. Temporal variation of flow measure when system is in normal condition, under low rate DDoS attack, and under high rate DDoS attack

As shown in figure 3.2, low rate attacks can be detected easily when flow measure is used to detect flooding DDoS attack. Therefore, we can conclude that both high rate and low rate flooding DDoS attacks performed using large number of zombie machines can be detected easily when both flow and volume measures are taken together.

3.2.2 Choice of Polling Interval

The choice of the polling interval plays an important role in determining number of false alarm rate and how quickly an attack can be detected. We conducted simulations for various polling intervals and found that false positive alarm number increases steadily with increasing polling interval as shown in figure 3.3. Though false positive rate is minimal at polling interval 100ms but detection rate is also very less i.e. 74 % as shown in figure 3.4. Therefore, in our experiments, optimum value of the polling interval chosen is 200ms.

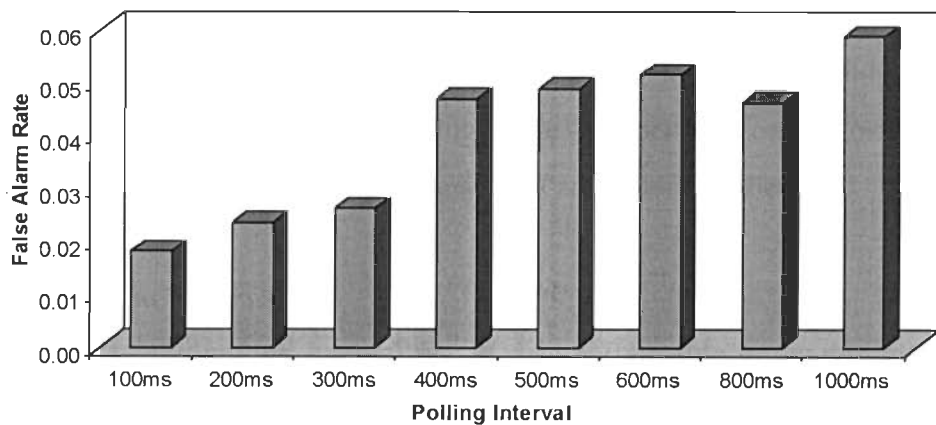


Figure 3.3 Variation of false alarm rate using varying polling intervals

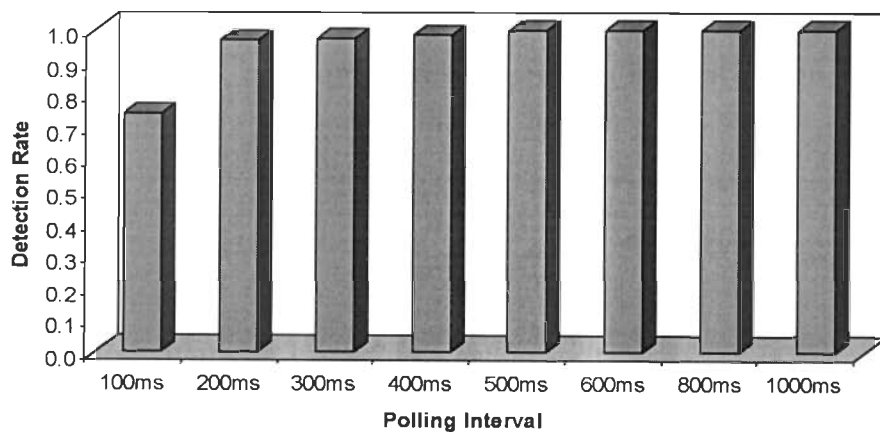


Figure 3.4. Variation of Detection rate using varying polling intervals

Total false positive alarms are minimal with high detection rate using this value of polling interval. The chosen polling interval tends to keep up the balance between both aspects. Moreover a short polling interval also helps in reducing the memory overheads at router as the routers have to store lesser number of packets.

3.3 Flow-Volume based Attack Detection Scheme

In this section, we present our detection scheme to detect an ongoing attack quickly with minimum false positives and negatives.

3.3.1 System Model

Transit-Stub network model [112] of Internet as shown in figure 3.5 is used for simulation. Transit-stub model is based on the hierarchical approach of Internet. Every domain in transit-stub model is classified either as transit network or stub network. Transit network is service provider and interconnects stub networks. Stub network connects end hosts to the Internet. Backbone ISPs and regional ISPs are examples of transit networks. As for the scenario of a DDoS attack, each of the attackers, legitimate users and the victim server are connected to a stub network. Model used for the simulation is a standard one that is used by previous researchers, such as [118]. Traffic flows pass through several stub/ transit domains before reaching to the destination. Monitoring of the traffic directed to protected server is performed at transit router connected to the server. Our aim is to protect the victim server and the corresponding network from DDoS attacks. We model the Internet to measure the volume and flow in transit-stub network. During an attack, the Internet is divided into the two networks; one for inside to be protected and the other is for outside where attackers may reside. Detection system is a part of border router connecting victim or can belong to separate unit that interacts with border router to detect attacks and identify attack traffic. Packets are monitored in a short sized polling interval to minimize memory overheads.

3.3.2 DDoS Detection Scheme

In this section, we will present our proposed detection scheme that may be part of border router connecting victim or can belong to separate unit that interacts with border router to detect attack traffic. The proposed scheme uses a flow-volume based approach (FVBA) to construct profile of the normal traffic seen in the network and identify anomalies whenever

traffic goes out of the normal profile. In FVBA, two statistical measures namely volume and flow are used for profile construction.

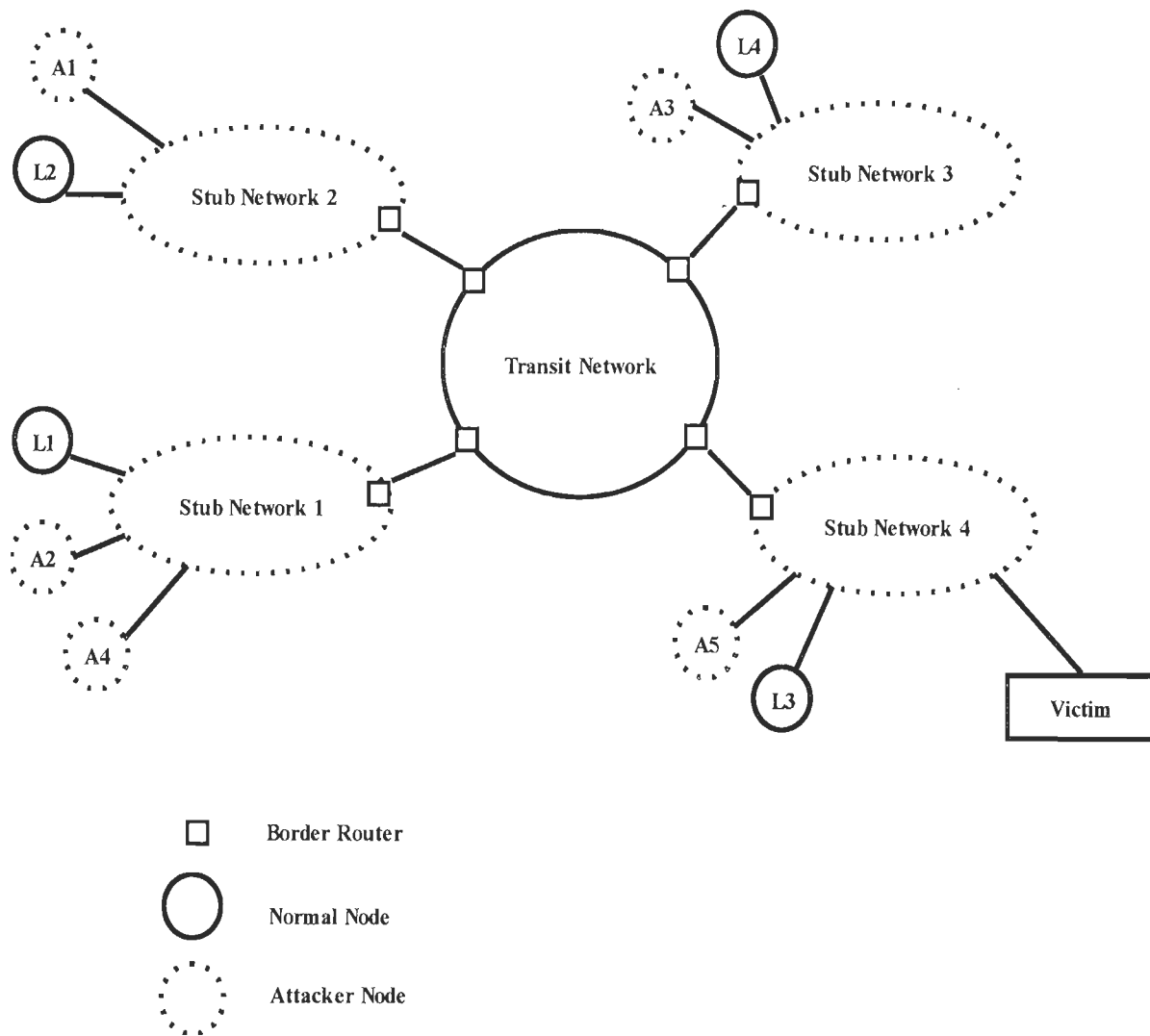


Figure 3.5 Transit-Stub Network Model of Internet

3.3.2.1 Analytical Model

Let $X(t)$ represents the total traffic arriving under normal conditions at the target machine in polling interval $\{t-\Delta, t\}$. $X(t)$ is calculated during polling interval $\{t-\Delta, t\}$ as follows:

$$X(t) = \sum_{i=1}^{N_f} n_i(t), \quad i = 1, 2, \dots, N_f \tag{3.1}$$

where $n_i(t)$ represents total number of bytes arrived for a flow i in $\{t - \Delta, t\}$ polling duration and N_f represents total number of flows. We use total bytes instead of packets to calculate volume measure, because it provides better accuracy, as different flows may contain packets of different sizes. $X(t)$ for a given polling interval $\{t - \Delta, t\}$ can be treated as a random variable and the set of these random variables can be considered as a random process $\{X(t), t = w\Delta, 1 \leq w \leq l, l \in N\}$, where Δ is a constant polling interval and N is the set of positive integers. Variable l is the number of polling intervals. We take first moment of $X(t)$ and designate that as X_n^* , the normal traffic volume.

Total traffic $X_{in}(t)$ at any time in polling interval $\{t - \Delta, t\}$ can be expressed as follows:

$$X_{in}(t) = X(t) + \hat{X}(t), \quad (3.2)$$

where $X(t)$ and $\hat{X}(t)$ are the normal and attack traffic respectively. Similarly we may define another random variable $F(t)$ whose value at time t is the total number of measured flows in polling interval $\{t - \Delta, t\}$ under normal condition. The first moment of random variable $F(t)$ is designated as F_n^* . Total number of flows $F_{in}(t)$ at any time in polling interval $\{t - \Delta, t\}$ can be expressed as follows:

$$F_{in}(t) = F(t) + \hat{F}(t), \quad (3.3)$$

where $F(t)$ and $\hat{F}(t)$ are the normal and attack flows.

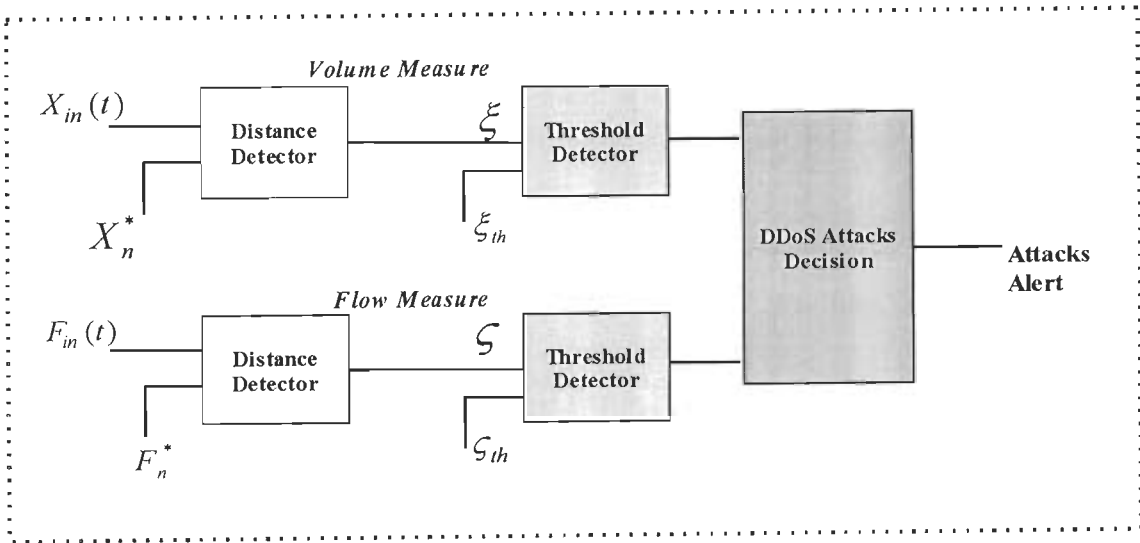


Figure 3.6. FVBA architecture

Figure 3.6 depicts the FVBA architecture. As it is not possible to measure the amount of normal and attack traffics separately in a given polling interval, therefore we use first moments X_n^* and F_n^* for detecting the attack. The values of X_n^* and F_n^* are calculated in advance when system is not under attack.

Algorithm 1: DDoS attacks Detection Algorithm

Input: X_n^* : Normal traffic Volume measure

F_n^* : Normal traffic Flow measure

ξ_{th} : Threshold value for Volume measure

ς_{th} : Threshold value for Flow measure

Output: DDoS attack alert generation.

Procedure:

01: $t = t_0$

02: Measure $X_{in}(t)$ and $F_{in}(t)$ for current polling interval using equation 3.2 and 3.3

03: **If** $((X_{in}(t) - X_n^*) > \xi_{th}) \parallel ((F_{in}(t) - F_n^*) > \varsigma_{th})$ **Then** Attack detected.

 Generate DDoS attack alarm.

04: $t = t + \Delta$

05: Go back to 02

Figure 3.7. DDoS attacks Detection Algorithm

To detect the attack, the value of volume metric $X_{in}(t)$ and flow metric $F_{in}(t)$ are measured in successive polling intervals continuously and whenever there is appreciable deviation from X_n^* and F_n^* , flooding DDoS attacks are detected. The algorithm used for detecting attacks is given in figure 3.7. Threshold values ξ_{th} and ς_{th} are set as follows:

$$\xi_{th} = r_1 * \sigma_V \tag{3.4}$$

$$\varsigma_{th} = r_2 * \sigma_F \tag{3.5}$$

where σ_v and σ_f represent standard deviation for volume measure and flow measure, respectively under normal traffic conditions. $r_1, r_2 \in I$ represent values of tolerance factors for volume and flow measure, respectively, where I is the set of integers.

Tolerance factors are tunable parameters, whose values can be chosen after conducting simulations at different attack strengths. Effectiveness of an anomaly based detection system highly depends on accuracy of threshold value settings. Inaccurate threshold values cause a large number of false positives and false negatives. Therefore, various simulations are performed using different values of tolerance factors.

The choice of tolerance factors vary for different network conditions. Values of tolerance factors also depend on the composition of the normal traffic and the desired degree of the ability to control a DDoS attack. Then, trade-off between detection and false positive rate using ROC curves provides guidelines for selecting values of tolerance factors for a particular simulation environment. In our approach, we use different values of tolerance factors for volume and flow measures, as distribution changes in both volume and flow measures are not necessary same in varying network conditions and attack loads. This makes detection of DDoS attacks more accurate with low false-positives.

3.4 Performance Evaluation

To investigate the effectiveness of the proposed DDoS attack detection scheme, various simulations are carried out for a large number of scenarios. Detailed experimental design and performance analysis are discussed in this section.

3.4.1 Simulation Model

Simulations are carried out using NS2 network simulator on Linux platform to evaluate our proposed detection scheme.

3.4.1.1 System Components

The system consists of the following components:

Clients:- A client is an application or system that accesses a remote service on another computer system, known as a server, through network [23]. For example, web browsers are clients that connect to web servers and retrieve web pages for display. Two types of clients are considered: legitimate clients and attackers. The legitimate clients obey the functionality of TCP protocol, whereas attackers use UDP and therefore do not adhere to the TCP

congestion avoidance protocols. The legitimate clients are modeled as user system running FTP applications. The attackers are modeled by CBR traffic using UDP protocol. A UDP sender does not need to wait for any acknowledgement from the receiver before sending out further packets. This property is apt to model an attacker as an attacker would normally send out large bursts of packets continuously with the aim of flooding the links leading to the server under attack.

Server:- A server is any combination of hardware or software designed to provide services to clients. For example, web servers are servers that provide web services to clients. We assume that the service provided by the server is a generic TCP-based service. The legitimate FTP clients connect to the server with the aim of downloading files, whereas the attackers aim at clogging the bottleneck link leading the server to fail and make the service unavailable to the legitimate clients.

Table 3.1. Simulation parameters

| S. No. | Parameter | Value |
|--------|-------------------------------|---|
| 1. | Simulator | Ns-2 |
| 2. | Traffic arrival process | Poisson |
| 3. | Simulation time | 60 seconds |
| 4. | Attack Duration | 25-50 seconds |
| 5. | Number of legitimate clients | 100-400 |
| 6. | Number of attackers | 10-100 |
| 7. | Polling interval | 200ms |
| 8. | Packet size | 1040 bytes |
| 9. | Tolerance factor α | 1-10 |
| 10. | Connection startup time | 1-8 seconds |
| 11. | Access link bandwidth | 1 Mbps |
| 12. | Backbone link bandwidth | 100Mbps |
| 13. | Backbone link delay | 0 seconds |
| 14. | Bottleneck link bandwidth | 310 Mbps |
| 15. | Mean attack rate per attacker | 0.1-1Mbps (low rate) 2.5-3.5Mbps (high rate) |

Agents:- Agents are software modules deployed at edge routers that runs detection algorithm to declare attacks. These agents receive packets from the clients (legitimate and attackers) that are actually aimed for the server and mark them before sending to the server, if attack is detected.

3.4.1.2 Simulation topology and parameters

The topology considered is similar to the one used traditionally in the Internet for simulation and validation purpose. A simplified view of topology used for the simulation is shown in figure 3.5. Total 400 legitimate client machines are used to generate background traffic. One FTP server is used to provide service to the clients. All FTP requests are originated randomly from different nodes. Total machines generating attack traffic range between 10 to 100.

Table 3.1 shows simulation parameters. Similar parameters are used by previous researchers [118] also. DDoS attacks start at 25th second and end at 50th second. The simulations are carried for different values of tolerance factors r_1 , r_2 and different attack scenarios are created by varying total number of zombie machines and the attack strengths. In our experiments, the polling interval is set to 200ms, as total false positive alarms are minimal with high detection rate using this value of polling interval.

3.4.2 Performance Metrics

For evaluating the performance of our scheme, we used the following performance metrics used in [118]:

1. Detection rate (R_d):- It is given by the following ratio:

$$R_d = d/n \quad (3.6)$$

where d is the number of DDoS attack detected during the simulation experiment and n is the total number of attack generated.

2. False-positive rate (R_{fp}):- It is given by the following ratio:

$$R_{fp} = f/m \quad (3.7)$$

where f is total number of false positive alarm raised by attack detection mechanism, and m is the total number of normal traffic events during the simulation.

3. Goodput:- Goodput is a measure of average rate of successful legitimate traffic transmitted over a communication channel and is calculated as the number of bytes transmitted during specified time interval. We calculate Goodput on the bottleneck link of the transit-stub network [23].
4. NPSR (normal Packet Survival Ratio):- As packets generated by different applications may be of different sizes, we also use NPSR as a performance measure. It is given as the ratio of number of legitimate packets among all packets received during current polling interval.

The ROC (Receiver Operating Characteristic) curve shows the tradeoff between detection rate and false-positive rate.

3.5 Results and Discussion

Results show that false positives and false negatives triggered by our scheme are very less. This implies that profiles built are reasonably accurate and are able to detect variety of DDoS attacks correctly. In following subsections, simulations results are explained.

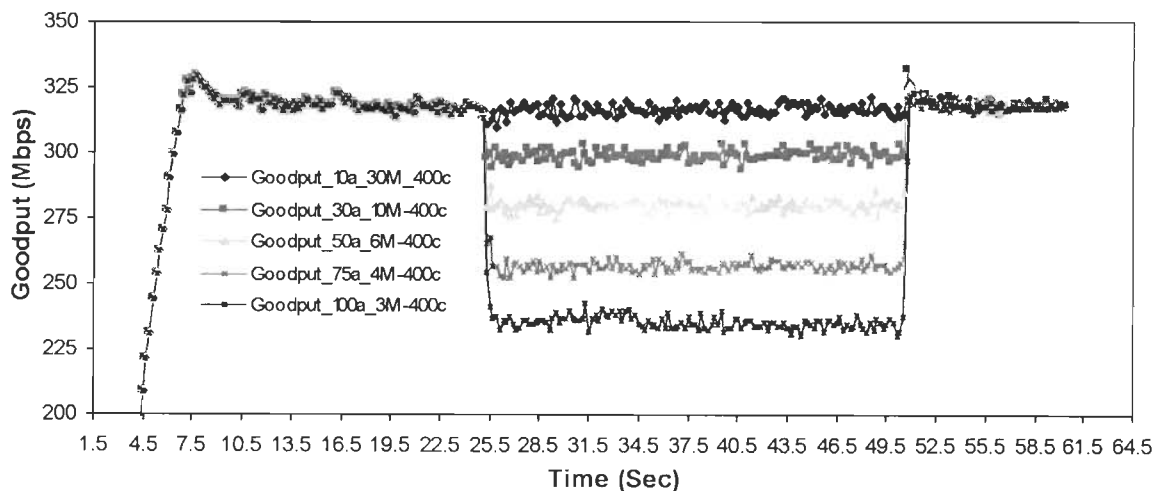


Figure 3.8. Temporal variation of Goodput at different attack strengths

3.5.1 Degradation of Goodput with Attack

The aim of any DDoS attack is to minimize legitimate traffic reaching at the server. Goodput is a measure of legitimate traffic reaching at server. Variation of goodput at different attack strengths is shown in figure 3.8. Here attack is conducted at attack strengths ranging

from 0.1Mbps to 1Mbps per zombie machine. In this experiment, 400 client machines are used to send legitimate traffic to the server while 100 zombie machines are used to send attack traffic. Figure 3.8 shows that, as attack starts at 25 seconds, goodput decreases. At low attack rates, number of attack packet drops is almost negligible, however as attack strength increases number of legitimate as well as attack packet drops also increases.

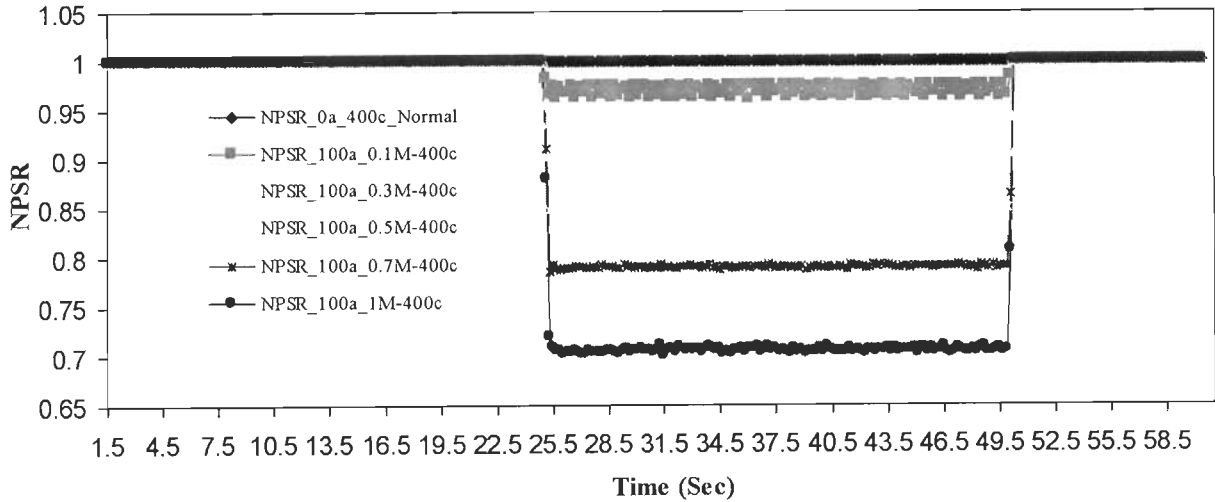


Figure 3.9. Temporal variation of NPSR at different attack strengths

3.5.2 Degradation of NPSR with Attack

In any DDoS attack, increase in attack strength not only decreases goodput but also affects the normal packet survival ratio (NPSR). Figure 3.9 shows temporal variation of NPSR at different attack strengths. The simulation scenarios are same as in section 3.5.1. Similar to Goodput, the NPSR also decreases proportionally as the attack strength increases.

3.5.3 Detection of Attack

As discussed earlier, effectiveness of an anomaly based detection system highly depends on accuracy of threshold value settings. Inaccurate threshold values cause a large number of false positives and false negatives. We use two different tolerance factors r_1 , r_2 for volume and flow metrics, respectively to set the threshold values accurately. Tolerance factors are tunable parameters and depend on network conditions. Thus, it is possible that values of tolerance factors for a particular network environment are not suitable for other network.

Therefore, various simulations are performed using different values of tolerance factors. Then, trade-off between detection and false positive rate provides guidelines for selecting values of tolerance factors for a particular simulation environment.

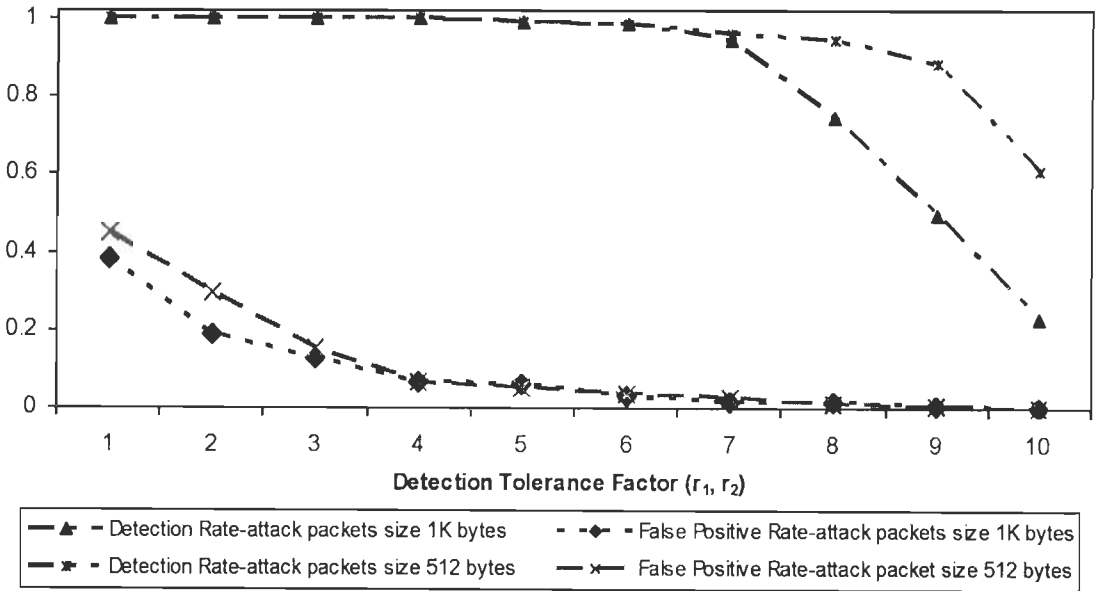


Figure 3.10. Effect of detection tolerance factors on the detection and false positive rate

Figure 3.10 illustrates the variation of the detection and false positive rate with respect to different values of detection tolerance factors r_1, r_2 . Simulation parameters taken are similar to the one explained in section 3.4.1.2. Additionally, two different attack packet sizes, 1K bytes and 512 bytes, are taken in the simulation to test the relation among detection rate, false positive rate, detection tolerance factors and packet size. We can see from the figure 3.10 that as values of tolerance factors increase, detection rate which is nearly 98.8 tend to decrease when $r_1 \geq 6$ and $r_2 \geq 6$. However, false positive rate is very high when $r_1 \leq 5$ and $r_2 \leq 5$.

Therefore, by careful investigations, we select values $r_1=6$ and $r_2=6$, which give detection rate close to 98.8% with less than 3% false positive rate. The ROC curve in figure 3.11 also shows same results.

Therefore, values of both tolerance factors r_1, r_2 are taken as 6 in our approach. Values of r_1, r_2 can vary for different network conditions and correct values can be selected depending on the tradeoff between detection and false positive rate.

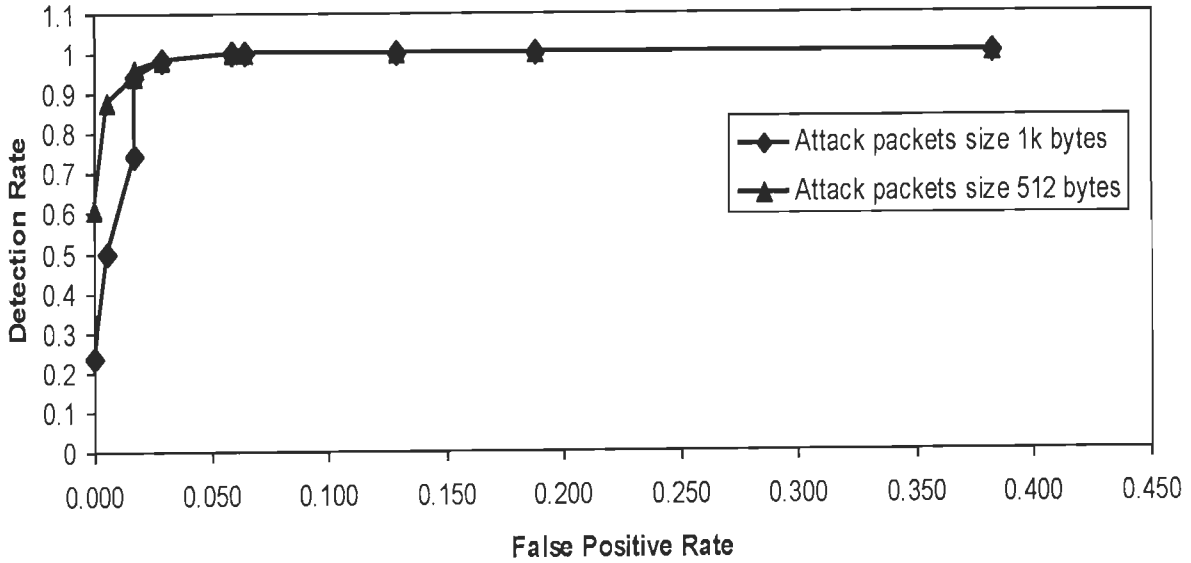


Figure 3.11. ROC curve showing the tradeoff between the detection rate and false positive rate of DDoS attacks

3.5.4 Results with KDD 99 Dataset

In this section, performance of proposed FVBA scheme is evaluated using KDD 99 dataset [142], which is publicly available dataset. Details about KDD 99 dataset are given in appendix-A. First, we filter out connection records of DoS attacks category and then remove labels from both the training and testing dataset. Then normal profile is set for each protocol category as each flow is determined by protocol. To set normal profile, volume and flow measures are calculated using training dataset. In the subsequent sections, training and testing of FVBA scheme are explained and results are displayed.

3.5.4.1 Training

Effectiveness of proposed detection system highly depends on accuracy of threshold value settings. Inaccurate threshold values cause a large number of false positives and false negatives. Therefore, various simulations are performed using different values of tolerance factors r_1 and r_2 . During training, best detection rate is 98.08% with 0.35% false positives, when $r_1 = 1$ and $r_2 = 5$ for TCP connection. For ICMP connection, best detection rate is 100% with 0.78% false positives, when $r_1 = 5$ and $r_2 = 6$. Similarly, for UDP connection, best detection rate is 100% with 0.87% false positives, when $r_1 = 6$ and $r_2 = 8$. Therefore, optimal

values of tolerance factors for different protocols to set the normal profile are given in table 3.2.

Table 3.2. Optimal values of tolerance factors to set normal profile

| Protocol Category | Tolerance Factors value |
|--------------------------|--------------------------------|
| TCP | $r_1=1, r_2=5$ |
| UDP | $r_1=6, r_2=8,$ |
| ICMP | $r_1=5, r_2=6$ |

3.5.4.2 Testing

After training, we apply proposed detection scheme on test dataset. Figure 3.12 summarizes the overall results of testing for different protocol category.

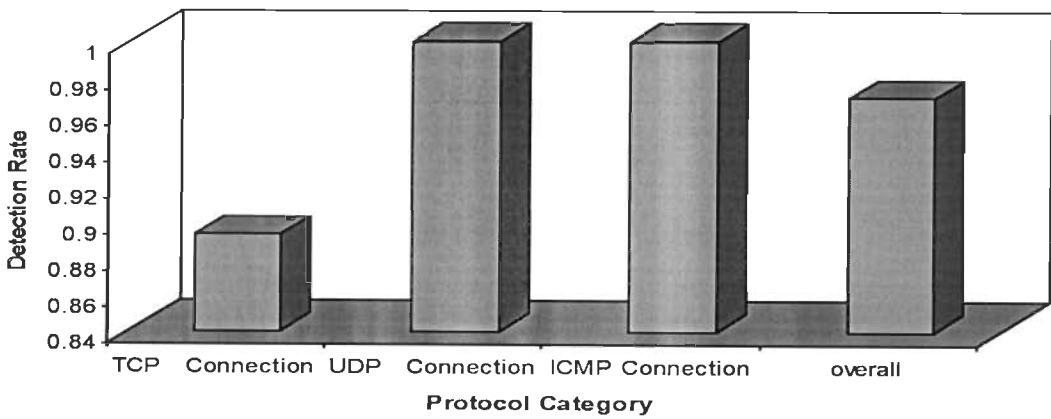


Figure 3.12. Overall detection results on test data for different protocol category

Figure 3.13 contains summary of different types of DoS attacks detected in test dataset. Above stated results show that our proposed approach yields 96.9 percent detection accuracy with less than 1 percent false alarms.

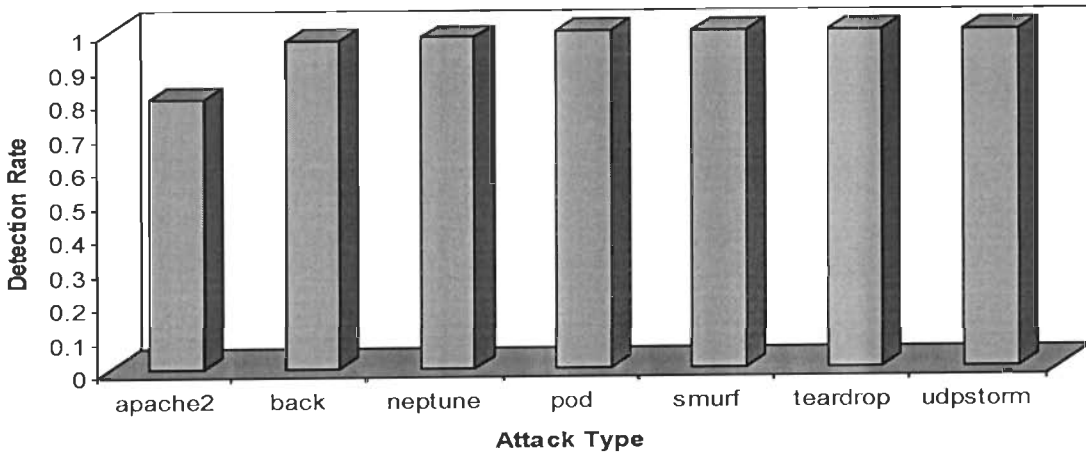


Figure 3.13. DoS Attacks detection summery in test dataset

3.5.5 Comparison with Volume Based Approaches

Comparison of our proposed approach with VBA (Volume Based Approach) [27] is reported below. Same parameters are used for evaluating the performance of both approaches. Following DDoS attack scenarios are taken for comparison:

A. Experiment 1: High Rate Attack

First we studied the effect of varying number of zombies, where each zombie performs attack with high rate (i.e. 3 Mbps).

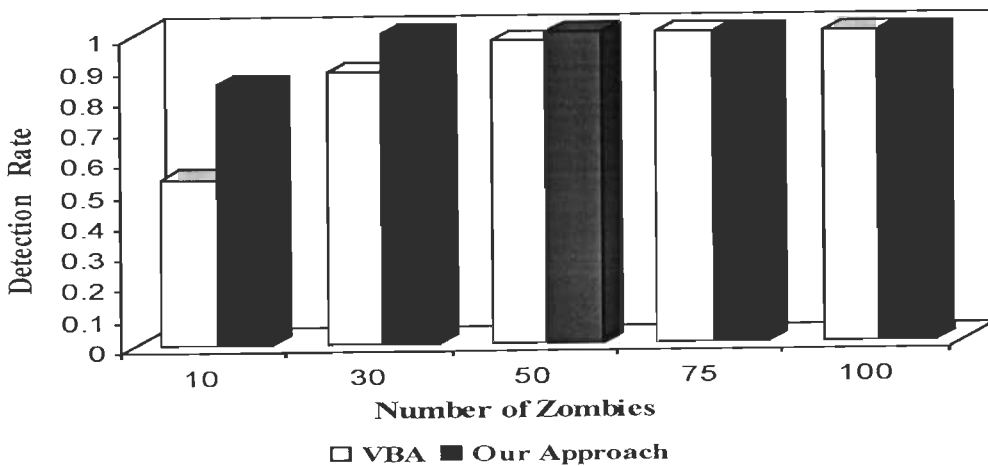


Figure 3.14. Variation of detection rate of VBA and our detection system when attack with high rate is performed by varying number of zombie machines

Figure 3.14 shows the variation in detection rate of VBA and our approach. In this experiment, total number of zombie machines varies from 10 to 100. It can be seen in figure 3.14 that detection rates are comparable when total number of zombie machines are more, but when the total number of zombie machines are less, our proposed approach provide better detection rate compared to volume based approach.

B. Experiment 2: Low Rate Attack

In this experiment, we studied the effect of varying number of zombies when each zombie performs attack with low rate (i.e. 0.1Mbps). Figure 3.15 shows the variation of detection rate of VBA and our scheme. It is clear from figure 3.15 that our detection system’s performance is far better than VBA.

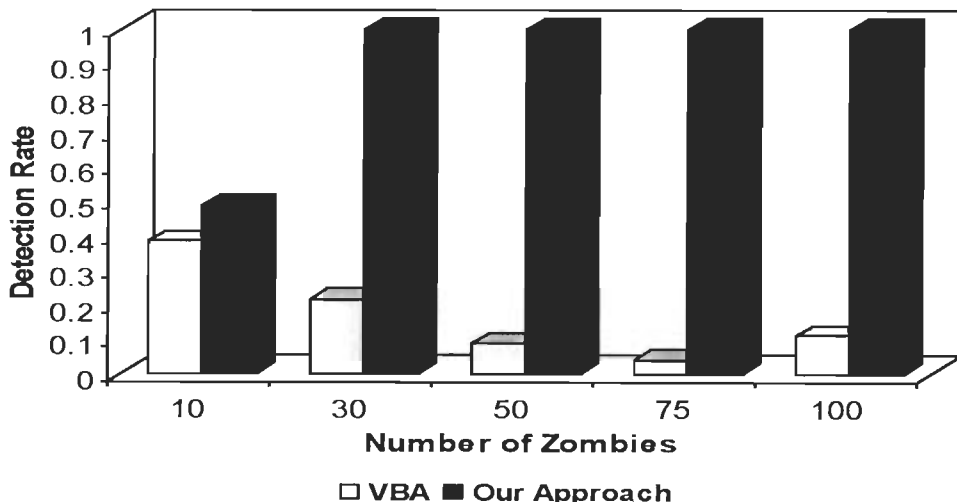


Figure 3.15. Variation of detection rate of VBA and our detection system when attack with low rate is performed by varying number of zombie machines

This is mainly due to the fact that in case of low rate degrading attacks, the total attack traffic does not exceed even normal fluctuations. But as we have considered total number of flows too, low rate degrading attacks are easily detected by our approach.

C. Experiment 3: Mixed Rate Attack

In this experiment, total number of zombie machines remain fixed i.e. 100 but the attack rate is varied to degrade performance of server machine. Figure 3.16 shows that our

detection system's performance is far better than VBA when attack strength is low. This is mainly because total arrived attack traffic does not exceed even normal fluctuation.

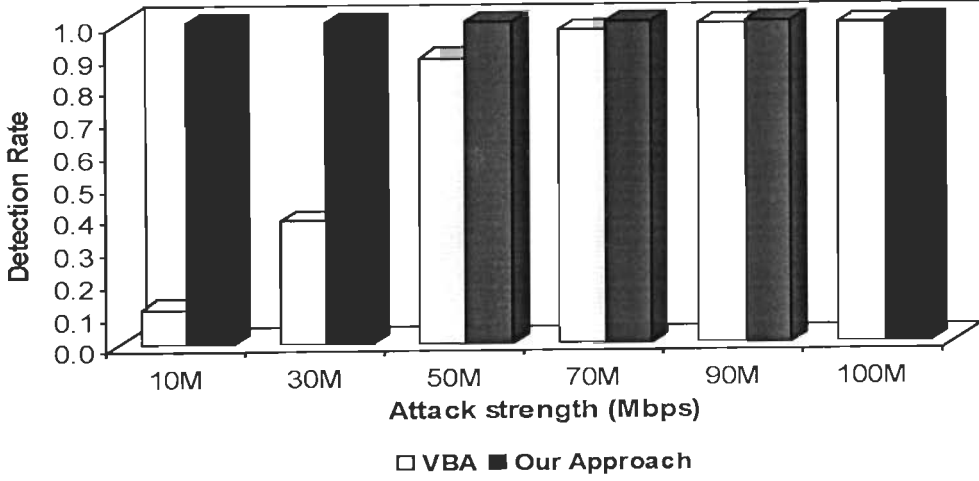


Figure 3.16. Variation of detection rate of VBA and our detection system when attack with varying attack rate is performed using hundred zombie machines

3.5.6 Comparison with Entropy Based Approaches

Comparison of detection performance of our proposed approach with entropy based DDoS attack detection systems [14, 119, 126] are reported below.

Carefully mixing of low and high rate zombie machines by attacker can make DDoS attack undetectable using entropy based approach. Let us assume that during polling interval Δ following flow arrive at the server:

- A. $(f_1^G, f_2^G, \dots, f_i^G)$ i genuine flows carrying $(n_1^G, n_2^G, \dots, n_i^G)$ bytes, respectively.
- B. $(f_1^H, f_2^H, \dots, f_j^H)$ j high rate attack flows carrying $(n_1^H, n_2^H, \dots, n_j^H)$ bytes, respectively.
- C. $(f_1^L, f_2^L, \dots, f_k^L)$ k low rate attack flows carrying $(n_1^L, n_2^L, \dots, n_k^L)$ bytes, respectively.

Then, total genuine, high rate attack and low rate attack traffic coming in Δ time duration can be calculated as follows:

$$X^G(t) = \sum_{i=1}^{N^G(t)} n_i^G \quad (3.8)$$

$$X^H(t) = \sum_{j=1}^{N_a^H(t)} n_j^H \quad (3.9)$$

$$X^L(t) = \sum_{k=1}^{N_a^L(t)} n_k^L \quad (3.10)$$

where, $X^G(t)$, $X^H(t)$ and $X^L(t)$ represent total genuine, high rate attack and low rate attack traffic respectively, coming during Δ time duration. Total traffic coming during Δ time duration, represented by $X^{Total}(t)$ will be as follows:

$$X^{Total}(t) = X^G(t) + X^H(t) + X^L(t) \quad (3.11)$$

During normal condition when system is attack free, value of entropy is:

$$H^{Normal}(t) = - \sum_{i=1}^{N_a^G(t)} f_i^G / X^G(t) \log_2(f_i^G / X^G(t)) \quad (3.12)$$

Value of entropy during varied rate attack is:

$$H^{Attack}(t) = - \sum_{i=1}^{N_a^G(t)} f_i^G / X^{Total}(t) \log_2(f_i^G / X^{Total}(t)) - \sum_{j=1}^{N_a^H(t)} f_j^H / X^{Total}(t) \log_2(f_j^H / X^{Total}(t)) - \sum_{k=1}^{N_a^L(t)} f_k^L / X^{Total}(t) \log_2(f_k^L / X^{Total}(t)) \quad (3.13)$$

In equation (3.12) and (3.13) $H^{Normal}(t)$ and $H^{Attack}(t)$ represent values of entropy when there is no attack and when system is under attack, respectively. By using some sophistic attack tools, intelligent attacker can mix low and high rate zombie machine in such a manner that overall entropy remains unchanged. i.e. $H^{Normal}(t) \approx H^{Attack}(t)$. If it is so, detection systems will fail to detect an ongoing attack. The below shows some cases in which entropy based detection system will not able to detect flooding attacks.

$$\text{Ex. I. } i=j=k=5, \quad n_1^G = n_2^G = n_3^G = n_4^G = n_5^G = 50, \quad n_1^H = n_2^H = n_3^H = n_4^H = n_5^H = 1000000, \\ n_1^L = n_2^L = n_3^L = n_4^L = n_5^L = 20$$

$$\text{Ex. II. } i=j=k=5, \quad n_1^G = n_2^G = n_3^G = n_4^G = n_5^G = 1000, \quad n_1^H = n_2^H = n_3^H = n_4^H = n_5^H = 5000000, \\ n_1^L = n_2^L = n_3^L = n_4^L = n_5^L = 20.$$

3.6 Chapter Summary

In this chapter, a new approach is proposed that accurately detects a wide range of DDoS attacks, ensuring good service to legitimate clients. Two metrics, volume and flow have been used in parallel and an analytical model has been constructed for detecting variety of flooding DDoS attacks. Consideration of varying tolerance factors make proposed detection system scalable to the varying network conditions and attack loads in real time.

In addition to controlled test-bed experiments, effectiveness of the proposed scheme is verified through intensive experiments with KDD 99 dataset. Proposed system has demonstrated an excellent performance in both test-bed experiments and in the real operation. It is found that combining flow and volume measures is a better way to find signs of attack as compared to volume or entropy measure alone. Entropy based schemes have also been proposed for detecting DDoS attacks. However to the best of our knowledge [118], maximum detection rate achieved using these approaches is close to 98% whereas our FVBA based scheme shows a detection rate upto 98.8%. Performance of proposed scheme is compared with existing volume based approach. The results show that proposed scheme gives 10-30% improvement in detection rate over earlier volume based schemes. We have implemented our approach in single ISP network but it can be easily deployed at multiple ISPs with help of trusted entities acting as interfaces between two ISPs so that two ISPs can share there information and thus more effectively stop the attack.

CHAPTER 4

DETECTING DISTRIBUTED DENIAL OF SERVICE ATTACKS USING GARCH MODEL

4.1 Introduction

In chapter 3, we presented flow-volume based approach (FVBA) to detect flooding DDoS attacks. Though the FVBA approach performs better than previous methods, further improvements can be made if statistical properties of DDoS attack traffic are considered. In line with this, some authors have tried to use linear time series models [6, 17, 32] for DDoS attack detection. However, most network applications generate bursty traffic having long tailed distribution. For this type of traffic, these models are not suitable as they can not capture long range dependence (LRD). To solve this problem, nonlinear statistical models can be used for fast and effective detection of flooding DDoS attacks. One of the most popular statistical nonlinear time series modeling technique is the Generalized Autoregressive Conditional Heteroskedastic (GARCH) model [189] which is used for detecting TCP SYN flooding and RESET/FIN attacks in [146]. In this chapter, we show that the same technique can be effectively used for detecting flooding DDoS attacks also by appropriately selecting traffic parameters to generate the time series. From our study, it is found that GARCH model when used with entropy based time series performs better than linear prediction model described in [105]. Proposed scheme detects flooding DDoS attacks with high detection rate.

4.2 Time Series Analysis

Time series analysis deals with data values that are collected over time [115, 165]. The time order of data is important. A major task in time series analysis is to uncover the probability law that governs the observed time series and the main objectives in such analysis are to understand the underlying dynamics, forecast future events and control future events via interventions. There are two types of time series depending up on how they are expressed mathematically. A time series that is expressible as the output of a linear model is called a linear time series. In contrast, the output from a nonlinear model is called a nonlinear time series. To see the difference in mathematical form of the linear and nonlinear time series

models briefly, consider a time series x_t observed at equally spaced intervals. Let us denote the observations by $x_t | t = 1, \dots, T$ where T is the sample size. Then x_t is said to be linear if it is expressible as

$$x_t = \mu + \sum_{i=0}^{\infty} a_i \varepsilon_{t-i} \quad (4.1)$$

where μ is a constant, a_i are constants and $\{\varepsilon_t\}$ is an independent and identically distributed random number. On the other hand, nonlinear time models are expressed as

$$x_t = g(F_{t-1}) + \sqrt{h(F_{t-1})} \varepsilon_t / \sigma_t \quad (4.2)$$

where F_{t-1} is dependent on available information of variance σ at time $t-1$, g and h are nonlinear functions.

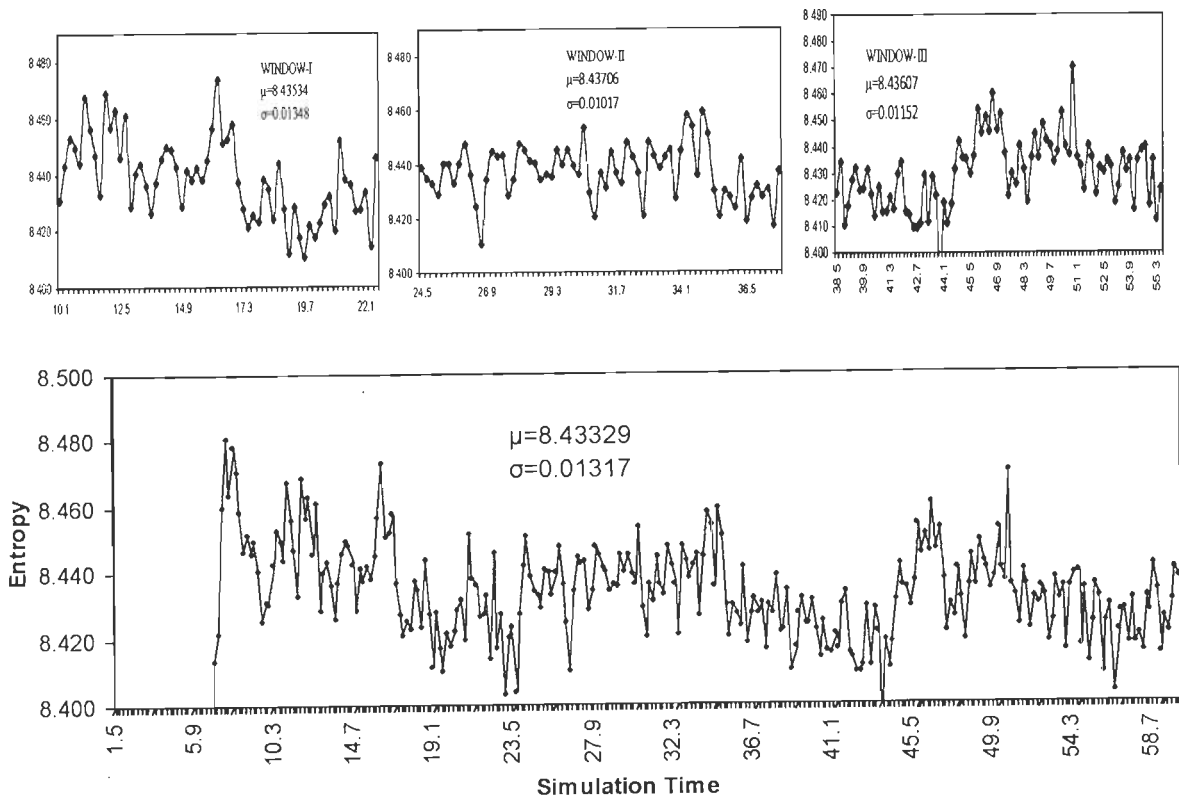


Figure 4.1. Evidence of stationarity

In this chapter, we will focus on nonlinear time series and nonlinear models. Before we present the details of time series analysis and the various types of time series models, we will give a brief account of the important property of time series: stationarity.

4.2.1 Stationarity

A time series can be stationary or non-stationary. Before doing any time series analysis, it is important to know about the stationarity of the time series as almost all time series models require the time series to satisfy this property [115]. We therefore, first check about the stationarity of our data set. A time series $\{x_t\}$ is said to be stationary if the joint distribution of $(x_{t_1}, \dots, x_{t_k})$ is identical to that of $(x_{t_1+t}, \dots, x_{t_k+t})$ for all t , where k is an arbitrary positive integer and (t_1, \dots, t_k) is a collection of k positive integers [165]. In a stationary time series, the statistical properties such as mean and variance are constant. On the other hand, if the stochastic structure of a time series itself changes over time, it is called a nonstationary time series.

Knowing the stationarity property of a time series data plays a major role in determining the future development of the series. Figure 4.1 shows stationarity of our network data set. It shows a snapshot of the data set (entropy of incoming traffic vs simulation time) and three windows of it at different time interval. It is seen that the mean (μ) and standard deviation (σ) is almost constant throughout the different window signifying the fact that the data is stationary.

4.2.2 Autocorrelation

Another important property of time series is the relation between the random data values at different time points. This relation is expressed by what is known as autocorrelation. For a stochastic process $\{x_t\}$ which can take the form of discrete time series $x_t, t = 0, 1, 2, \dots, N$, the autocorrelation coefficient measures the linear dependence between x_t and x_{t+k} . For a stationary time series, the autocorrelation coefficient is constant. Mathematically, the autocorrelation coefficient is given as [165]

$$\rho(k) = \frac{E[(x_t - \mu)(x_{t+k} - \mu)]}{\sigma^2} \quad (4.3)$$

where E is the expected value and k is the time shift being considered (usually referred to as the lag). This function has the attractive property of being in the range $[-1, 1]$ with 1 indicating perfect correlation (the signals exactly overlap when time shifted by k) and -1 indicating perfect anti-correlation. Depending up on the value of k , i.e. the time gap between the data points used for the calculation of autocorrelation, there are two types of dependences:

short range dependence (SRD) and long range dependence (LRD). SRD is a phenomenon in which the coupling between different times decreases rapidly as the time difference increases. On the other hand, LRD is a phenomenon in which the coupling between different times decreases slowly as the time difference increases and shows much stronger coupling. All short-range dependent processes are characterized by an autocorrelation function which decays exponentially fast; processes with long-range dependence exhibit a much slower decay of the correlations.

4.3 Non-linear Time Series Modeling

One important objective in time series analysis is modeling of the time series. In time series modeling, we capture the stochastic structure of time series by identifying an appropriate model. Since there are various types of time series, it is necessary to select an adequate model class and to estimate parameters included in the model, depending on the characteristics of the time series and the objective of the time series analysis. There are two important classes of models, namely linear and nonlinear model. Linear models have been used for long time and are very popular. An interested reader may refer well documented books like [115, 165]. However, linear models have certain limitations in modeling nonlinear time series due to their basic assumption. In linear time series modeling, it is assumed that the difference between actual value and predicted value follow a Gaussian white noise distribution with mean zero and constant variance. However, this is not the case in many practical situations like network traffic. Network traffic generally has leptokurtic distribution with long tail instead of normal distribution. For checking leptokurtic distribution of data, hetroskedastic properties, size of window and for finding order of GARCH model, authors in [146] have used certain tests; similar tests have been performed on our dataset consisting of entropy based time series. We computed kurtosis (using histogram plot) using our dataset (entropy of incoming traffic per simulation time). From figure 4.2, it is clearly seen that it is skewed, with kurtosis being 31.28.

To strengthen this approach, below we apply standard tests to check the hetroskedasticity of our network traffic data. A synthetic data is used for the test and the abnormal traffic data is generated with different attack rates starting from 10Mbps to 100Mbps. For each time slot, entropy of incoming packets is calculated as explained in section 4.6.1. Since the data is simulated, it can be labeled, that is the exact time at which

attack happen is known a priori. Hence the values in the datasets constitute entropy of incoming traffic over 200ms intervals.

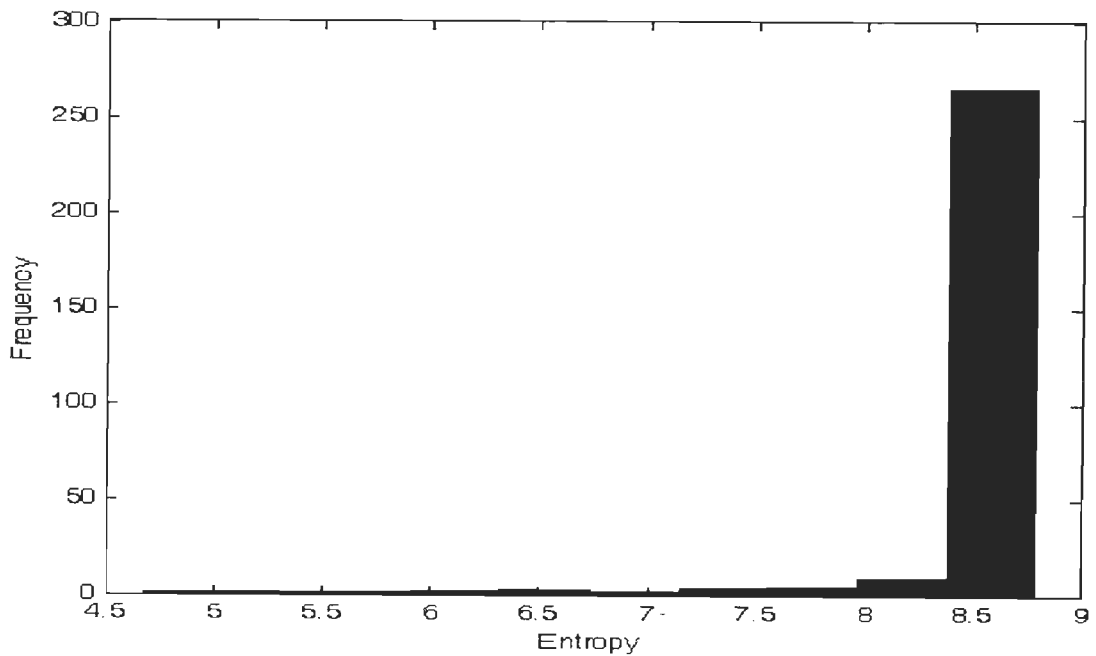


Figure 4.2. Kurtosis for the sample dataset

4.4 GARCH Model

GARCH is a type of nonlinear time series model which is concerned with the evolution of the square of conditional variance. It uses an exact function to describe the manner in which the square of the conditional mean evolves over time [189]. The name GARCH stands for Generalized Autoregressive Conditional Heteroskedasticity. The term heteroskedasticity refers to the time-varying variance or volatility of the time series considered. Conditional implies a dependence on the observations of the immediate past. Autoregressive describes a feedback mechanism that incorporates past observation into present. The formal definition of the model is given below.

Definition: A generalized autoregressive conditional heteroskedasticity (GARCH) model of order $(p>0)$ and $(q>0)$ is defined as [115, 165]

$$X_t = \sigma_t \varepsilon_t \text{ and } \sigma_t^2 = c_0 + \sum_{i=1}^p b_i x_{t-i}^2 + \sum_{j=1}^q a_j \sigma_{t-j}^2 \quad (4.4)$$

where $c_0 \geq 0$, $b_i \geq 0$, and $a_j \geq 0$ are constants, $\{\varepsilon_t\}$ is independent and identically distributed with mean 0 and variance 1, and ε_t is independent of $\{x_{t-k}, k \geq 1\}$ for all t . A stochastic process $\{x_t\}$ defined by the equations above is called a GARCH (p, q) process. These conditions on the coefficients 'a' and 'b' ensure that the conditional variance is always positive.

Unlike its predecessor, the Autoregressive Conditional Heteroskedasticity (ARCH process), which takes the weighted average of the square of past observations only as an estimate of the square of the variance, the GARCH model given above takes the average of not only square of past observations x^2 but also square of past conditional variance σ^2 's. This explains why simple GARCH models, such as GARCH(1, 1), may provide a cost effective representation for some complex autodependence structure of $\{x_t^2\}$, that can only be accommodated by an ARCH(p) model with large p. In fact, the GARCH(1, 1) model has been tremendously successful in empirical work and is regarded as the benchmark model by many econometricians.

The necessary and sufficient condition for the above model to define a unique and stationary process is that

$$\sum_{i=1}^q a_i + \sum_{j=1}^p b_j < 1 \quad (4.5)$$

4.5. Test for Heteroskedasticity

Modeling conditional heteroskedasticity amounts to augmenting a dynamic equation, which governs the time evolution of the conditional variance of a time series data, to a time series model. The first step in building such a nonlinear time series model is to test for conditional heteroskedasticity which is also known as ARCH effect. There are various standard experimental analyses that could be performed to test conditional heteroskedasticity of a collected data. The most important ones are the Engle's ARCH test and the Ljung-Box-Pierce Q-test. These two tests show that there is sufficient evidence to confirm that network data is heteroskedastic in nature. We also perform these tests on our data set and our results show

that network traffic is heteroskedastic in nature. Because of this, we use GARCH as the model of choice. Below we will describe these two tests and our test results.

4.5.1 Engle's ARCH Test

We used `archtest` function in MATLAB for Engle's ARCH test [4] to test the heteroskedasticity of our data set. Given sample residuals obtained from a curve fit (for example, a regression model), ARCH-test check for the presence of M^{th} order ARCH effect. This is to mean that it tests if the first M lags of the ACF of x_t^2 are zero or not. It does so by regressing the squared residuals on a constant and the lagged values of the previous M squared residuals. In our experiments, MATLAB commands are used to performing Engle's ARCH test and result shows significant evidence in support of GARCH effects (i.e heteroskedasticity). Table 4.1 is a snapshot of the ARCH-test for our data set.

In Table 4.1, H is a boolean decision vector for the hypothesis. In the Engle's test, the null hypothesis is that the first M lags of the ACF are zero, i.e. the first M a_i coefficients of equation 4.4 are zero. Accordingly in table 4.1, $H=0$ indicates acceptance of the null hypothesis that no ARCH effect exist while $H=1$ indicates rejection of the null hypothesis. The field "pValue" signifies a vector of P-values (significance levels) at which ARCH-test rejects the null hypothesis of no ARCH effects at each lag. "ARCHstat" denotes a vector of ARCH test statistics for each input lag. "CriticalValue" signifies a vector of critical values of the chi-square distribution for comparison with the corresponding element of "ARCHstat". The values are computed for three different lags of 10, 15 and 20. The value of $H=1$ clearly indicates the existence of ARCH effects on the given data set.

Table 4.1. Engle ARCH Test

| Lags | H | pValue | ARCHstat | CriticalValue |
|------|---|--------|----------|---------------|
| 10 | 1 | 0 | 283.1350 | 15.9872 |
| 15 | 1 | 0 | 274.4596 | 22.3071 |
| 20 | 1 | 0 | 256.4146 | 28.4120 |

4.5.2 Ljung-Box-Pierce Q-Test

Ljung-Box-Pierce Q-Test [4] is a measure for the departure from randomness based on the ACF of the data. Using LBQ-test, we can verify at least approximately, that no significant correlation is present in the data sets when tested for 10, 15, and 20 lags of the ACF at the 0.05 level of significance. Table 4.2 is a snapshot of what the LBQ-test-test gives the data. Similar to the ARCH test, H is the Boolean decision variable for the hypothesis to be tested, pValue is a vector of P-values (significance levels). The last two columns of the table 4.2 also have same meaning to that of the Engle's test.

Table 4.2. Ljung-Box-Pierce Q-Test

| Lag | H | pValue | ARCHstat | CriticalValue |
|-----|---|--------|----------|---------------|
| 10 | 1 | 0 | 453.1352 | 18.3070 |
| 15 | 1 | 0 | 457.7174 | 24.9958 |
| 20 | 1 | 0 | 457.8511 | 31.4104 |

From the above tests, it can be concluded that network exhibits indeed heteroskedasticity. Hence it does not fit the normal distribution for the residual while modeling this data set. Therefore, for such types of time series data, nonlinear models like ARCH/GARCH model which take into account dependency of second order moments are effective. This is the underlying fact for our assumption to use GARCH model for DDoS attack detection.

4.6 GARCH Model based DDoS Attack Detection

4.6.1 Choice of Parameter for Modeling Flooding Attacks

An attacker can generate a flooding attack using many different type of protocols. Therefore, if our objective is to detect a variety of flooding attacks, the properties of a particular protocol can not be used for detecting the attack and we have to depend on the properties of the incoming traffic for detecting the attack. Therefore, in this work, we have

chosen entropy as a metric to model the flooding attacks. A metric that captures the degree of dispersal or concentration of a distribution is called sample entropy. Sample entropy $H(X)$ [118] is calculated using the following formula:

$$H(X) = -\sum_{i=1}^N p_i \log_2(p_i) \quad (4.6)$$

where p_i is n_i/S . Here n_i represent total number of bytes arrivals for a flow i in duration $\{t - \Delta, t\}$ and N_f represents total number of flows. $S = \sum_i^{N_f} n_i, i = 1, 2, \dots, N_f$. The value of sample entropy lies in the range $0 - \log_2 N_f$.

4.6.1.1 Choice of polling interval

Polling interval of 200ms is used in this work as this result in lowest false positive and highest detection rate. The results of our experiments and detailed discussion for the choice of this pooling interval are given in chapter 3.

4.6.1.2 Choice of the order of model and size of window

To decide order of model in statistical modeling, many criteria are used. Akaike Information Criteria (AIC) developed by Hirotugu Akaike, [1] is most commonly used information criterion. It was originally proposed for time-series models, but has also been used in regression. AIC provides a measure of model quality by simulating the situation where the model is tested on a different data set. After computing several different models, you can compare them using this criterion. According to Akaike's theory, the most accurate model has the smallest AIC. Akaike's Information Criterion (AIC) is defined by the following equation:

$$AIC = -2 \log p(L) + 2p \quad (4.7)$$

where L refers to likelihood under fitted model and p is the number of parameters in the model. In addition to AIC, another important information criterion that is also widely used is Bayesian Information Criteria (BIC) [2]. BIC is developed by Gideon E. Schwarz. BIC is a criterion for model selection among a class of parametric models with different numbers of parameters. Bayesian Information Criteria (BIC) is defined by the following equation:

$$BIC = -2 \log p(L) + p \log(n) \quad (4.8)$$

where L refers to likelihood under fitted model and p is the number of parameters in the model. In the equation 4.8, n is used to represents the sample size. In our experiments, we use both AIC and BIC to decide order of GARCH model. Table 4.3 shows the results of our test. From the table, we can see that GARCH(1,1) fits better than any other order. In most of the cases GARCH(1,1) performs better than other orders of the model. When the family of competing models are compared to the GARCH(1,1) model, it can be inferred that none of the competing models are better than the GARCH(1,1) [150]. Also, it is believed that according to the principle of parsimony, “lesser the parameters to estimate, lesser can we go wrong” [5]. Hence GARCH (1,1) is chosen to model the network traffic. Hence, we model the network traffic data using GARCH(1,1).

Table 4.3. Prediction errors using GARCH(P,Q) for various value of P and Q

| Framesize | P=1, Q=1 | P=2, Q=1 | P=1, Q=2 | P=2, Q=2 |
|-----------|----------|----------|----------|----------|
| 10 | 0.03810 | 0.0390 | 0.0387 | 0.0391 |
| 20 | 0.0203 | 0.0204 | 0.0215 | 0.0200 |
| 30 | 0.0160 | 0.0185 | 0.0173 | 0.0165 |
| 40 | 0.0164 | 0.0180 | 0.0163 | 0.0171 |

For predicting attacks, in our experiments, we divided the dataset into frames containing $N=40$ samples. Adjacent frames are separated by M (in our experiment $M=5$) samples so they overlap with each other by $N-M$. For example, if the first frame contains samples s_i to s_{i+40} , next frame starts from sample s_{i+5} and contains samples up to s_{i+45} . We then computed GARCH coefficients for each of the overlapping frames in the dataset. After the coefficients are computed, we can predict the variance of entropy of the series [124]. The error between the actual variance of entropy (σ_a) and the predicted variance of entropy (σ_t) is given by:

$$e_t = (\sigma_a - \sigma_t) / \sigma_a \quad (4.9)$$

where the actual variance of entropy is given by:

$$\sigma_a^2 = (x_t - X)^2 \quad (4.10)$$

where x_t is the entropy value at time t and X is the mean value of entropy taken for frame size.

For the selection of optimal window size, we performed a comparison of the prediction error values with frame size starting from 10 to 40. Table 4.3 shows the result of the comparison and we can see that there is a gradual decrease in the error with increase in frame size. However as evident from the table 4.3, the decrease in error from frame size 30 to 40 is not much. Beyond this, there is only a slight decrease in error values as we increase the frame size further. Therefore, we selected a frame size of $N=40$ to achieve an appreciable number of frames with the total data set size under consideration. As stated above, we model the network traffic data using GARCH (1,1) because of its versatility [123].

4.6.2 Detection Algorithm

The detection algorithm performs time series analysis on the input network traffic to detect any flooding attack. It computes GARCH(1,1) coefficients and predict the variance of entropy. The data set is divided into overlapping frames of size N .

In this detection algorithm, frames containing N consecutive samples are given as input, where each sample corresponds to one polling interval. We assume α to be the threshold value on the error. For detecting flooding DDoS attacks, we have used same steps as given in [146] with different frame size and frame overlapping on entropy based time series generated by us. These steps are given below.

- for each polling interval
 - Calculate volume in each flow
 - Compute the entropy using equation 4.6
- Initialize $i=0$, $M=5$, $N=40$.
- L1: Construct frame for entropy values x_i to x_{i+N}
 - Compute GARCH(1,1) coefficients for the frame using equation 4.4

- Compute next step actual variance
- Compute next step predicted variance
- Compute the prediction error e_t using equation 4.9 & 4.10
- Calculate average prediction error ε using e_t , e_{t-1} and e_{t-2}
- If $\varepsilon > \alpha$ Then
 - Raise alarm informing a possible attack
 - Discard these attack samples
- Else
 - $i = i + M$
- End If
- Go to L1 if not end of data

4.7 Performance Evaluation

4.7.1 Experimental Setup

The topology and simulation parameters discussed in chapter 3 are used again for these experiments given in this chapter. The simulations are repeated and the results obtained using GARCH model are compared with linear prediction for different attack scenarios generated by varying attack strengths at fixed total number of zombie machines.

4.7.2 Performance Metrics

For evaluating the performance of this approach, we used the detection and false positive rate performance metrics discussed in chapter 3.

4.7.3 Prediction Error

A very important measure in evaluating the performance of GARCH model is prediction error. In our model, we calculate the prediction error and compare it with LP model. The error increases from normal to anomalous frames. It settles down to acceptable values when normal behavior resumes. When we compare the prediction error value of GARCH(1,1) with linear prediction of order 4, for the same data sets and frame size, it is observed that the error values using GARCH(1,1) is lower than that of the LP. Table 4.4

shows the mean prediction error values using GARCH(1,1) and LP models. From the table 4.4, we can clearly conclude that GARCH(1,1) fits the data well and shows lower prediction error than LP. Figure 4.3 shows a plot comparing GARCH(1,1) and LP. Although both model shows almost a linear increase, the prediction error for the LP is higher that of GARCH(1,1).

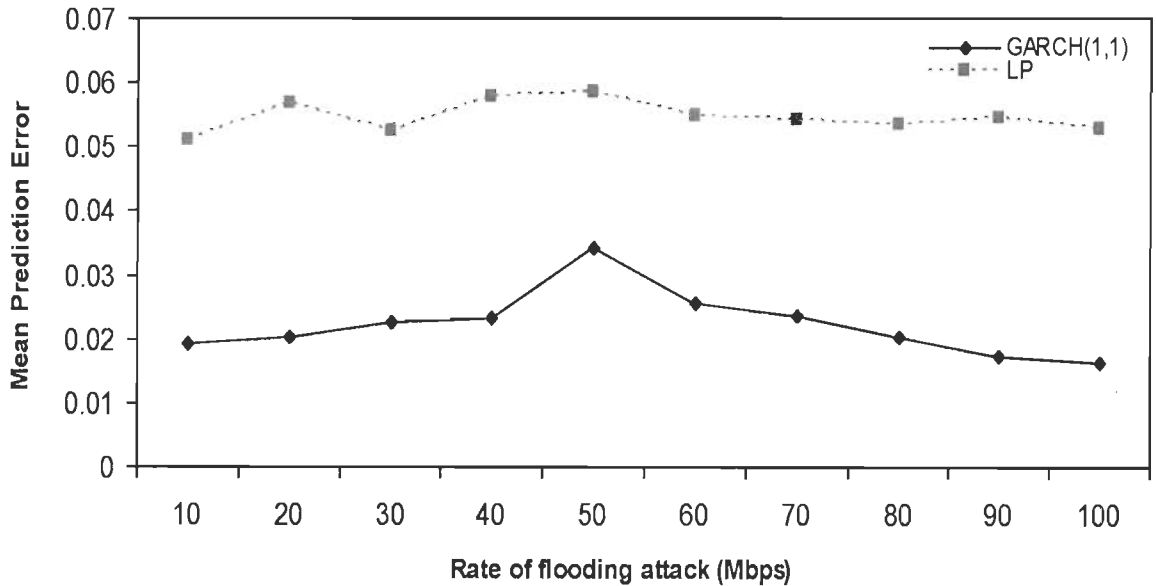


Figure 4.3 Comparison of prediction error values

4.7.4 Results and Discussion

Using normal frames, we set approximate threshold limit α on error values for GARCH(1,1). Using this threshold, we detected the attacks using algorithm discussed for GARCH(1,1) in section 4.6.2.

4.7.4.1. Detection Rate

We collected traces of normal traffic from the simulations in NS-2 network simulator to evaluate GARCH model based detection scheme. The abnormal traffic is generated with different attack strengths ranging from 10Mbps to 100Mbps. Detection rate for GARCH(1,1) and LP for various attack strengths are shown in table 4.5 and table 4.6, respectively. As we can see from the table 4.5 and table 4.6, detection rate for the GARCH(1,1) model is much better than LP model for all attack strength values considered.

Table 4.4. Mean prediction error for GARCH (1,1) and LP

| Attack Strength (Mbps) | GARCH(1,1) | LP |
|-----------------------------------|-------------------|-----------|
| 10 | 0.0194 | 0.0512 |
| 20 | 0.0202 | 0.0568 |
| 30 | 0.0224 | 0.0523 |
| 40 | 0.0232 | 0.0576 |
| 50 | 0.0343 | 0.0584 |
| 60 | 0.0256 | 0.0547 |
| 70 | 0.0234 | 0.0542 |
| 80 | 0.0201 | 0.0534 |
| 90 | 0.0174 | 0.0545 |
| 100 | 0.0164 | 0.0528 |

Table 4.5. Detection rate and false positive rate for GARCH (1,1)

| Attack Strength (Mbps) | Detection rate | False positive rate |
|-----------------------------------|---------------------------|--------------------------------|
| 10 | 92.4 | 6.0588 |
| 20 | 94.7 | 7.6471 |
| 30 | 95.3 | 8.2353 |
| 40 | 96 | 8.8235 |
| 50 | 98.2 | 8.2353 |
| 60 | 98.2 | 8.8235 |
| 70 | 98.8 | 9.4118 |
| 80 | 99.4 | 9.2532 |
| 90 | 99.6 | 11.641 |
| 100 | 99.6 | 11.695 |

Table 4.6. Detection rate and false positive rate for LP

| Attack Strength (Mbps) | Detection rate | False positive rate |
|-----------------------------------|---------------------------|--------------------------------|
| 10 | 87.6 | 4.9364 |
| 20 | 89.2 | 5.6426 |
| 30 | 91.7 | 5.2329 |
| 40 | 92.5 | 6.3215 |
| 50 | 92.9 | 6.8303 |
| 60 | 94.0 | 7.8235 |
| 70 | 94.3 | 8.710 |
| 80 | 94.8 | 9.132 |
| 90 | 94.8 | 9.132 |
| 100 | 95.4 | 9.896 |

4.7.4.2. False positive rate

The analysis for false positive rate is performed similar to the detection rate. We collected traces of normal traffic from the simulations in NS-2 network simulator to evaluate the detection mechanism. The abnormal traffic is generated with different attack strengths ranging from 10Mbps to 100Mbps. The false positive rate for GARCH(1,1) and LP models for various attack strengths are shown in table 4.5 and table 4.6, respectively. The results in table 4.5 and 4.6 show that the false positive rate of GARCH(1,1) is slightly greater than LP model. This can be mentioned as one of the drawback of GARCH model. However, false-positive rate shows the rate at which the model will give alarm while there is no attack. This is less serious drawback considering its superior performance in detecting real attack.

4.7.4.3. Detection delay

The detection delay for the GARCH(1,1) model based attack detection and LP are shown in table 4.7 and table 4.8, respectively. In the table, best, average and worst case detection delays are given. It can be vividly seen that GARCH(1,1) has lesser detection delay compared to the LP model for all attack strength values. This is a very important performance of the model. The same result is displayed in figure 4.4. The figure shows the average detection delay in case of GARCH(1,1) and LP models and the superior performance of former is clear.

Table 4.7. Detection Delay in seconds using GARCH (1,1) for flooding attacks

| Attack Strength (Mbps) | Best | Avg. | Worst |
|-----------------------------------|-------------|-------------|--------------|
| 10 | 0.0930 | 0.1264 | 0.2030 |
| 20 | 0.0930 | 0.1405 | 3.6560 |
| 30 | 0.0930 | 0.1263 | 0.2030 |
| 40 | 0.0930 | 0.1393 | 3.8130 |
| 50 | 0.0930 | 0.1257 | 0.2190 |
| 60 | 0.0930 | 0.1230 | 0.2180 |
| 70 | 0.0930 | 0.1260 | 0.2030 |
| 80 | 0.0930 | 0.1241 | 0.2030 |
| 90 | 0.1090 | 0.1250 | 0.2030 |
| 100 | 0.0940 | 0.1259 | 0.2030 |

Table 4.8. Detection Delay in seconds using LP for flooding attacks

| Attack Strength (Mbps) | Best | Avg. | Worst |
|------------------------|--------|--------|--------|
| 10 | 0.0930 | 0.1528 | 0.2820 |
| 20 | 0.0930 | 0.1725 | 3.2650 |
| 30 | 0.0930 | 0.1420 | 0.2480 |
| 40 | 0.0910 | 0.1420 | 3.8130 |
| 50 | 0.0910 | 0.1545 | 0.2190 |
| 60 | 0.1010 | 0.1445 | 0.2480 |
| 70 | 0.1010 | 0.1510 | 0.2820 |
| 80 | 0.0930 | 0.1345 | 0.2480 |
| 90 | 0.9850 | 0.1265 | 0.2820 |
| 100 | 0.1010 | 0.1297 | 0.2820 |

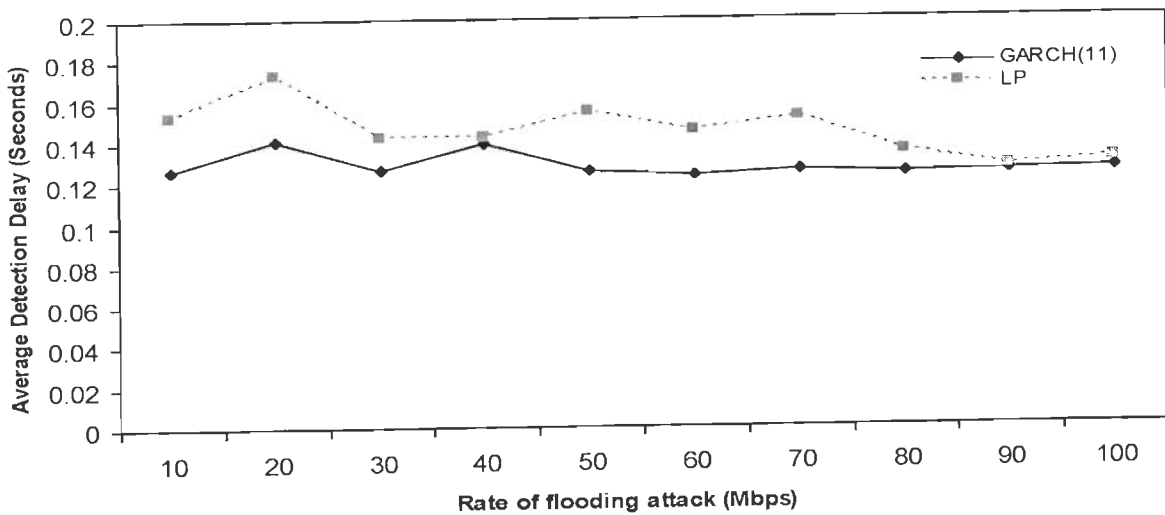


Figure 4.4 Detection delay in both GARCH(1,1) and LP model

4.8 Comparison between FVBA and GARCH Model based DDoS Attack Detection Schemes

In this section, detection performance of FVBA scheme proposed in previous chapter is compared with GARCH model based detection scheme proposed in this chapter.

Table 4.9. Comparison between FVBA and GARCH model based scheme for DDoS attack detection

| Scheme considered | Detection rate | False positive rate |
|--------------------------|----------------|---------------------|
| FVBA scheme | 98.8% | 3% |
| GARCH model based scheme | 99.6% | 11.69% |

Table 4.9 gives the detection rate and false positive rate for both schemes. It can be seen that detection performance of GARCH model based detection scheme is marginally better than FVBA but false positive rate is more in GARCH based scheme. One may consider the slight increase in false positive rate of GARCH as a drawback but the fact is that false positive simply means that the system gives an alert for attack while there is none. This is less serious than having an attack and not detecting it. GARCH model has a better detection performance and detection of DDoS attacks using non-linear time series may be preferable in critical applications.

4.9 Chapter Summery

In this chapter, we have shown how nonlinear time series model can be used to detect flooding DDoS attacks efficiently. Specifically, we used the GARCH model. The model is an autoregressive one and makes use of the hetroskedastic nature of network traffic data. A detailed discussion on the selection of order of model has been given based on the AIC and BIC criteria and it was found that GARCH(1,1) is able to detect flooding DDoS attack with acceptable error limit. The simulation results show that it is able to capture long range dependence and hence is more efficient in detecting flooding DDoS attack than other models like linear prediction. We have also compared performance of the GARCH model with flow-volume based approach (FVBA) and results show that it marginally outperforms FVBA.

CHAPTER 5

PREDICTING NUMBER OF ZOMBIES IN A DDoS ATTACK USING VARIOUS REGRESSION MODELS

Anomaly based DDoS detection systems construct profile of the traffic normally seen in the network, and identify anomalies whenever traffic deviate from normal profile beyond a threshold. This deviation in traffic beyond threshold is used in the past for DDoS detection but not for finding number of zombies. This chapter presents an approach that utilizes this deviation in traffic to predict number of zombies using various regression models i.e. linear, polynomial, exponential, power, logarithmic and multiple. A relationship is established between number of zombies and observed deviation in sample entropy and between number of zombies and observed deviation in volume and flow for simple and multiple regression, respectively. Various statistical performance measures, such as coefficient of determination (R^2), coefficient of correlation (CC), sum of square error (SSE), mean square error (MSE), normalized mean square error (NMSE) and nash–sutcliffe efficiency index (η) [22, 133] are used to study the strength of various regression models for predicting number of zombies. Network topologies similar to Internet are used for simulation and are generated using Transit-Stub model of GT-ITM topology generator. NS-2 network simulator on Linux platform is used for launching DDoS attacks with varied number of zombies. A comparative study of different regression models for predicting number of zombies is performed. The simulation results are promising as we are able to predict number of zombies efficiently using various regression models.

5.1 Introduction

In anomaly based DDoS detection mechanisms, the profile of the traffic normally seen in the network is constructed and anomalies are identified whenever traffic deviates from normal profile beyond a threshold. Proposed approach utilizes this deviation in traffic beyond threshold to predict number of zombies using various regression models. A real time estimation of the number of zombies in DDoS scenario is helpful to suppress the effect of

attack by choosing predicted number of most suspicious attack sources for either filtering or rate limiting. Moore et. al [65] have already made a similar kind of attempt, in which they have used backscatter analysis to estimate number of spoofed addresses involved in DDoS attack. This was an offline analysis based on unsolicited responses. In another approach [121], authors have used linear regression and correlation analysis to predict number of zombies. But due to the nonlinear nature of DDoS attack traffic, this method is unable to predict the number of zombies accurately.

Our objective is to find the relationship between number of zombies involved in a flooding DDoS attack and deviation in sample entropy and number of zombies and observed deviation in volume and flow for simple and multiple regression, respectively. In order to predict number of zombies, several models are developed using various regression techniques. A comparative study is performed between different regression models for predicting number of zombies.

5.2 Regression Models

Regression analysis [69, 86, 130] is a statistical tool used to investigate relationships between variables. Usually, the investigator seeks to find out the causal effect of one variable upon another. More specifically, regression analysis helps to understand how the typical value of the dependent variable changes when any one of the independent variables is varied, while the other independent variables are held constant. Variables which are used to explain other variables are called explanatory variables. Variables which are explained are called response variables. A response variable is also called a dependent variable and an explanatory variable is called an independent variable. When there is only one explanatory variable the regression model is called simple regression, whereas if there are more than one explanatory variables, the regression model is called multiple regression.

5.2.1 Types of Regression Model Used

A. Simple Regression Models

1. **Linear regression:** Linear regression [71, 184] includes any approach to model the relationship between a dependent variable Y and an independent variable X , such that the model depends linearly on the unknown parameter to be estimated from the data. Such a

model is called a linear model. On the other hand, multiple regression uses two or more independent variables to predict the outcome. The general form of linear regression is:

$$M1: Y_i = \hat{Y}_i + \varepsilon_i \tag{5.1}$$

$$\hat{Y}_i = \beta_0 + \beta_1 X_i,$$

where

- Y_i is dependent variable
- X_i is independent variable
- β_0 is intercept
- β_1 is slope
- ε is regression residual

2. Polynomial regression: Polynomial regression [179, 194] is a form of regression in which the relationship between the independent variable X and the dependent variable Y is modeled as an i^{th} order polynomial. The general form of this regression model is as follows:

$$M2: Y_i = \hat{Y}_i + \varepsilon_i \tag{5.2}$$

$$\hat{Y}_i = \beta_0 + \beta_1 X + \beta_2 X^2 + \dots + \beta_n X^n$$

where, β_i is i^{th} regression coefficient and X and Y_i are given above.

3. Logarithmic regression: A logarithmic regression [18, 81, 83] is also known as logarithmic least squares fittings. For the relation between dependent and independent variables, it finds the logarithmic function that best fits a given set of data points. Logarithmic data will exhibit a straight-line relationship when graphed with the X values on a log scale and the Y values on a linear scale. A logarithmic regression has the following general form:

$$M3: Y_i = \hat{Y}_i + \varepsilon_i \tag{5.3}$$

$$\hat{Y}_i = \beta_0 \ln(X_i) + \beta_1$$

where, β_0 and β_1 are regression coefficients and X_i and Y_i are given above.

4. Power regression: Power regression [152, 203], also known as log-log regression, takes the input signal and fits the function to it where X is the variable along the x-axis. The function is based on the linear regression, with both axes scaled logarithmically. Power regressions will not allow an independent variable value of zero. A power regression has the following general form:

$$M4: Y_i = \hat{Y}_i + \varepsilon_i \quad (5.4)$$

$$\hat{Y}_i = \beta_0 \cdot X_i^{\beta_1}$$

Where, β_0, β_1, X_i and Y_i are given above.

5. Exponential regression: An exponential regression [84, 206] is also known as exponential least square fitting. For the relation between two variables, it finds the exponential function that best fits a given set of data points. Exponential regression takes the input signal and fits an exponential function to it where X is the variable along the x-axis. An exponential regression has the following general form:

$$M5: Y_i = \hat{Y}_i + \varepsilon_i \quad (5.5)$$

$$\hat{Y}_i = \beta_0 \cdot e^{\beta_1 X_i}$$

Where, β_0, β_1, X_i and Y_i are given above.

B. Multiple Regression Model

The general purpose of multiple regression [61, 86] is to learn more about the relationship between several independent variables and a dependent variable. In the multivariate case, when there is more than one independent variable, the regression line can not be visualized in the two dimensional space, but can be computed just as easily. In general form of multiple regression given in equation 5.6, there are p independent variables:

$$Y_i = \hat{Y}_i + \varepsilon_i \quad (5.6)$$

$$\hat{Y}_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_p X_{pi},$$

where

- Y is dependent variable

- X_1, X_2, \dots, X_p are p independent variables

- β_0 is intercept
- $\beta_1, \beta_2, \dots, \beta_p$ are the coefficients of p independent variables
- ε is regression residual

5.2.2 Estimating Number of Zombies

A Using Simple Regression

To predict number of zombies, we established relationship between number of zombies Y (output) and observed deviation in entropy X (input). For different given (known) zombies, deviation in sample entropy X is calculated as $(H_c - H_n)$, where H_c and H_n are entropy values at the time of attack detection and for normal profile, respectively. Regression equations are then determined by the process of curve fitting. These equations are used for predicting number of zombies.

B Using Multiple Regression

For the case of multiple regression, similar procedure as described above is used. To predict number of zombies, we established relationship between number of zombies Y (output) and observed deviation in volume X_1 (input) and flow X_2 (input). Regression equation is then determined by the process of curve fitting. This equation is used for predicting number of zombies.

5.3 Statistical Performance Measures

Various statistical performance measures, such as coefficient of determination (R^2), coefficient of correlation (CC), sum of square error (SSE), mean square error (MSE), normalized mean square error (NMSE) and nash–sutcliffe efficiency index (η) [22, 133] are used to evaluate the performance of various regression models. These measures are defined below. In the definitions, N represents the number of feature vectors prepared, Y_o and Y_c denote the actual and the predicted values of dependent variable, respectively, \bar{Y}_o and σ_{obs}^2 are the mean and the standard deviation of the actual dependent variable, respectively.

i). Coefficient of Determination (R^2): Coefficient of determination (R^2) is a descriptive measure of the strength of the regression relationship, a measure how well the regression line fit to the data [22]. R^2 is the proportion of variance in dependent variable which can be predicted from independent variable. The coefficient of determination (R^2) can be defined as:

$$R^2 = \frac{\left(\sum_{i=1}^N (Y_o - \bar{Y}_o)(Y_c - \bar{Y}_c) \right)^2}{\left[\sum_{i=1}^N (Y_o - \bar{Y}_o)^2 \cdot \sum_{i=1}^N (Y_c - \bar{Y}_c)^2 \right]} \quad (5.7)$$

ii). Coefficient of Correlation (CC): The Coefficient of Correlation (CC) [22] can be defined as:

$$CC = \frac{\sum_{i=1}^N (Y_o - \bar{Y}_o)(Y_c - \bar{Y}_c)}{\left[\sum_{i=1}^N (Y_o - \bar{Y}_o)^2 \cdot \sum_{i=1}^N (Y_c - \bar{Y}_c)^2 \right]^{1/2}} \quad (5.8)$$

iii). Sum of Squared Errors (SSE): The Sum of Squared Errors (SSE) [22] can be defined as:

$$SSE = \sum_{i=1}^N (Y_o - Y_c)^2 \quad (5.9)$$

iv). Mean Square Error (MSE): The Mean Square Error (MSE) [22] can be defined as:

$$MSE = \frac{\sum_{i=1}^N (Y_c - Y_o)^2}{N} \quad (5.10)$$

v). Normalized Mean Square Error (NMSE): The Normalized Mean Square Error (NMSE) [219] can be defined as:

$$NMSE = \frac{\frac{1}{N} \sum_{i=1}^N (Y_c - Y_o)^2}{\sigma_{obs}^2} \quad (5.11)$$

vi). Nash–Sutcliffe efficiency index (η): The Nash–Sutcliffe efficiency index (η) [145] can be defined as:

$$\eta = 1 - \frac{\sum_{i=1}^N (Y_c - Y_o)^2}{\sum_{i=1}^N (Y_o - \bar{Y}_o)^2} \quad (5.12)$$

5.4 Simulation Setup

The topology and simulation parameters as discussed in chapter 3 are used in this work also. However, the simulation experiments are done in different manner i.e. earlier the number of zombies were kept constant, but in this case attack strength is kept constant and number of zombies are varied. In our simulation experiments, attack traffic rate is fixed to 25Mbps in total; and, mean attack rate per zombie varies from 0.25Mbps to 2.5Mbps as total number of zombie machines range between 10 and 100 to generate attack traffic. The simulations are repeated and different attack scenarios are generated by varying total number of zombie machines and at fixed attack strength. Figure 5.1 shows entropy variation with time for 10-100 numbers of zombies, where $H(n)$ is the entropy value for n zombies. Figure 5.2 and figure 5.3 show flow and volume variation with time for 10 to 100 numbers of zombies, where $F(n)$ and $X(n)$ are the flow and volume values for n zombies, respectively.

5.5 Model Development and Experimental Analysis

In this section, we describe our experiments to study the strength of various regression models for predicting number of zombies involved in a DDoS attack. For simple regression models, we collected deviation in entropy by varying total number of zombies from 10 to 100 and the data is shown in table 5.1. Similarly, for multiple regression model, volume and flow data is collected by varying number of zombies as shown in table 5.2. The inputs to the multiple regression model are number of zombies Y and observed deviation in sample volume X_1 and flow X_2 . Coefficients of regression equations are determined through a process of curve fitting. The main objective in the process of the curve fitting is to minimize the error between the actual number of zombies and the predicted number of zombies. Figure 5.4 to 5.8 show the regression equation and coefficient of determination for simple regression models

(M1 to M5) as discussed in section 5.2. Using these equations and deviation in entropy values, predicted number of zombies are calculated.

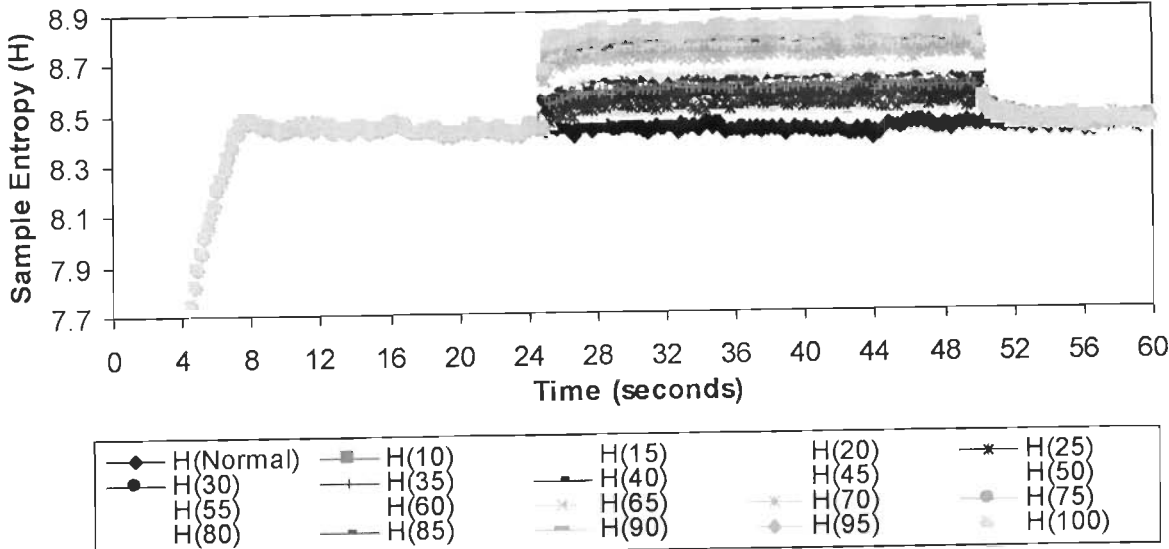


Figure 5.1. Entropy variation with varied number of zombies

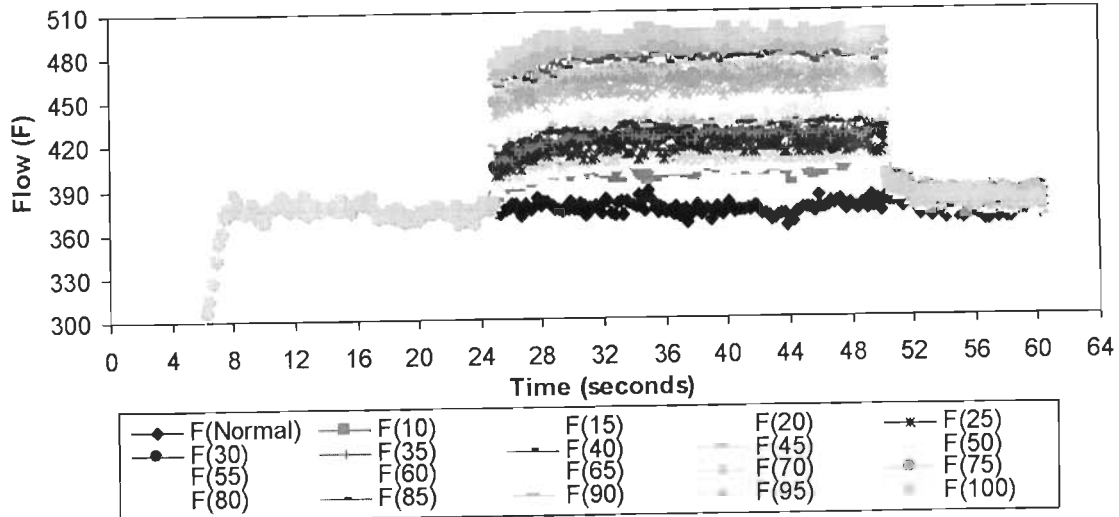


Figure 5.2. Flow variation with varied number of zombies

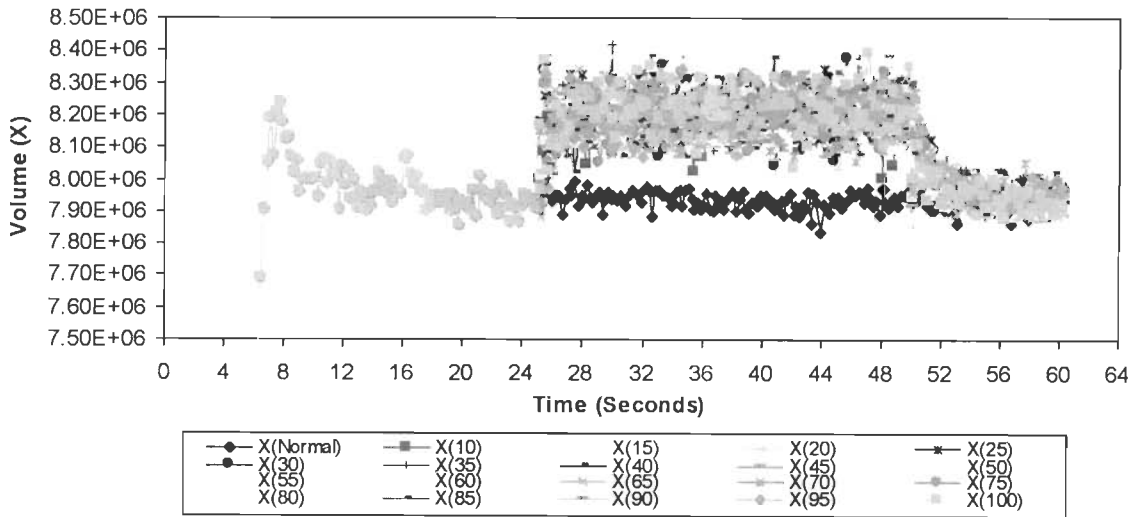


Figure 5.3. Volume variation with varied number of zombies

Table 5.1. Deviation in entropy with actual number of zombies

| Actual Number of Zombies (Y) | Deviation in Entropy (X) ($H_c - H_n$) |
|------------------------------|--|
| 10 | 0.045 |
| 15 | 0.046 |
| 20 | 0.048 |
| 25 | 0.050 |
| 30 | 0.068 |
| 35 | 0.087 |
| 40 | 0.099 |
| 45 | 0.111 |
| 50 | 0.121 |
| 55 | 0.130 |
| 60 | 0.139 |
| 65 | 0.148 |
| 70 | 0.157 |
| 75 | 0.163 |
| 80 | 0.170 |
| 85 | 0.176 |
| 90 | 0.182 |
| 95 | 0.189 |
| 100 | 0.192 |

In a similar fashion, using flow and volume as inputs to the multiple regression equation, predicted number of zombies are obtained. For the multiple regression, the regression equation is given in equation 5.13. The coefficient of determination for the multiple regression model is 0.99.

$$Y = X_1 * (1.389E - 05) + X_2 * 1.984 - 19.25 \quad (5.13)$$

where X_1 and X_2 represent deviation in sample volume and flow, respectively.

Table 5.2. Deviation in volume and flow with actual number of zombies

| Actual Number of Zombies (Y) | Deviation in volume (X_1) | Deviation in Flow (X_2) |
|------------------------------|-------------------------------|-----------------------------|
| 10 | 126288.57 | 13.69 |
| 15 | 134290.16 | 16.15 |
| 20 | 140013.65 | 18.77 |
| 25 | 150433.33 | 21.14 |
| 30 | 141798.73 | 24.64 |
| 35 | 144329.52 | 26.91 |
| 40 | 139543.17 | 28.90 |
| 45 | 139947.94 | 31.60 |
| 50 | 144346.03 | 33.69 |
| 55 | 144883.17 | 35.86 |
| 60 | 141096.51 | 38.23 |
| 65 | 142149.84 | 41.25 |
| 70 | 133992.06 | 44.18 |
| 75 | 142261.27 | 46.38 |
| 80 | 132418.73 | 48.93 |
| 85 | 138190.159 | 51.85 |
| 90 | 133394.286 | 54.17 |
| 95 | 140716.825 | 57.10 |
| 100 | 143495.873 | 58.81 |

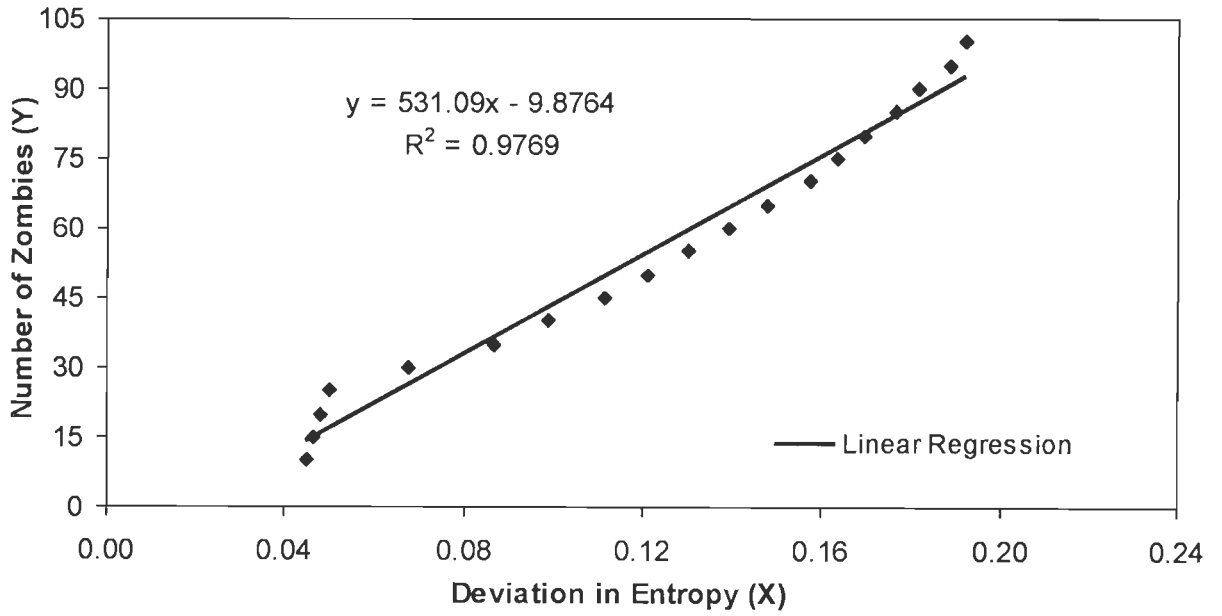


Figure 5.4. Regression equation and coefficient of determination for linear regression based model M1

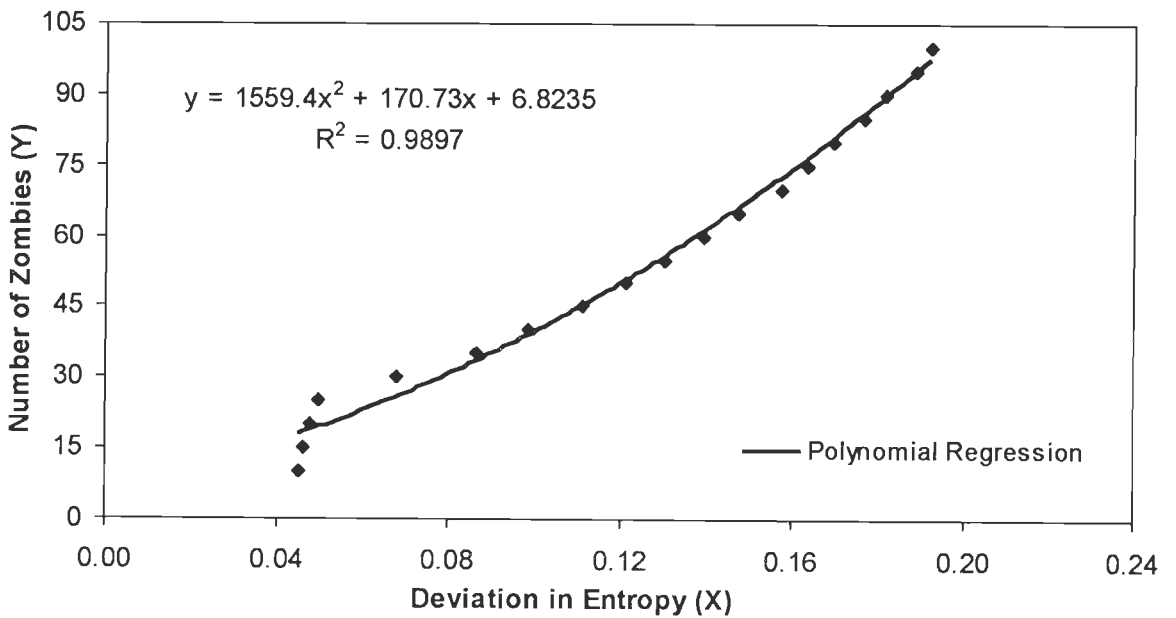


Figure 5.5. Regression equation and coefficient of determination for polynomial regression based model M2

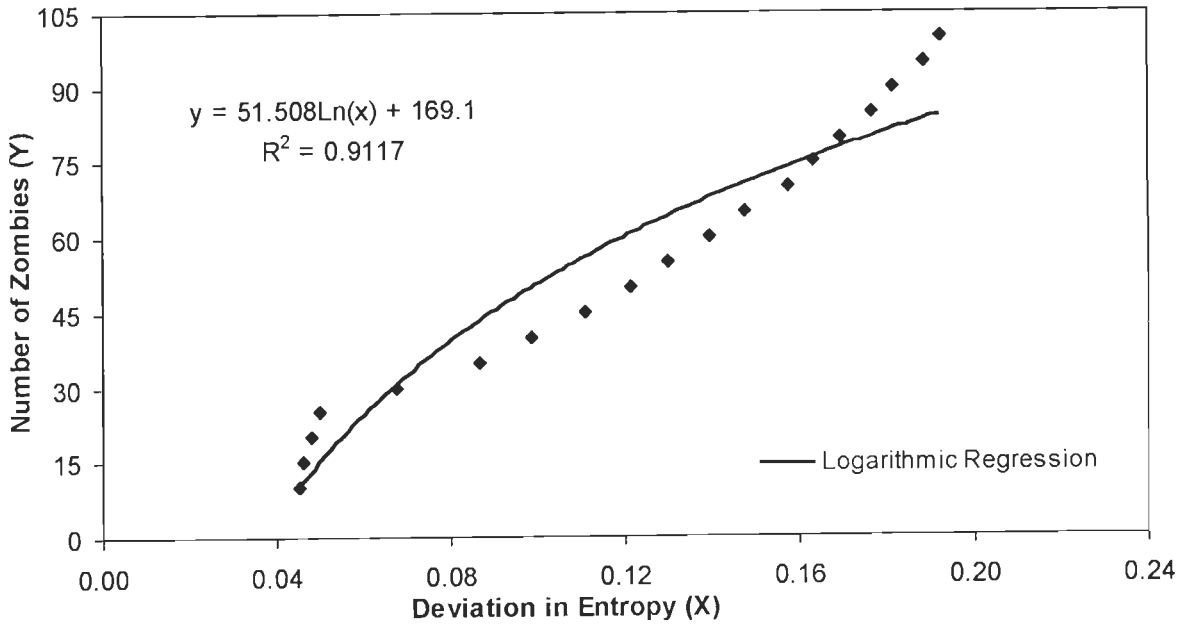


Figure 5.6. Regression equation and coefficient of determination for logarithmic regression based model M3

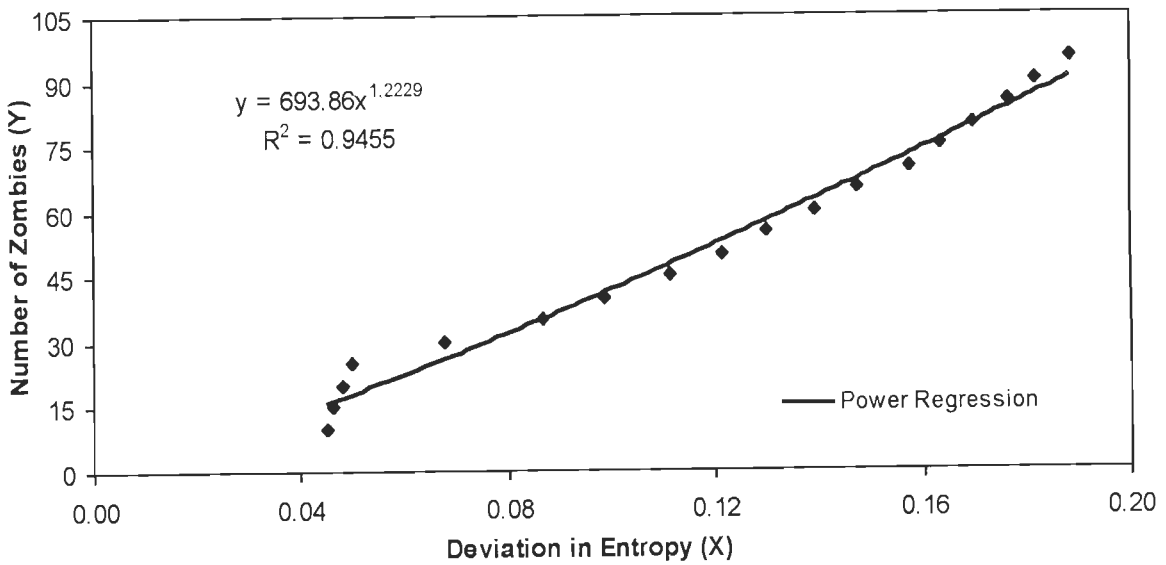


Figure 5.7. Regression equation and coefficient of determination for power regression based model M4

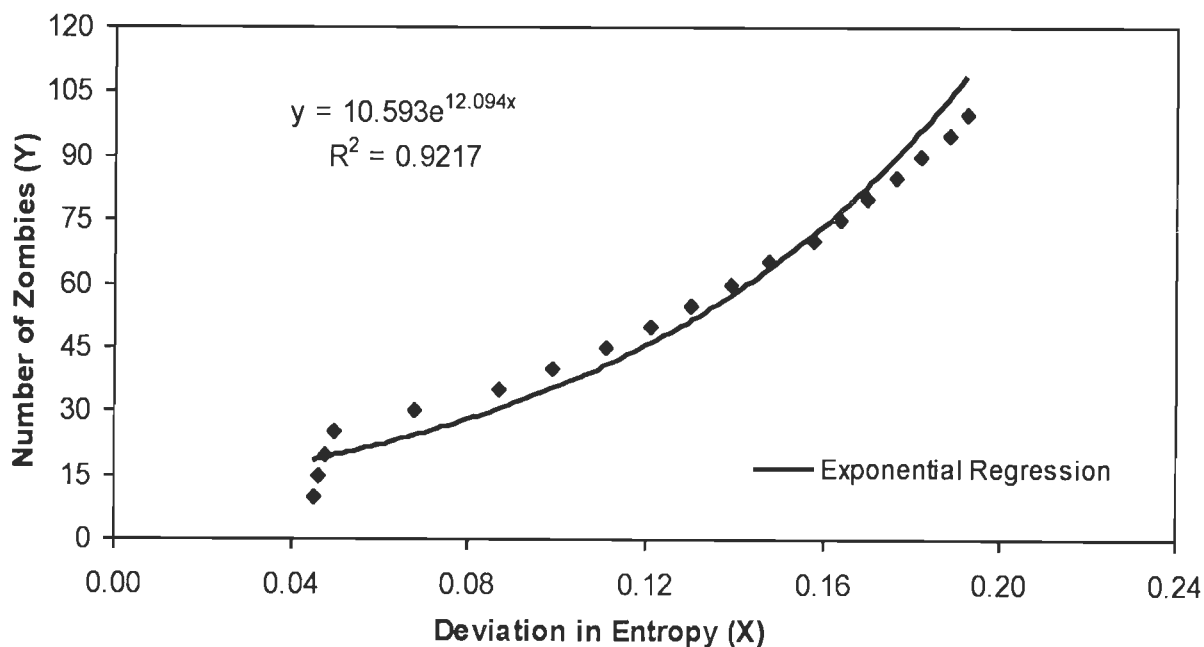


Figure 5.8. Regression equation and coefficient of determination for exponential regression based model M5

From the above figures, it can be inferred that from all basic regression models, the polynomial regression model M2 has the best curve fitting. However, an exhaustive comparison of the suitability of both the basic and multiple regression model is discussed in the following section.

5.6 Results and Discussion

Below we give the results of the comparison of the simple and multiple regression models. For clarity of the presentation, first simple regression models are separately compared and then the comparison of best found polynomial regression model with multiple regression is given.

A. Simple regression models

In this section, simulation results of models M1 to M5 given in section 5.5 are presented. The comparison between actual number of zombies and predicted number of zombies using various regression models (M1 to M5) is depicted in figures 5.9 to 5.13.

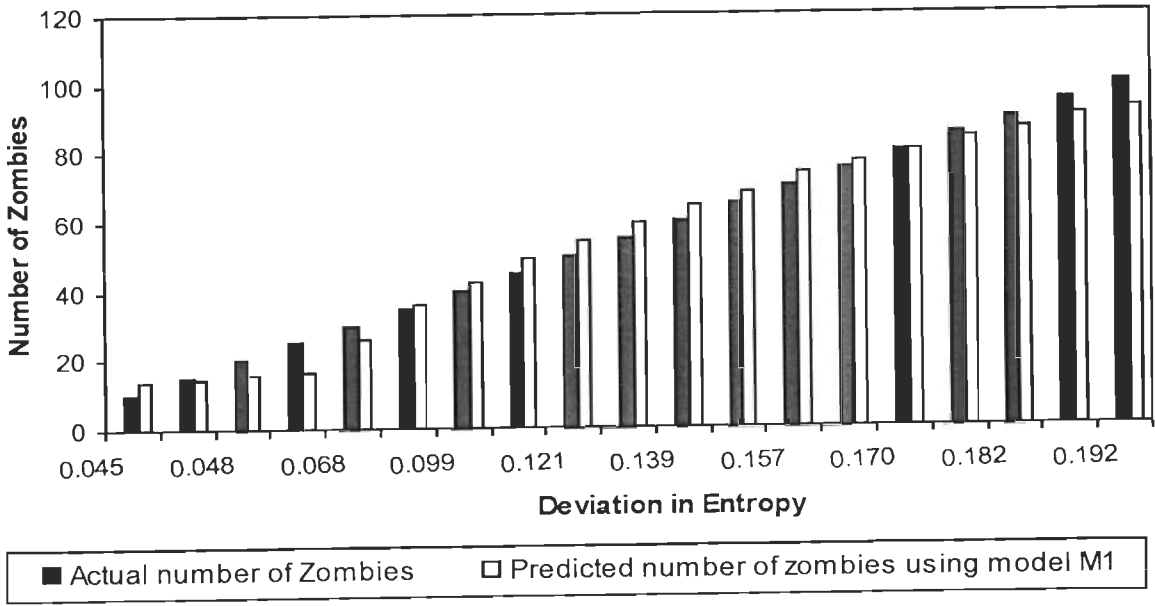


Figure 5.9. Comparison between actual number of zombies and predicted number of zombies using linear regression based model M1

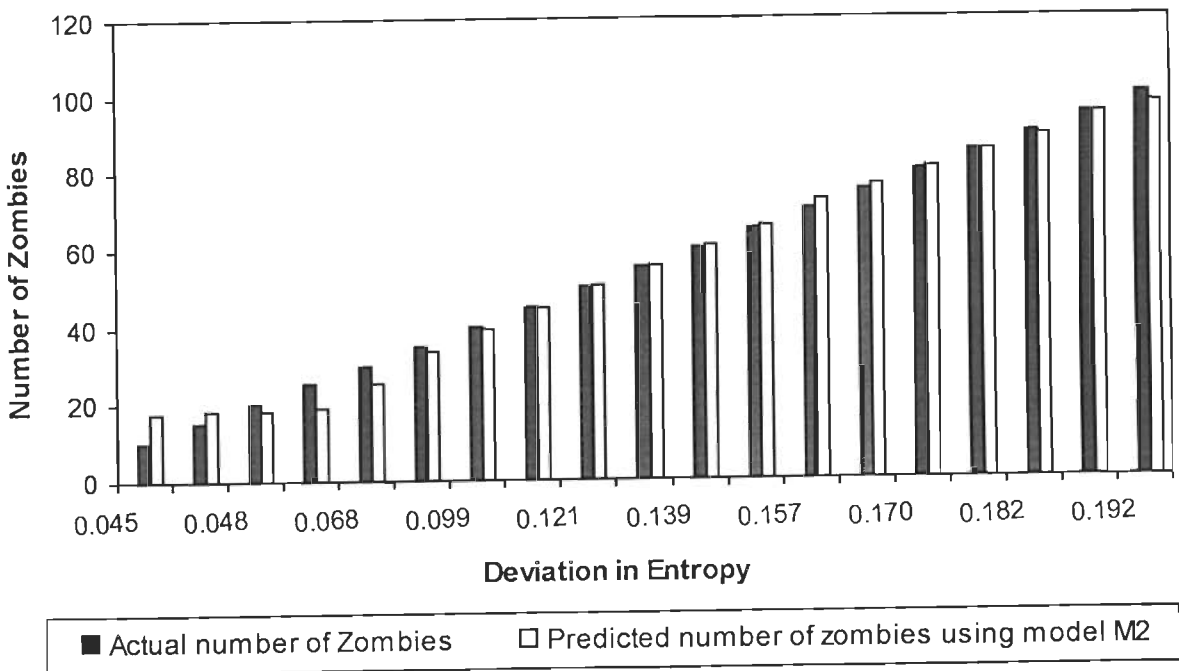


Figure 5.10. Comparison between actual number of zombies and predicted number of zombies using polynomial regression based model M2

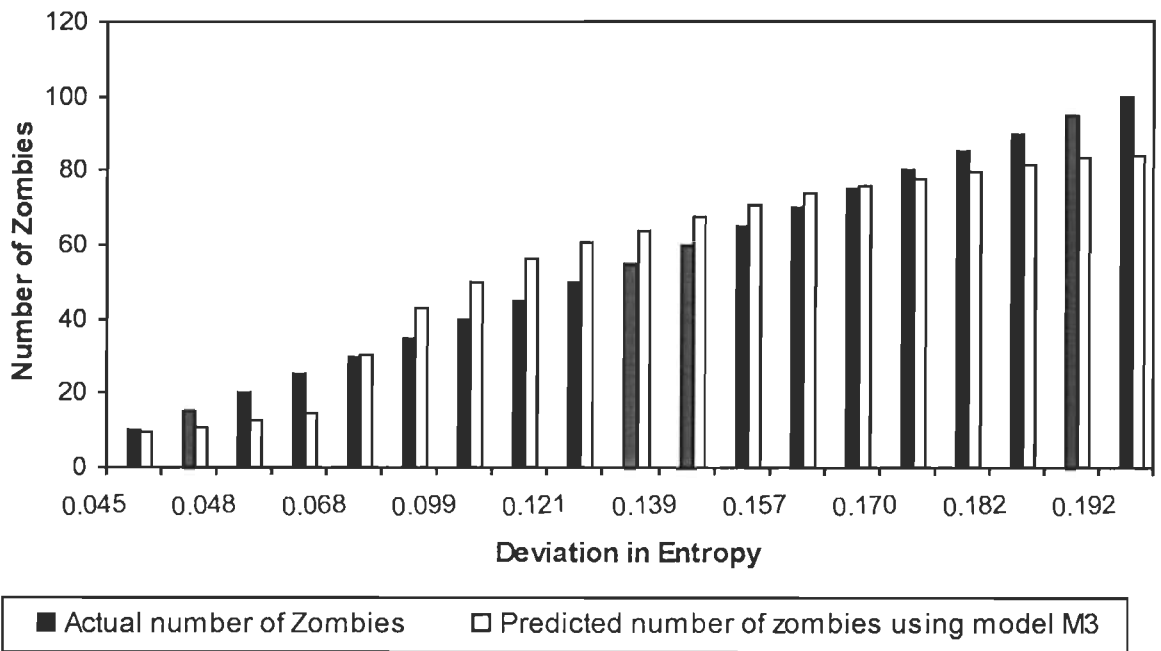


Figure 5.11. Comparison between actual number of zombies and predicted number of zombies using logarithmic regression based model M3

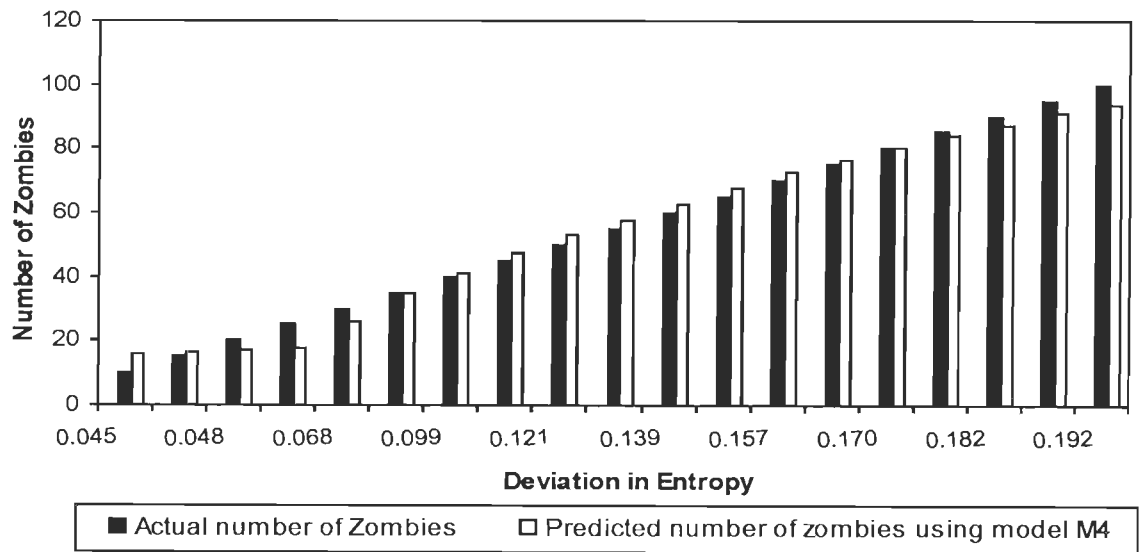


Figure 5.12. Comparison between actual number of zombies and predicted number of zombies using power regression based model M4

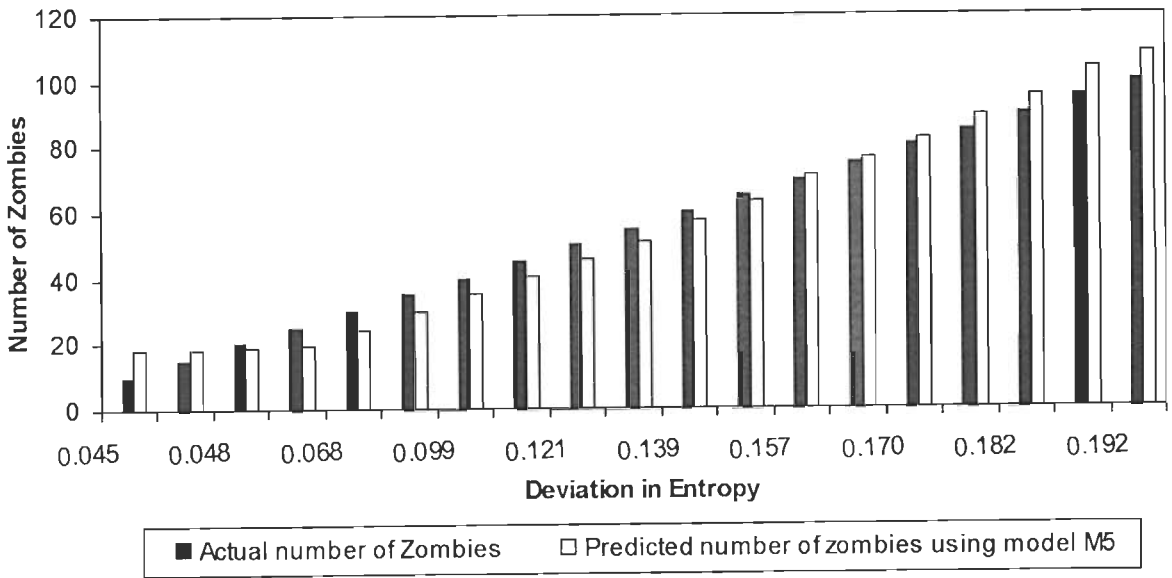


Figure 5.13. Comparison between actual number of zombies and predicted number of zombies using exponential regression based model M5

Figure 5.14 shows comparison between actual number of zombies and predicted number of zombies using various regression models M1 to M5.

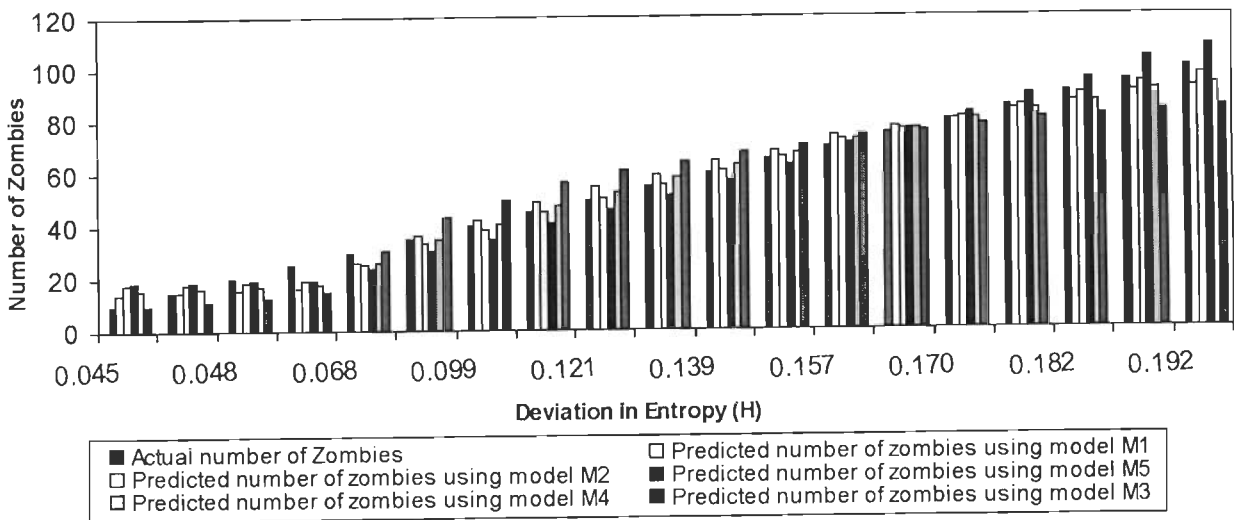


Figure 5.14. Comparison between actual number of zombies and predicted number of zombies using various regression models M1-M5

To represent false positive (falsely predicted normal clients as zombies) and false negative (zombies are identified as normal client), we plot residual error [121] for various regression models. Table 5.3 shows residual error for various regression models (M1 to M5). Figure 5.15 depicts summary of residual error in various regression models. Figures 5.16 to 5.21 show values of R^2 , CC, SSE, MSE, NMSE and η for various regression models (M1 to M5).

Table 5.3. Summary of residual error for various regression models

| (X) Entropy Variation | (Y) Number of Zombies | Residual error | | | | |
|-----------------------------|--------------------------------|----------------|-------------|-------------|-------------|-------------|
| | | Model M1 | Model M2 | Model M3 | Model M4 | Model M5 |
| 0.045 | 10 | 4.07 | 7.69 | -0.53 | 5.62 | 8.27 |
| 0.046 | 15 | -0.31 | 3.06 | -4.22 | 1.12 | 3.53 |
| 0.048 | 20 | -4.46 | -1.43 | -7.45 | -3.19 | -1.10 |
| 0.050 | 25 | -8.39 | -5.78 | -10.33 | -7.31 | -5.64 |
| 0.068 | 30 | -3.91 | -4.46 | 0.43 | -4.22 | -5.97 |
| 0.087 | 35 | 1.14 | -1.68 | 8.11 | -0.08 | -4.80 |
| 0.099 | 40 | 2.53 | -1.15 | 9.81 | 0.99 | -5.06 |
| 0.111 | 45 | 4.17 | 0.08 | 10.96 | 2.49 | -4.36 |
| 0.121 | 50 | 4.51 | 0.45 | 10.42 | 2.84 | -4.10 |
| 0.130 | 55 | 4.22 | 0.43 | 9.05 | 2.63 | -3.91 |
| 0.139 | 60 | 4.12 | 0.88 | 7.58 | 2.71 | -2.88 |
| 0.148 | 65 | 3.51 | 1.00 | 5.55 | 2.33 | -1.87 |
| 0.157 | 70 | 3.75 | 2.37 | 3.88 | 2.91 | 1.13 |
| 0.163 | 75 | 1.94 | 1.40 | 0.81 | 1.35 | 1.49 |
| 0.170 | 80 | 0.23 | 0.67 | -2.27 | -0.07 | 2.43 |
| 0.176 | 85 | -1.17 | 0.50 | -5.25 | -1.11 | 4.50 |
| 0.182 | 90 | -3.40 | -0.70 | -8.75 | -3.05 | 5.31 |
| 0.189 | 95 | -4.75 | -0.56 | -11.84 | -3.98 | 8.57 |
| 0.192 | 100 | -7.80 | -2.76 | -15.85 | -6.79 | 8.27 |

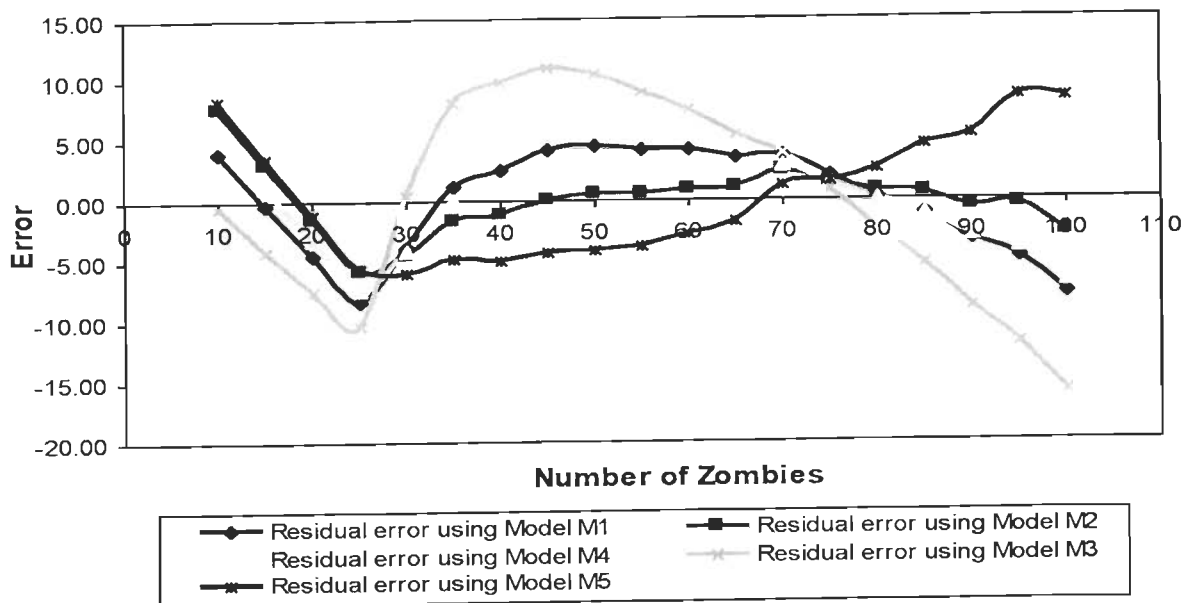


Figure 5.15. Summary of Residual error in various regression models

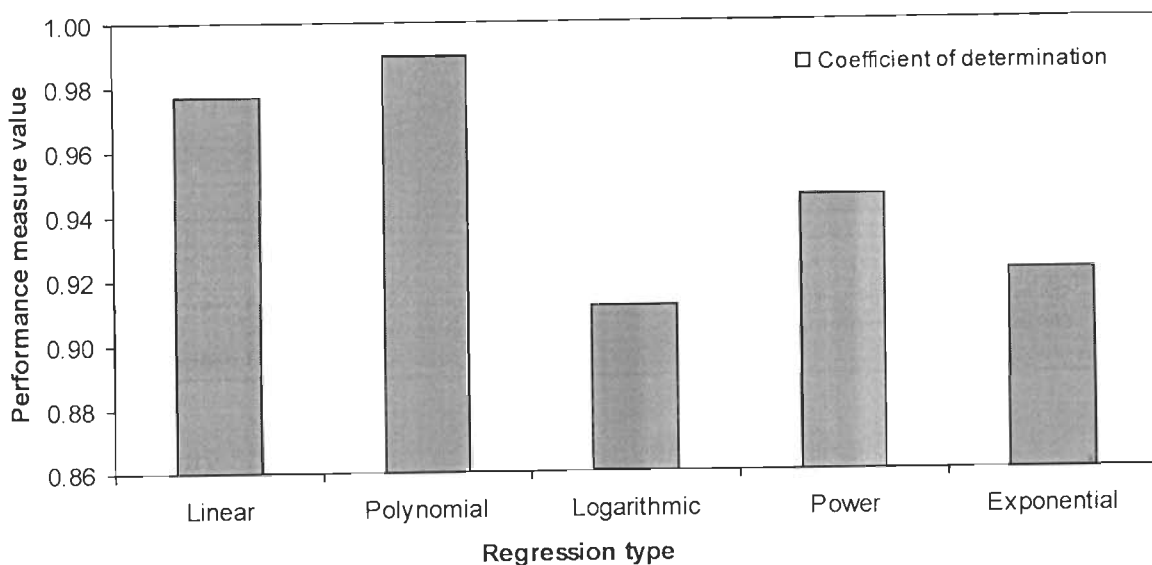


Figure 5.16. Value of coefficient of determination (R^2) for various regression models

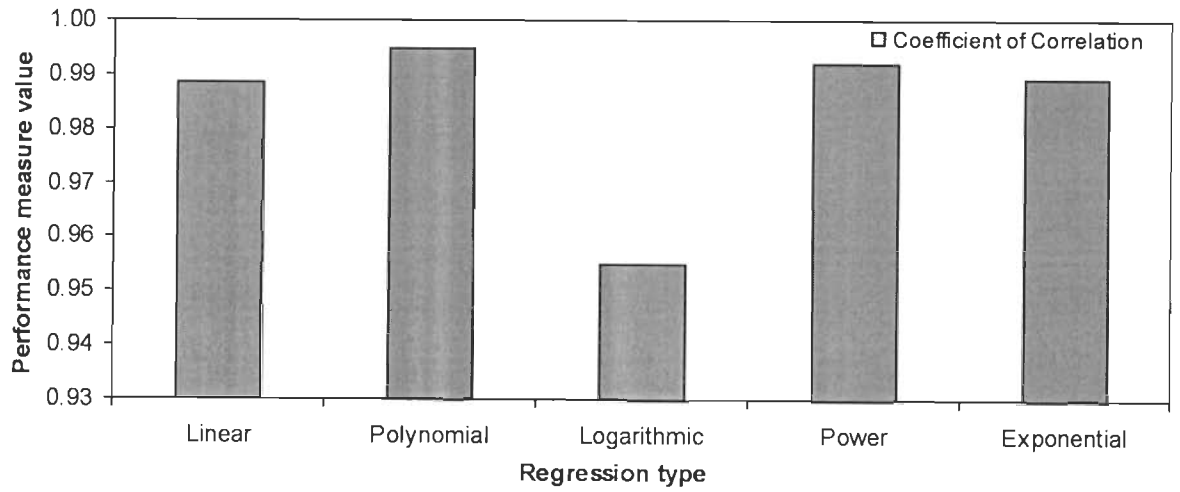


Figure 5.17. Value of coefficient of correlation (CC) for various regression models

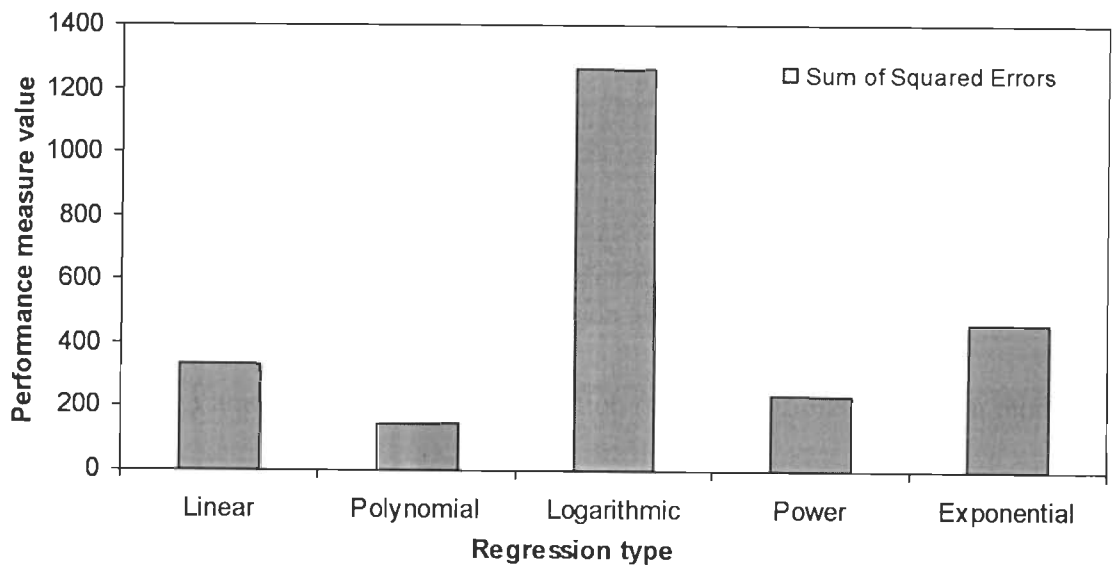


Figure 5.18. Value of sum of squared errors (SSE) for various regression models

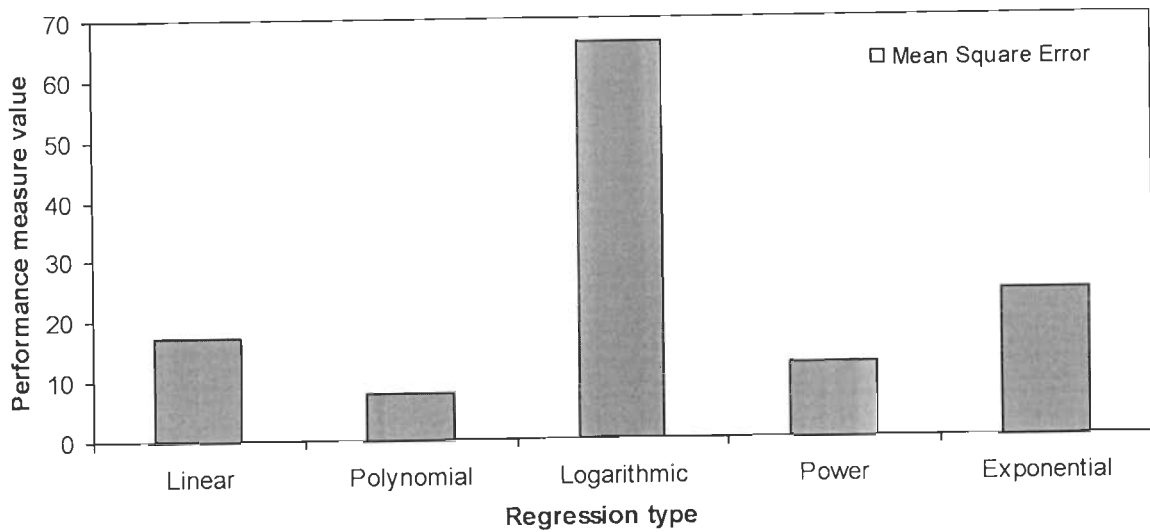


Figure 5.19. Value of mean square error (MSE) for various regression models

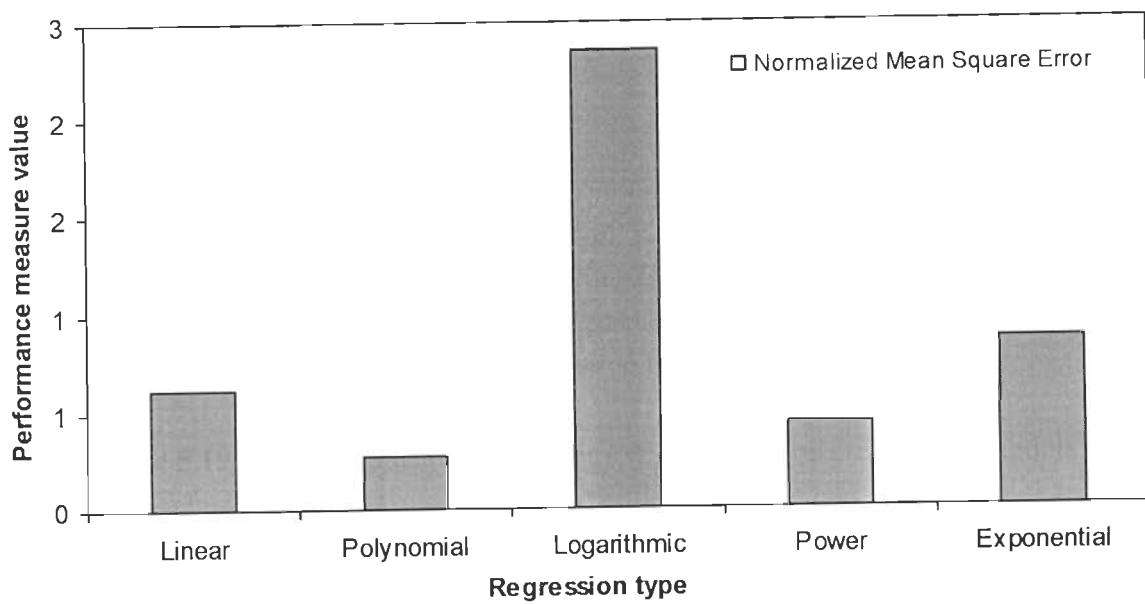


Figure 5.20. Value of normalized mean square error (NMSE) for various regression models

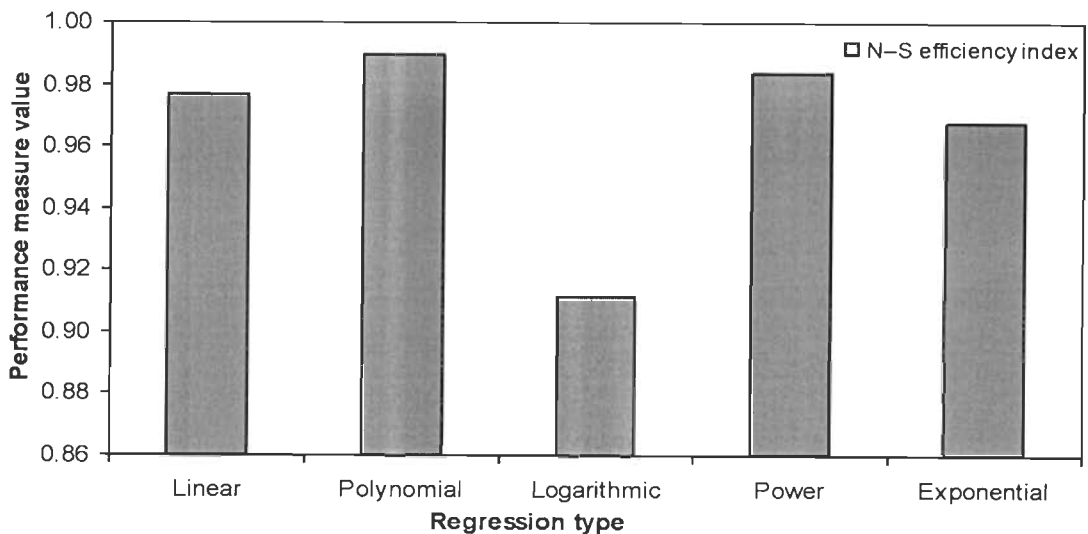


Figure 5.21. Value of Nash–Sutcliffe efficiency index (η) for various regression models

Table 5.4. Summary of various performance measures for simple regression models

| | Linear | Polynomial | Logarithmic | Power | Exponential |
|--------|--------|------------|-------------|--------|-------------|
| R^2 | 0.98 | 0.99 | 0.91 | 0.95 | 0.92 |
| CC | 0.99 | 0.99 | 0.95 | 0.99 | 0.99 |
| SSE | 328.88 | 146.88 | 1257.90 | 231.85 | 460.81 |
| MSE | 17.31 | 7.73 | 66.21 | 12.20 | 24.25 |
| NMSE | 0.62 | 0.27 | 2.35 | 0.43 | 0.86 |
| η | 0.98 | 0.99 | 0.91 | 0.98 | 0.97 |

Table 5.4 shows summary of various performance measures for simple regression models. As described in section 5.3, coefficient of determination (R^2) is a descriptive measure of the strength of the regression relationship, a measure how well the regression line fit to the data. R^2 is the proportion of variance in dependent variable which can be predicted from independent variable and CC is its square root. The Nash–Sutcliffe efficiency index is a widely used and potentially reliable statistic for assessing the goodness of fit of models. Essentially, the closer the model efficiency is to 1, the more accurate the model is. On the other hand, values of SSE, MSE and NMSE quantify the error in the prediction using various

regression models. Therefore, when comparing various regression models, model is selected with highest value of coefficient of determination, coefficient of correlation and Nash–Sutcliffe efficiency index and lowest values of SSE, MSE and NMSE. Accordingly, it can be seen from table 5.4 and figures 5.16 to 5.21 that polynomial regression based model M2 has highest value of coefficient of determination, coefficient of correlation and Nash–Sutcliffe efficiency index and lowest values of SSE, MSE and NMSE. Thus, it can be concluded that it performs better than other models.

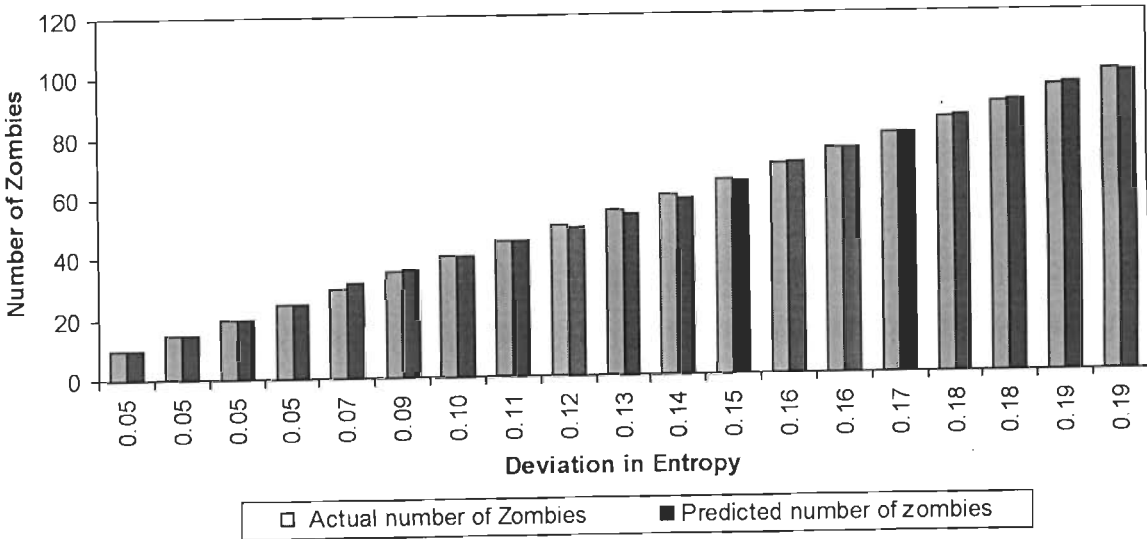


Figure 5.22. Comparison between actual number of zombies and predicted number of zombies using multiple regression model

To compare the performance of the various regression models (M1 to M5), the actual and predicted number of zombies are shown in figures 5.9 to 5.13. Observing the difference between the actual and predicted number of zombies in each figure, it can be verified that the polynomial regression based model M2 shown in figure 5.10 has the least difference between the actual and predicted number of zombies compared to other regression based models. The residual error for various regression model given in table 5.3 also verify the same fact. Though for some entropy values, residual errors in polynomial regression model are high but for most entropy values, it is having least residual errors. Hence, we can conclude that, number of zombies predicted by polynomial regression model is closest to the actual number of the zombies.

B. Multiple regression model

In this section, simulation results of the multiple regression model developed in section 5.5 are presented. The comparison between actual number of zombies and predicted number of zombies using multiple regression model is depicted in figure 5.22. To represent false positive and false negative, we plot residual error in figure 5.23 for multiple regression model. Table 5.5 shows values of various performance measures for multiple regression model.

Table 5.5. Values of various performance measures for multiple regression model

| | |
|--------|-------|
| R^2 | 0.99 |
| CC | 0.99 |
| SSE | 9.74 |
| MSE | 0.51 |
| NMSE | 0.018 |
| η | 0.99 |

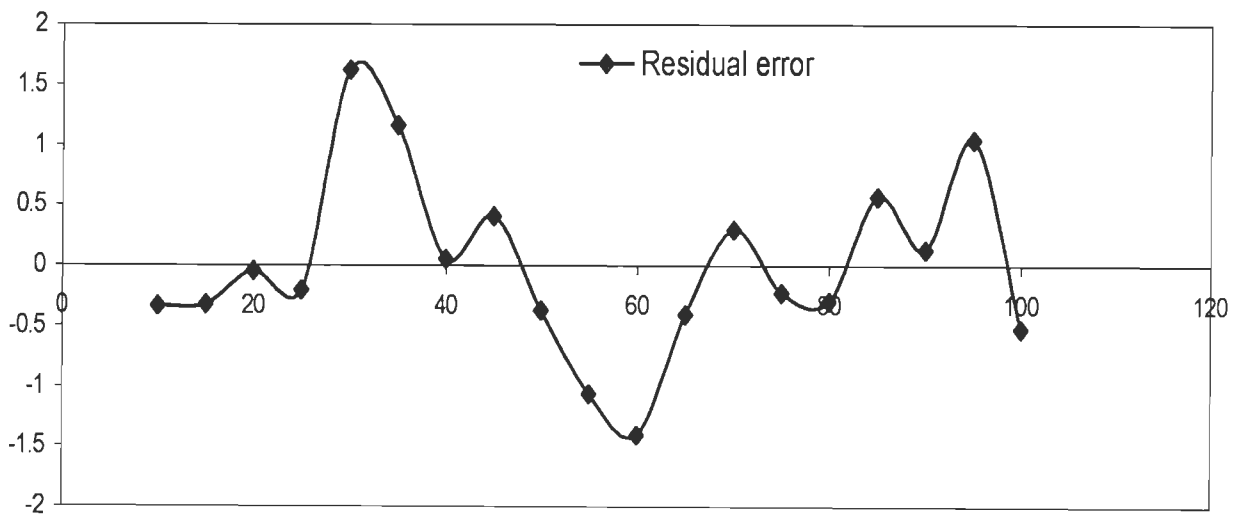


Figure 5.23. Residual error in multiple regression model

C. Comparison between Polynomial and multiple regression

Here performance between polynomial and multiple regression model is compared to predict number of zombies involve in a DDoS attack. Polynomial regression is compared with

multiple regression as it gives best performance among the simple regression models discussed in section 5.2.

It can be seen from table 5.6 that multiple regression model has higher value of coefficient of determination, coefficient of correlation and N-S efficiency index and lower values of SSE, MSE and NMSE. Thus, it can be concluded that it performs better than other models. It can also be verified from figure 5.24 that the difference between the actual and predicted number of zombies in multiple regression model is lower compared to that of the polynomial regression based model. Hence, we can conclude that, number of zombies predicted by multiple regression model is closest to the actual number of the zombies.

Table 5.6. Summary of various performance measures for polynomial and multiple regression

| Model | | |
|----------------|------------|----------|
| | Polynomial | Multiple |
| R ² | 0.99 | 0.99 |
| CC | 0.99 | 0.99 |
| SSE | 146.88 | 9.74 |
| MSE | 7.73 | 0.51 |
| NMSE | 0.27 | 0.018 |
| H | 0.99 | 0.99 |

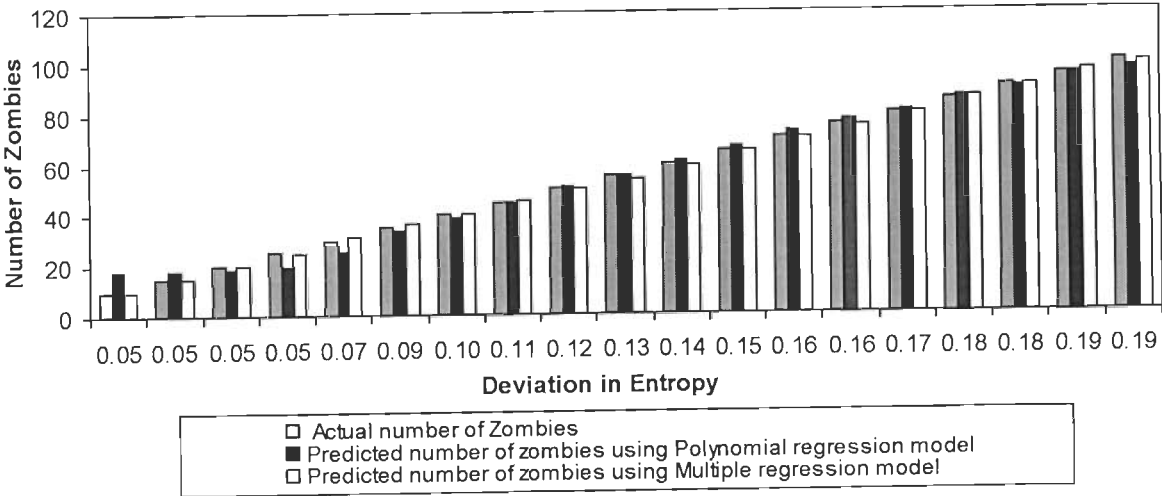


Figure 5.24. Comparison between actual number of zombies and predicted number of zombies using polynomial and multiple regression model

5.7 Chapter Summary

This chapter investigated suitability of various regression models to predict number of zombies involved in a flooding DDoS attack from deviation in sample entropy when simple regression models are used and from deviation in volume and flow values when multiple regression model is used, respectively. In order to predict number of zombies, several models are developed using various regression techniques. For each regression model, we have calculated various statistical performance measures. Based on the statistical measures, we found that multiple regression based model performs better than any other model explored in this study. Therefore, the total number of predicted zombies using multiple regression model are close to actual number of zombies.

CHAPTER 6

ANN BASED SCHEME TO PREDICT NUMBER OF ZOMBIES IN A DDOS ATTACK

In the previous chapter, we discussed the importance of predicting number of zombies. A real time estimation of the number of zombies in DDoS scenario is helpful to suppress the effect of attack by choosing predicted number of most suspicious attack sources for either filtering or rate limiting. Because of this importance, we have taken a step further to the concept of predicting number of zombies. In this chapter, we discuss how feed forward neural networks of different sizes (i.e. architectures) are used to estimate number of zombies involved in a DDoS attack, and compare its performance with the method proposed in previous chapter. The sample data used to train the feed forward neural networks is generated using network simulator running on Linux platform. The generated sample data is divided into training data and test data and mean square error (MSE) is used to compare the performance of various feed forward neural networks. Various sizes of feed forward networks are compared for their estimation performance. The generalization capacity of the trained network is promising and the network is able to predict number of zombies involved in a DDoS attack more efficiently.

6.1 Introduction

Artificial Neural Network (ANN) is interconnection of massively parallel computing elements which are effective in nonlinear estimation. Several authors have used ANN in anomaly based DDOS attack detection. In [185] ANN is used to classify a network while under attack. In their implementation, data extracted in a network probing phase is fed to a feed forward neural network and it is trained to output 1 when there is attack and 0 when there is no attack. In [94], feed forward neural network is used to detect different DoS attacks. Recently [80] have proposed an approach to enhance the detection capacity of ANN. In all the above approaches, ANN is trained using normal and attack traffic data and ANN decides the presence or absence of an attack. In our approach, ANN is used to decide the number of

zombies used in a DDoS attack. To measure the performance of the proposed approach, we have calculated mean square error (MSE) and test error. Training and test data are generated using simulation. DDoS attacks are launched with varied number of zombies and the data collected through simulation is used to train the neural network. In our simulation experiments, attack traffic rate is fixed to 25Mbps in total; therefore, mean attack rate per zombie is varied from 0.25Mbps to 2.5Mbps and total number of zombie machines used to generate attack traffic range between 10 and 100. Various sizes of feed forward neural networks are compared for their estimation performance. The result obtained is very promising as we are able to predict number of zombies involved in DDoS attack effectively.

6.2 Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) [31, 200] is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true for ANNs as well. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages include:

- a) Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
- b) Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.

- c) Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
- d) Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage

A. Operation of a single artificial neuron

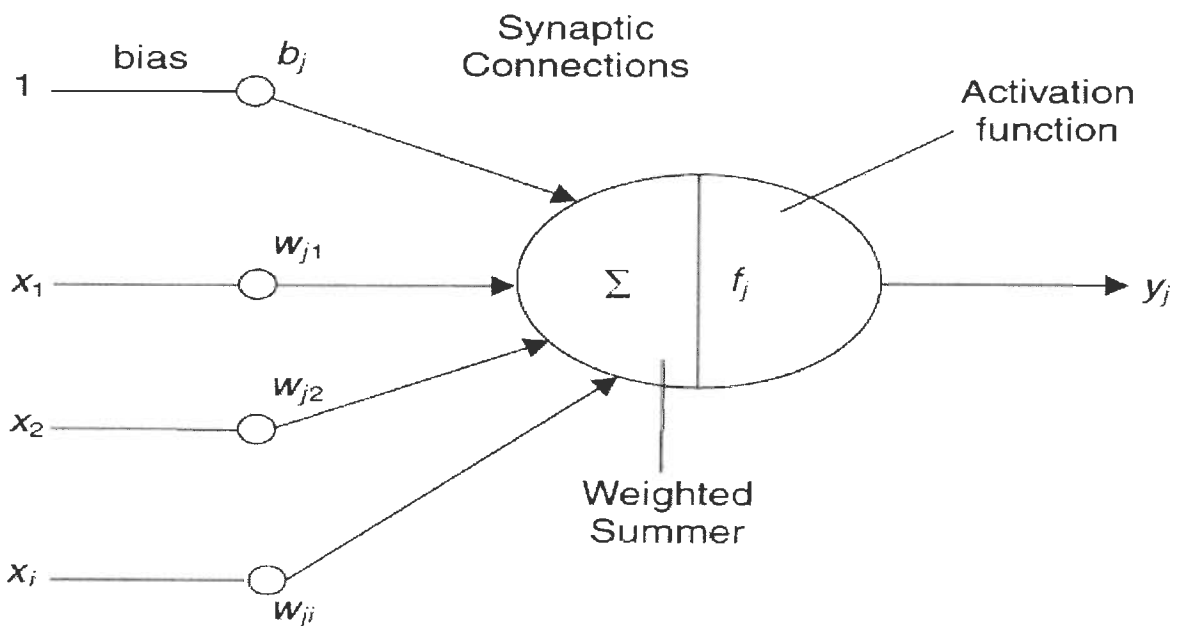


Figure 6.1. Operation of single neuron

In the single neuron shown above in figure 6.1,

- $x_1, x_2, x_3 \dots x_i$ represent the inputs to the neuron
- b_j and w_{ji} represent the connection weights to the individual inputs
- The summation units calculates the weight sum of the inputs
- The activation function f_j calculates the output of the neuron

Mathematically the output is given as:

$$y_j = f_j(w_{ji}X_i + b_j) \tag{6.1}$$

The activation function determines the type of neuron and the application where the neuron is to be used. But the sigmoid activation function as shown in figure 6.2, is famous for most neural networks and is given by

$$f_i s = \frac{1}{1 + e^{-s_j}} \tag{6.2}$$

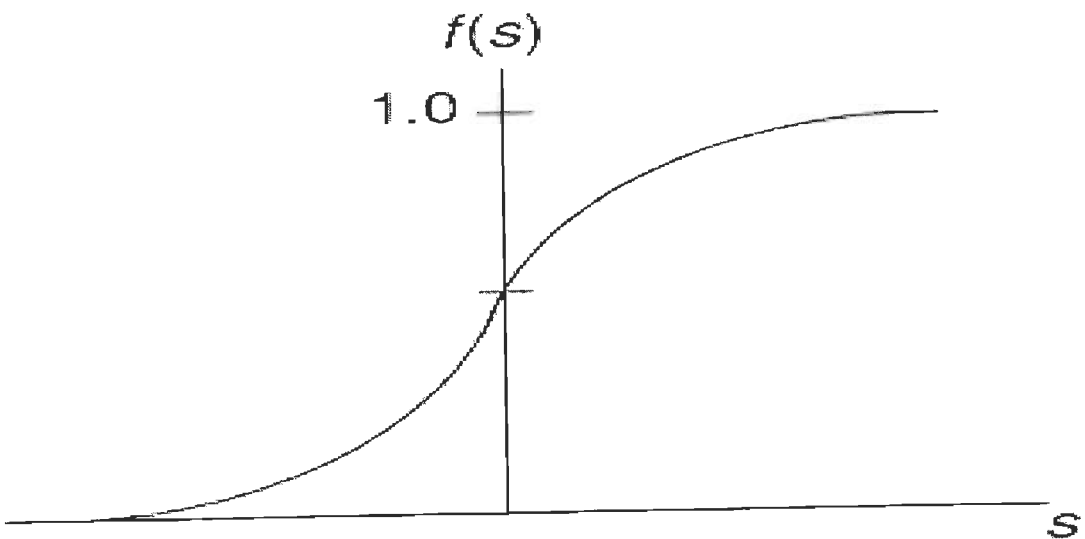


Figure 6.2. Sigmoid function

B. Network Architecture

Artificial neural networks (ANNs) are interconnections of individual neurons. There are various network architectures based on the type of connection. A most important type of network is the feed forward neural network shown in figure 6.3. It is a three layer neural network: an input layer with three inputs, a hidden layer with four neurons and an output layer with two neurons. This three layer network with enough number of hidden layer neurons and sigmoid activation function has the capacity to learn any nonlinear mapping. The neurons in the layers are interconnected by strength called weights. The name feed forward is given to this network because signal flows in forward manner without any feedback.

C. Learning in neural networks

Learning in the context of neural networks is the process of adjusting the connection weights and biases such that for a given input a desired output is achieved. There are two basic training modes.

- i) Supervised learning – This is a learning paradigm where the neural network is given samples of the input and desired output and the error between the desired output and the actual output of the neural network is used to adjust the connection weights. A famous algorithm of supervised learning is back propagation.
- ii) Unsupervised learning – This does not need any feedback for adjustment of the weights.

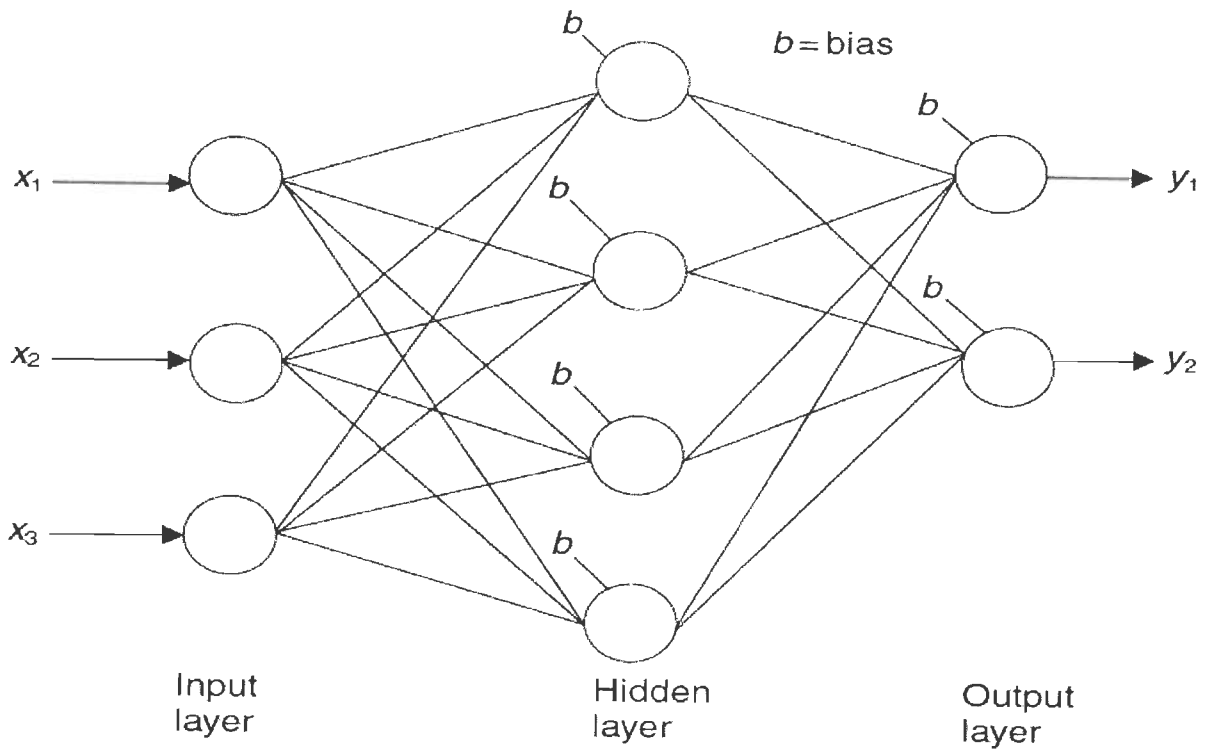


Figure 6.3. Fully connected, three layers feed forward network

The output y_k of the feed forward neural network is generated as shown in the following equation:

$$y_k = f_{ko} \left(\sum_{j=1}^N W_{jk}^h f_k \left(\sum_{i=1}^p W_{ji}^l X_i + b_j \right) + b_k \right) \quad (6.3)$$

where

- W_{jk}^h is connection weight from hidden layer to output
- W_{ji}^l is connection weight from input to hidden layer
- b_o is bias of output
- b_j is bias of hidden layer
- f_k is activation function of hidden layer, and
- f_o is activation function of output layer

D. Back propagation algorithm

Back propagation is a famous algorithm used to train neural networks. It uses the gradient decent optimization method to train a network. In most cases the sum of squared error is used as objective function.

$$J = \frac{1}{M} \sum_{j=1}^M (d_j - y_j)^2 \quad (6.4)$$

where d_j and y_j are the desired and actual network outputs.

Weight update algorithm is given by:

$$W_{\text{new}} = W_{\text{old}} + \Delta W \quad (6.5)$$

$$\text{where } \Delta W = -\mu \frac{\partial J}{\partial w}$$

E. Input and Output

In feed forward neural network, a relationship is developed between number of zombies Y (output) and observed deviation in sample entropy X as input. Here X is equal to $(H_c - H_n)$. Our proposed feed forward neural network based approach utilizes this deviation in sample entropy X to predict number of zombies.

6.3 Experimental Setup and Performance Analysis

The topology and simulation parameters discussed in chapter 3 are used again in this work. However, the simulation experiments are done in different manner i.e. earlier the number of zombies was kept constant, but in this case attack strength is kept constant and

number of zombies is varied. The simulations are repeated and different attack scenarios are taken by varying total number of zombie machines and at fixed attack strengths.

6.4 Results and Discussion

A. Training Data generation

Neural network has to be trained by giving sample inputs and corresponding output values and a training algorithm will adjust the connection weight and bias values until a minimum error or other stopping criteria is reached. The training data has to be taken carefully to consider the complete input range. Normalization and other preprocessing of the data improve the training performance.

In this chapter, in order to predict number of zombies (\hat{Y}) from deviation ($H_c - H_n$) in entropy value, training and testing data samples are generated using simulation experiments in NS-2 network simulator. Simulation experiments are done at the same attack strength 25Mbps in total and varying number of zombies from 10 to 100. We have performed several experiments of ANN model development on a number of data sets. The sample datasets for training and testing out of a whole dataset are shown in Table 6.1 and Table 6.2 , respectively.

B. Network Training

For the prediction of the number of zombies in a DDoS attack, three feed forward neural networks have been tested. The feed forward networks used have different sizes. The size of a network refers to the number of layers and the number of neurons in each layer. There is no direct method of deciding the size of a network for a given problem and one has to use experience or trial error method.

Table 6.1. Training Data-Deviation in entropy with actual number of zombies

| Actual Number of Zombies (Y) | Deviation in Entropy (X) |
|------------------------------|--------------------------|
| 10 | 0.045 |
| 15 | 0.046 |
| 25 | 0.050 |
| 30 | 0.068 |
| 35 | 0.087 |
| 40 | 0.099 |
| 45 | 0.111 |
| 55 | 0.130 |
| 60 | 0.139 |
| 65 | 0.148 |
| 75 | 0.163 |
| 80 | 0.170 |
| 85 | 0.176 |
| 90 | 0.182 |
| 100 | 0.192 |

Table 6.2. Testing Data-Deviation in entropy with actual number of zombies

| Actual Number of Zombies (Y) | Deviation in Entropy (X) |
|------------------------------|--------------------------|
| 20 | 0.048 |
| 50 | 0.121 |
| 70 | 0.157 |
| 95 | 0.189 |

In general, when a network is large, the complexity of the function that it can approximate will also increase. But as the network size increases, both training time and its

implementation cost increase and hence optimum network size has to be selected for a given problem. For the current problem, feed forward networks with 5, 10 and 15 hidden layer neurons are compared. The training algorithm used is the Levenberg-Marquardt back propagation algorithm of MATLAB's neural network toolbox. The training results are given in Table 6.3. Figure 6.4 shows the training performance of the feed forward networks used.

Table 6.3. Training results of various feed forward networks

| <i>Network Used</i> | <i>Network Size</i> | <i>Number of Epochs</i> | <i>MSE in training</i> |
|----------------------|---------------------|-------------------------|------------------------|
| Feed forward Network | 5-1 | 400 | 6.86 |
| | 10-1 | 400 | 0.36 |
| | 15-1 | 400 | 0.0025 |

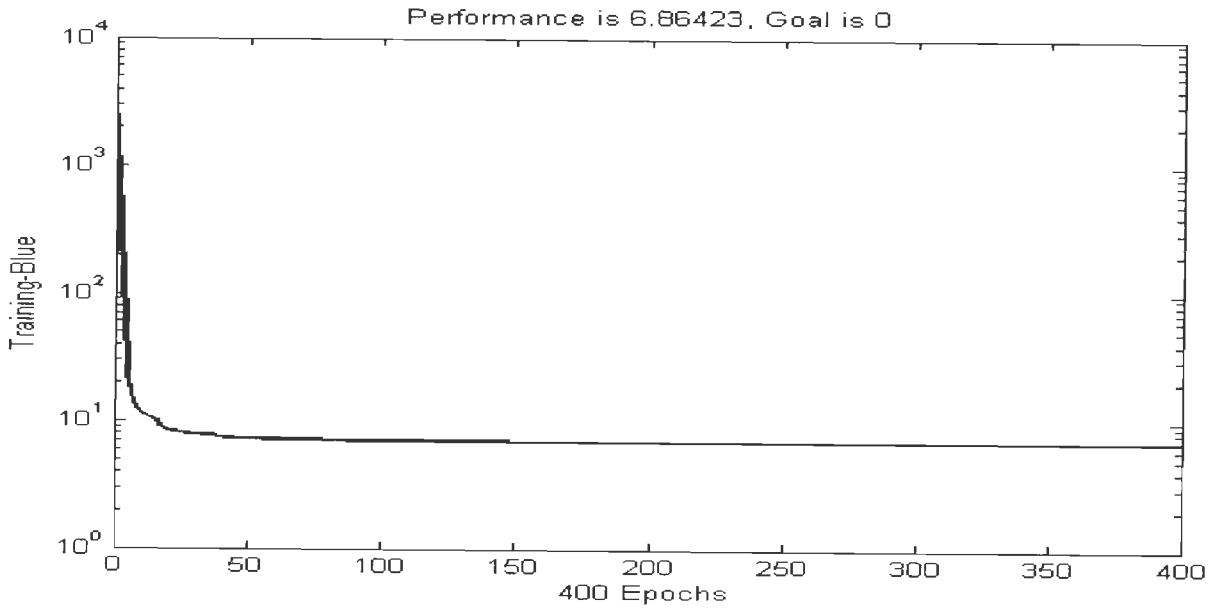


Figure 6.4 Training performance of feed forward network (5-1)

C. Network Testing

Table 6.4 shows the result of the testing of the networks using the test data values given in Table 6.2.

Table 6.4. Test results of various feed forward networks

| <i>Network Used</i> | <i>Network Size</i> | <i>MSE in Testing</i> |
|----------------------|---------------------|-----------------------|
| Feed forward Network | 5-1 | 2.91 |
| | 10-1 | 2.59 |
| | 15-1 | 3.14 |

From the result of table 6.3, we can see that the MSE in training decreases linearly as the network size increase. This is as expected. But in table 6.4, we can see that in spite of the smaller MSE in training and the increase in network size, the test result for the feed forward network having 15 hidden layer neurons is greater than the networks having 5 and 10 neurons. One reason for this is, for a good network performance, the ration of number of tunable parameters to that of training data size has to be very small. Here network size has increased but training data size has remained same. For the last network, the number of tunable parameters are 31 and ration is 1.63. Due to this, over fitting has occurred and the generalization performance of the last network is poor though it has good training performance. The training performance is measured using the mean square error (MSE). MSE is the difference between the actual and the neural network's estimated output. So, the best MSE is the closest to 0. If MSE is 0, this indicates neural network's output is equal to the actual value, which is the best situation.

Numbers of zombies of the individual networks have been compared with actual number of zombies for each test data values of table 6.2 and the results are given in figures 6.5, 6.6 and 6.7.

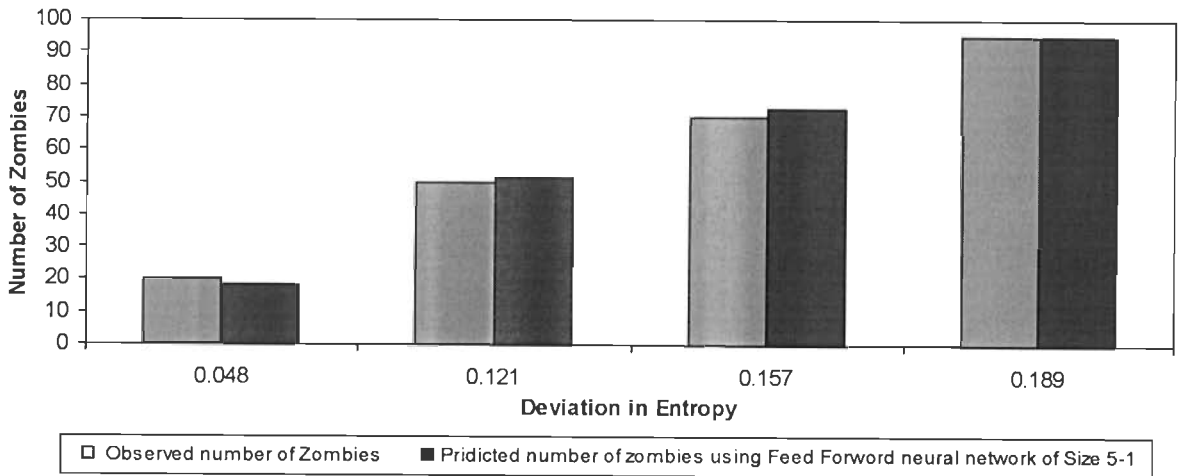


Figure 6.5. Comparison between actual number of zombies and predicted number of zombies using feed forward neural network of size 5-1

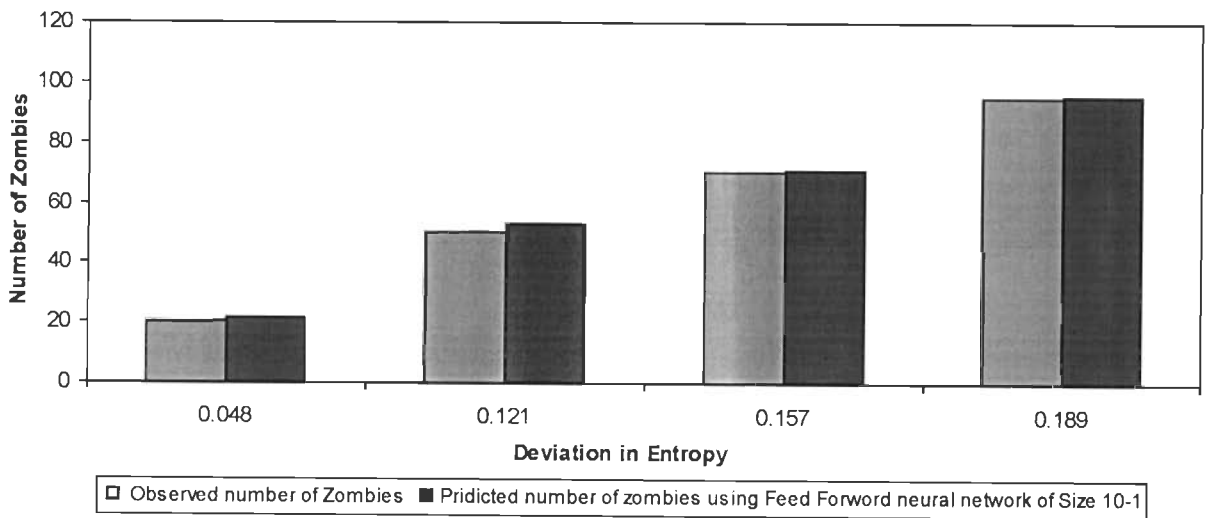


Figure 6.6. Comparison between actual number of zombies and predicted number of zombies using Feed forward neural network of size 10-1

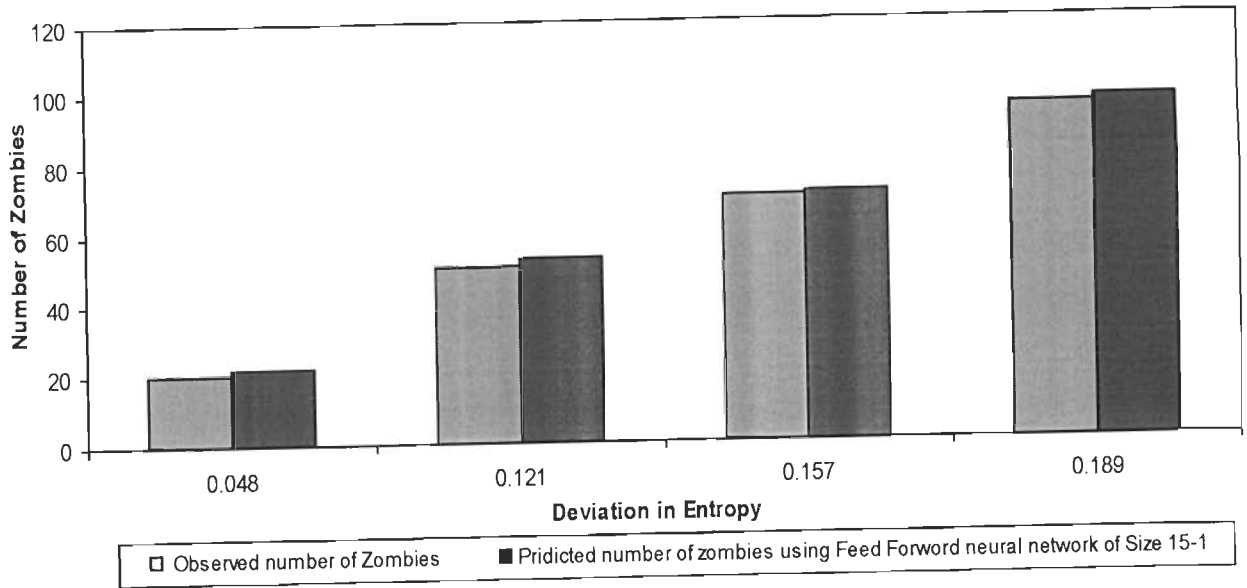


Figure 6.7. Comparison between actual number of zombies and predicted number of zombies using Feed forward neural network of size 15-1

To represent false positive i.e. falsely predicted normal clients as zombies and false negative i.e. zombies are identified as normal client, we plot test error. Positive cycle of test error curve represents false positive, while negative cycle represents false negative. The test error of the individual network is calculated for each test data values of table 6.2 and the results are given in table 6.5, 6.6 and 6.7. The results show that the prediction capacity of the neural networks is very close to the actual number of zombies and hence neural networks have the potential to be used to predict number of zombies in real DDoS attack scenarios.

Table 6.5. Summary of test error for feed forward neural network for network size 5-1

| (X) Entropy Variation | (Y) Number of Zombies | test error |
|-----------------------|-----------------------|------------|
| 0.048 | 20 | -1.79 |
| 0.121 | 50 | 1.31 |
| 0.157 | 70 | 2.59 |
| 0.189 | 95 | -0.07 |

Table 6.6. Summary of test error for feed forward neural network for network size 10-1

| (X) Entropy Variation | (Y) Number of Zombies | test error |
|-----------------------|-----------------------|------------|
| 0.048 | 20 | 1.33 |
| 0.121 | 50 | 2.88 |
| 0.157 | 70 | 0.40 |
| 0.189 | 95 | 0.36 |

Table 6.7. Summary of test error for feed forward neural network for network size 15-1

| (X) Entropy Variation | (Y) Number of Zombies | test error |
|-----------------------|-----------------------|------------|
| 0.048 | 20 | 1.88 |
| 0.121 | 50 | 2.20 |
| 0.157 | 70 | 0.85 |
| 0.189 | 95 | 1.86 |

6.5 Comparison between Regression and ANN based schemes for Predicting number of Zombies in a DDoS attack

In this section, performance of regression based scheme, proposed in the previous chapter for predicting number of zombies in a DDoS attack, is compared with ANN based scheme proposed in this chapter. The comparison is done for randomly taken sample data values shown in table 6.8 below. The criteria used for the comparison is absolute prediction error.

Table 6.8. Comparison between Regression and ANN based schemes for predicting number of zombies in a DDoS attack

| Scheme used | Actual number of zombies | Predicted number of zombies | Absolute error |
|-------------------------|--------------------------|-----------------------------|----------------|
| ANN based scheme | 20 | 21.33 | 1.33 |
| | 50 | 52.88 | 2.88 |
| | 70 | 70.4 | 0.4 |
| | 95 | 95.36 | 0.36 |
| Regression based scheme | 20 | 19.95 | 0.05 |
| | 50 | 49.63 | 0.37 |
| | 70 | 70.30 | 0.3 |
| | 95 | 96.04 | 1.04 |

Table 6.8 shows actual number of zombies, predicted number of zombies and values of absolute error, when regression and ANN based schemes are used. Figure 6.8 presents the comparison of absolute error using ANN and regression based scheme. As shown in figure 6.8, in regression based scheme, value of absolute error increases when total number of zombies increase. But in ANN based scheme, error value decreases when total number of zombies increase. This shows that when attack is serious i.e. total number of zombies performing attack is more, prediction performance of ANN based scheme is better than regression based scheme.

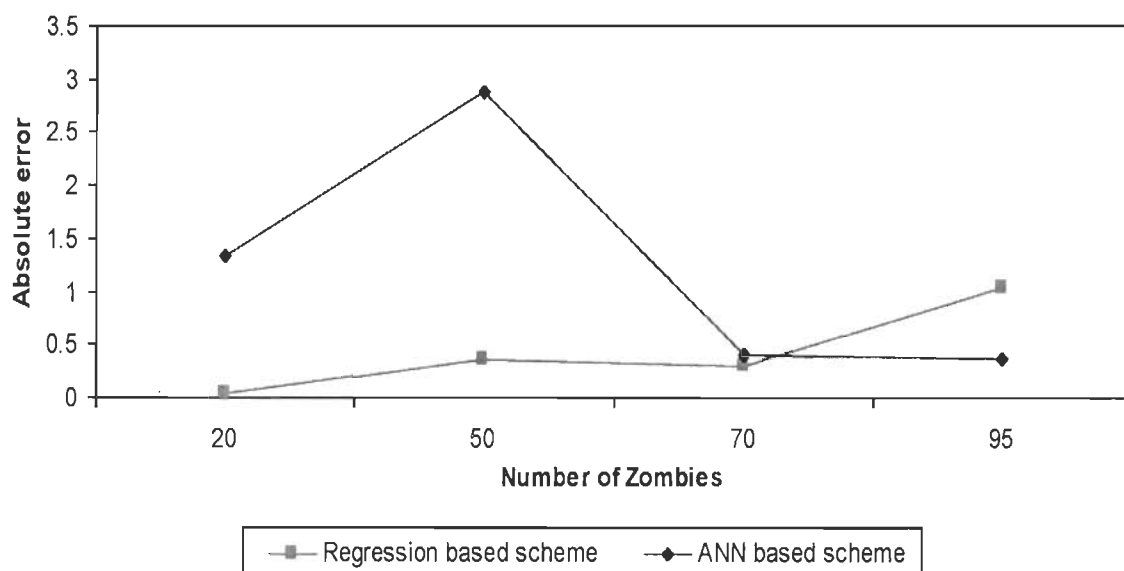


Figure 6.8. Comparison of Absolute error using ANN and Regression based scheme

6.6 Chapter Summary

The potential of feed forward neural network for predicting number of zombies involved in a flooding DDoS attack is investigated. The deviation in sample entropy is used as an input and MSE is used as the performance measure. Feed forward networks with hidden neurons 5, 10 and 15 have shown maximum mean square error (MSE) of 2.91, 2.59 and 3.14 respectively, in predicting the number of zombies. Various sizes of feed forward networks are compared for their estimation performance. The simulation results show that feed forward neural networks with 10 neurons in hidden layer performs best and is able to predict number of zombies in a DDoS attack efficiently. Prediction performance of ANN based scheme is compared with regression based scheme. It can be concluded that the selection of ANN and regression based scheme is based on the severity of attack. When attack is more severe, ANN based scheme performed better than regression based scheme and if attack is not much severe, regression based scheme performed better than ANN based scheme. However, simulation results are promising as we are able to predict number of zombies efficiently, experimental study using a real time test bed can strongly validate our claim.

CHAPTER 7

ESTIMATING STRENGTH OF A DDoS ATTACK USING REGRESSION MODELS

This chapter presents an approach for estimating strength of a DDoS attack using various regression models. Strength of a DDoS attack is the rate of attack traffic which is coming to the victim. Estimating strength of attack is helpful to suppress the effect of attack, as it enables a security administrator to effectively equip his arsenal with proper defense mechanisms for fighting against DDoS threat according to the strength of attack. If attack strength is very high, i.e. attack is very serious, best available defense mechanisms can be used to handle the attack; otherwise other cheaply available defense mechanisms can be used. Hence, in this chapter, to estimate strength of attack, a relationship is established between strength of attack and observed deviation in sample entropy and strength of attack and observed deviation in volume and flow for simple and multiple regression, respectively. Various statistical performance measures are used to evaluate the performance of several regression models. Internet type topologies used for simulation are generated using Transit-Stub model of GT-ITM topology generator. NS-2 network simulator on Linux platform is used as simulation test bed for launching DDoS attacks with varied attack strength. A comparative study is performed using different regression models for estimating strength of DDoS attack. The simulation results are promising as we are able to estimate strength of DDoS attack efficiently using various regression models. The simulation results show that multiple regression models are most suitable for estimating strength of DDoS attacks.

7.1 Introduction

Kumar [118] in his research work has discussed the advantage of estimating strength of DDoS attack. Kumar mentioned that estimating strength of DDoS attack can be helpful to suppress the effect of attack. Our objective is to find the relationship between strength of DDoS attack and deviation in sample entropy and strength of DDoS attack and observed deviation in volume and flow for simple and multiple regression, respectively. In order to estimate strength of DDoS attack, several models are developed using various regression

techniques. A comparative study is performed between different regression models for estimating strength of a DDoS attack.

7.2 Regression Models

In estimating the strength of attack, various regression models have been utilized. As described in chapter 5, two different types of regression models are employed. The first one is simple regression model and the second one is multiple regression model. Simple regression model establishes a relation between two variables only, one dependent and one independent variable. In our analysis, we used five types of simple regression models, namely: linear, polynomial, logarithmic, power and exponential. In the case of multiple regression, there can be more than one independent variables. In our analysis, we used two independent variables. In the model development, curve fitting using mean square error minimization is used.

7.3. Statistical Performance Measures

Statistical performance measures are important in analyzing the suitability of a regression model for a given problem. Accordingly, various performance measures such as, coefficient of determination, coefficient of correlation, sum of square error, mean square error, normalized mean square error and Nash–Sutcliffe efficiency index are used to evaluate the performance of regression models explored in this study. The detail mathematical descriptions for these performance measures have been given in chapter 5. As stated there, a regression model is best suited if it has highest value of coefficient of determination and coefficient of correlation and Nash-Sutcliffe efficiency index. Moreover, an efficient model should have lowest value of sum of square error, mean square error, and normalized mean square error.

7.4. Simulation Setup

The topology and simulation parameters discussed in chapter 3 are used again in this work. As described in chapter 3, the topology considered is similar to the one used traditionally in the Internet for simulation and validation purposes. A total of 400 legitimate client machines are used to generate background traffic. One FTP server is used to provide service to the clients. All FTP requests are originated randomly from different nodes. The simulation parameters used in this experiment are shown in table 7.1. As we can see from the

table, simulation runs for 60 seconds and DDoS attacks start at 25th second and end at 50th second. The simulations are repeated and deviation in entropy and volume/flow for simple and multiple regression, respectively are calculated for different attack strengths and at fixed total number of zombie machines, i.e. 100. Figure 7.1 shows entropy variation with time for attack strength 10Mbps-100Mbps. Figure 7.2 and figure 7.3 show flow and volume variation with time for attack strength 10Mbps-100Mbps.

Table 7.1. Simulation parameters

| S. No. | Parameter | Value |
|--------|-------------------------------|---------------|
| 1. | Simulator | ns-2 |
| 2. | Traffic arrival process | Poisson |
| 3. | Simulation time | 60 seconds |
| 4. | Attack Duration | 25-50 seconds |
| 5. | Number of legitimate clients | 100-400 |
| 6. | Number of attackers | 100 |
| 7. | Polling interval | 200ms |
| 8. | Packet size | 1040 bytes |
| 9. | Tolerance factor α | 1-10 |
| 10. | Connection startup time | 1-8 seconds |
| 11. | Access link bandwidth | 1 Mbps |
| 12. | Backbone link bandwidth | 100Mbps |
| 13. | Backbone link delay | 0 seconds |
| 14. | Bottleneck link bandwidth | 310 Mbps |
| 15. | Mean attack rate per attacker | 0.1-1Mbps |

7.5. Model Development and Experimental Analysis

In this section, we describe our experiments to study the use of various regression models for estimating strength of a DDoS attack. For simple regression models, we collected deviation in entropy by varying total strength of DDoS attack from 10Mbps to 100Mbps and the data is shown in table 7.2. Similarly, for multiple regression model, volume and flow data

is collected by varying total strength of DDoS attack as shown in table 7.3. The inputs to the multiple regression model are strength of DDoS attack Y and observed deviation in sample volume X_1 and flow X_2 . In our simulation experiment, number of zombies is fixed to 100. Regression equations are determined through a process of curve fitting. The main objective in the process of the curve fitting is to minimize the error between the actual strength of DDoS attack and the predicted strength of DDoS attack. Figures 7.4 to 7.8 show the regression equation and coefficient of determination for simple regression models as discussed in section 7.2. Using these equations and deviation in entropy values, predicted strength of DDoS attack is calculated. In a similar fashion, using flow and volume as inputs to the multiple regression equation, predicted strength of DDoS attack is obtained. For the multiple regression, the regression equation is given in equation 7.1. The coefficient of determination for the multiple regression model is 0.97.

$$Y = X_1 * 0.00050 + X_2 * (-1.80) + 66.76 \tag{7.1}$$

where X_1 and X_2 represent deviation in sample volume and flow, respectively.

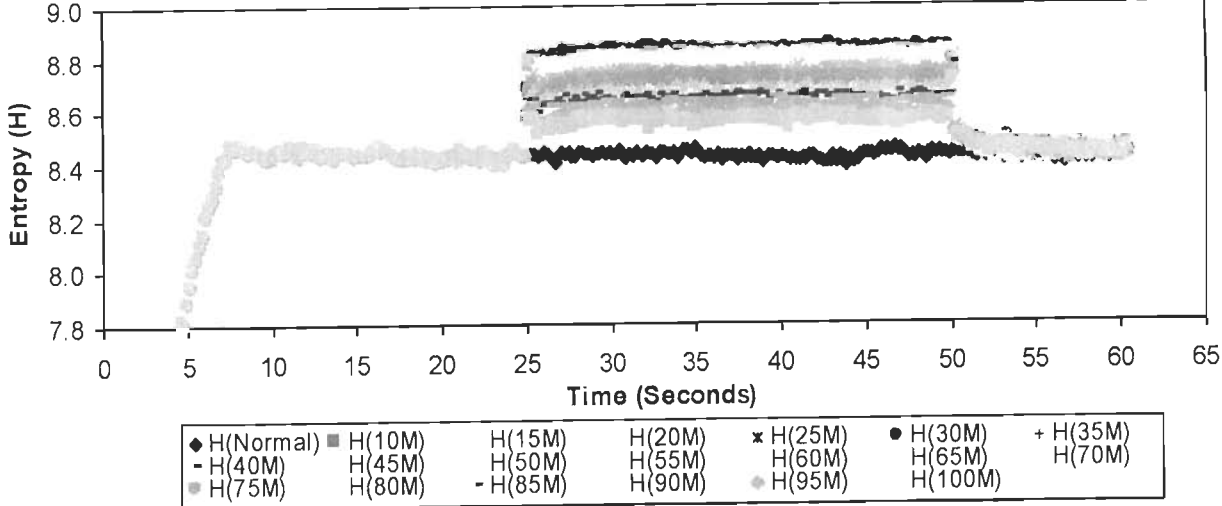


Figure 7.1. Entropy variation with varied attack strength

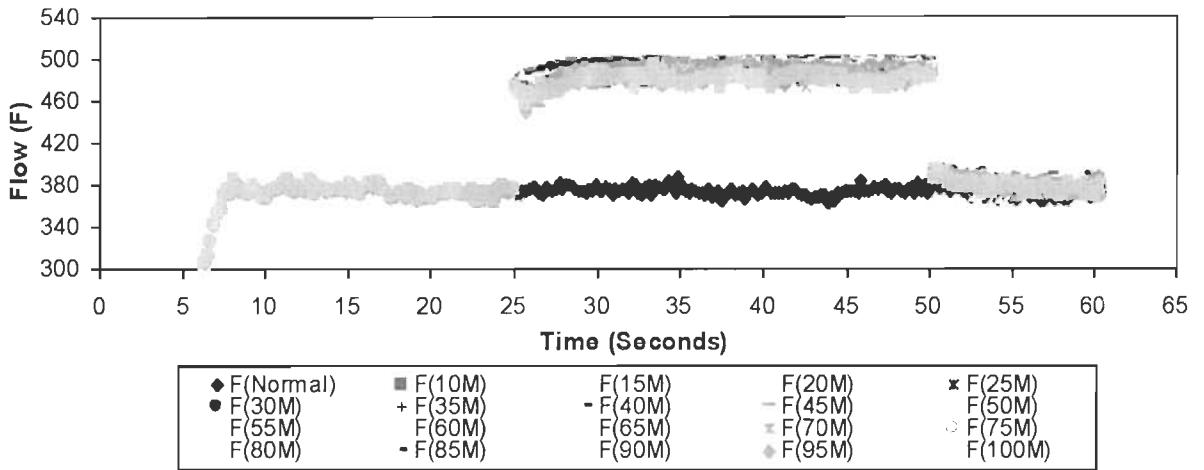


Figure 7.2. Flow variation with varied attack strength

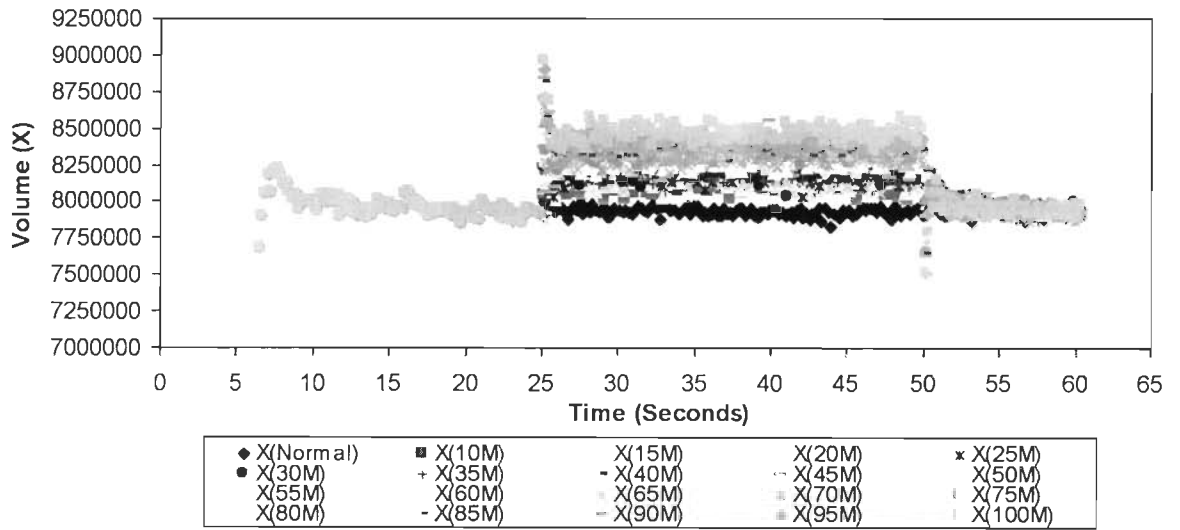


Figure 7.3. Volume variation with varied attack strength

Table 7.2 Deviation in entropy with actual strength of DDoS attack

| Actual strength of DDoS attack (Y) | Deviation in Entropy (X) |
|------------------------------------|--------------------------|
| 10M | 0.149 |
| 15M | 0.169 |
| 20M | 0.184 |
| 25M | 0.192 |
| 30M | 0.199 |
| 35M | 0.197 |
| 40M | 0.195 |
| 45M | 0.195 |
| 50M | 0.208 |
| 55M | 0.212 |
| 60M | 0.233 |
| 65M | 0.241 |
| 70M | 0.244 |
| 75M | 0.253 |
| 80M | 0.279 |
| 85M | 0.280 |
| 90M | 0.299 |
| 95M | 0.296 |
| 100M | 0.319 |

Table 7.3 Deviation in volume and flow with DDoS attack strength

| Attack Strength (Y) | Deviation in volume (X_1) | Deviation in Flow (X_2) |
|---------------------|-------------------------------|-----------------------------|
| 10M | 90855.56 | 59.96 |
| 15M | 109515.24 | 59.13 |
| 20M | 133721.59 | 59.37 |
| 25M | 143495.87 | 58.81 |
| 30M | 146886.67 | 59.10 |
| 35M | 144870.16 | 58.28 |
| 40M | 156592.38 | 57.23 |
| 45M | 160320.63 | 58.67 |
| 50M | 213209.52 | 56.64 |
| 55M | 178804.44 | 57.58 |
| 60M | 181885.71 | 57.61 |
| 65M | 187367.94 | 56.24 |
| 70M | 199750.48 | 57.48 |
| 75M | 209413.33 | 57.25 |
| 80M | 219707.62 | 56.82 |
| 85M | 227447.30 | 53.72 |
| 90M | 227771.75 | 55.23 |
| 95M | 249654.60 | 54.71 |
| 100M | 269721.59 | 53.09 |

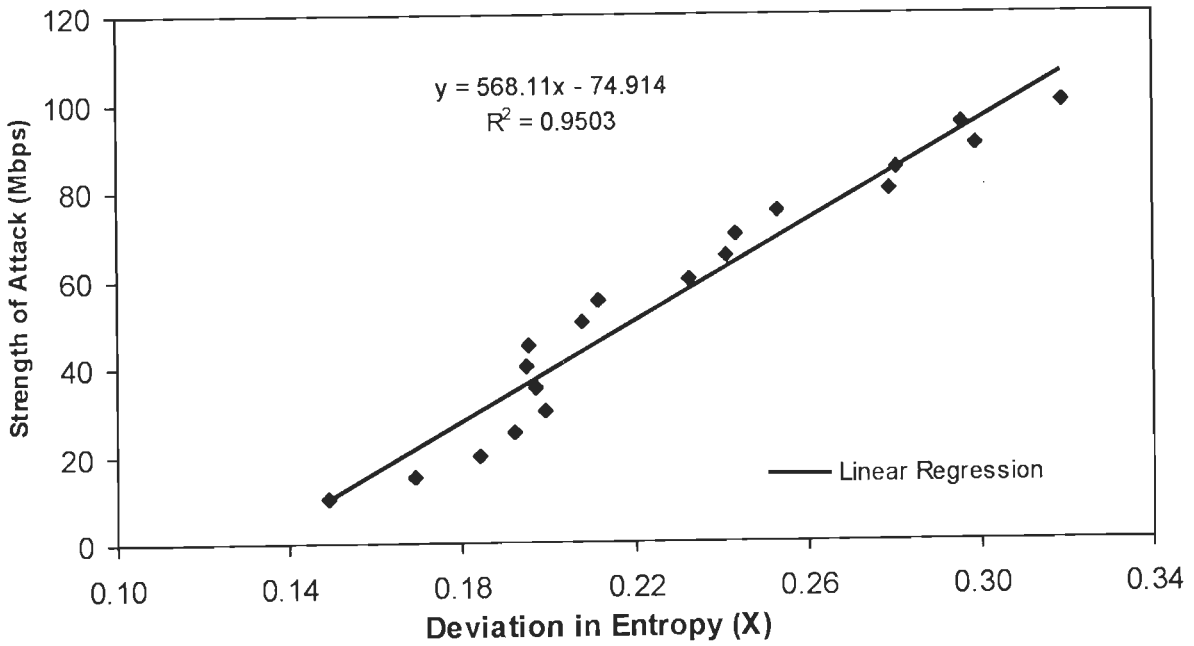


Figure 7.4. Regression equation and coefficient of determination for linear regression based model M1

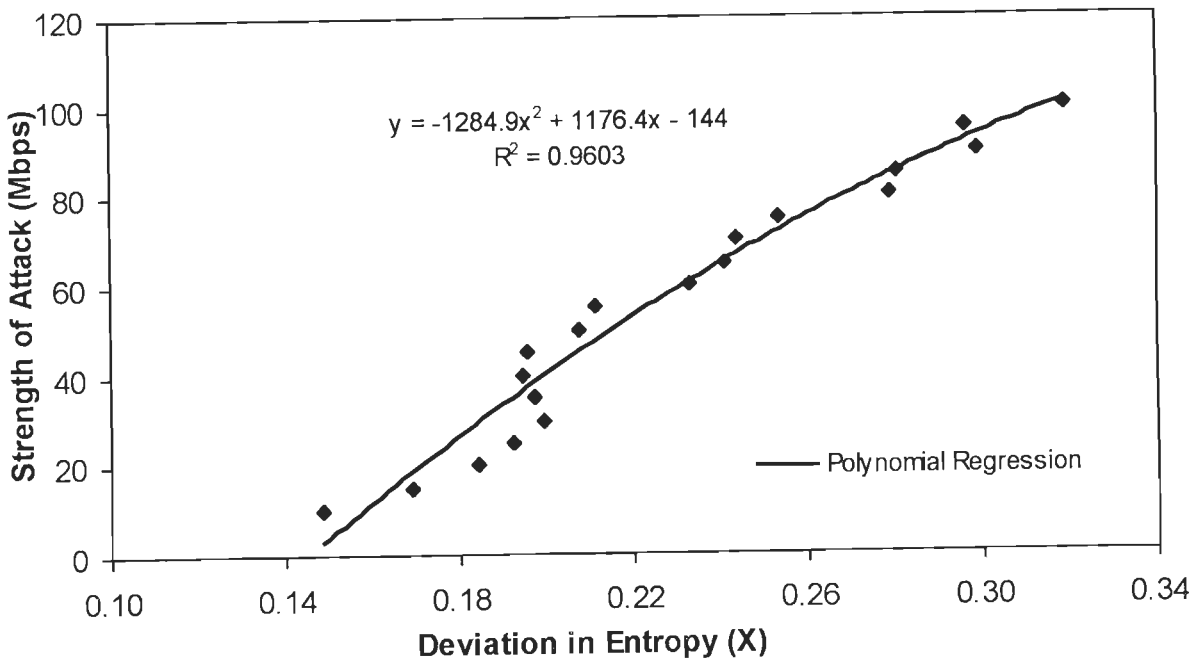


Figure 7.5. Regression equation and coefficient of determination for polynomial regression based model M2

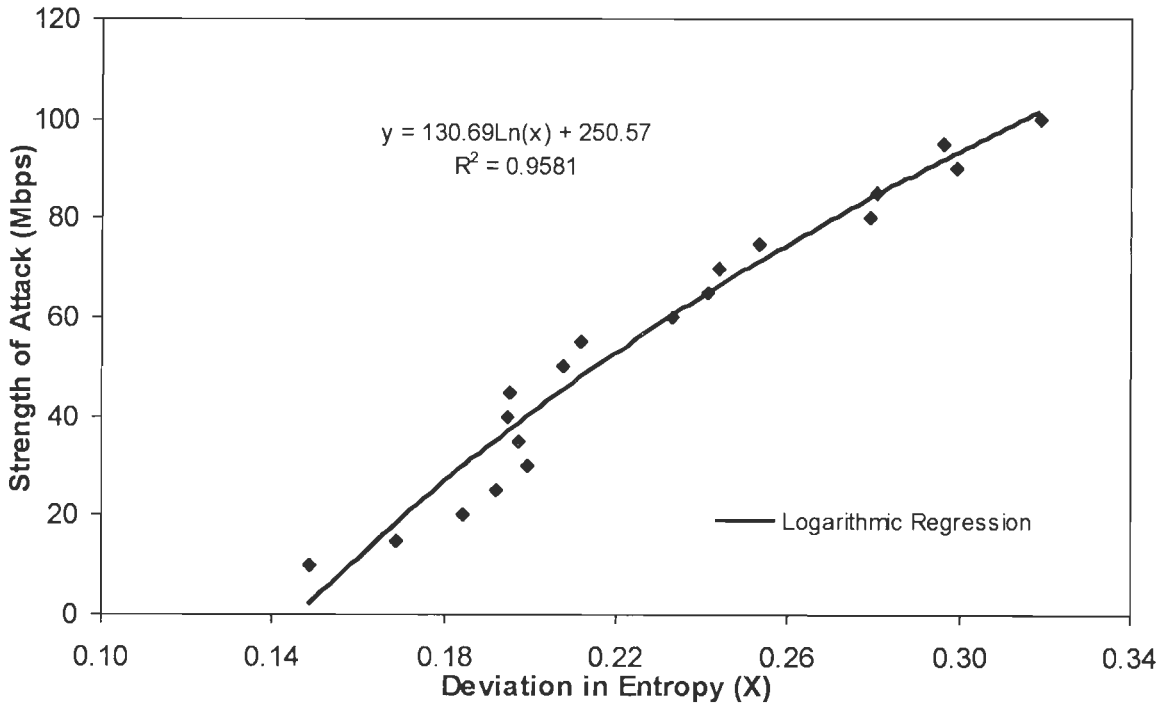


Figure 7.6. Regression equation and coefficient of determination for logarithmic regression based model M3

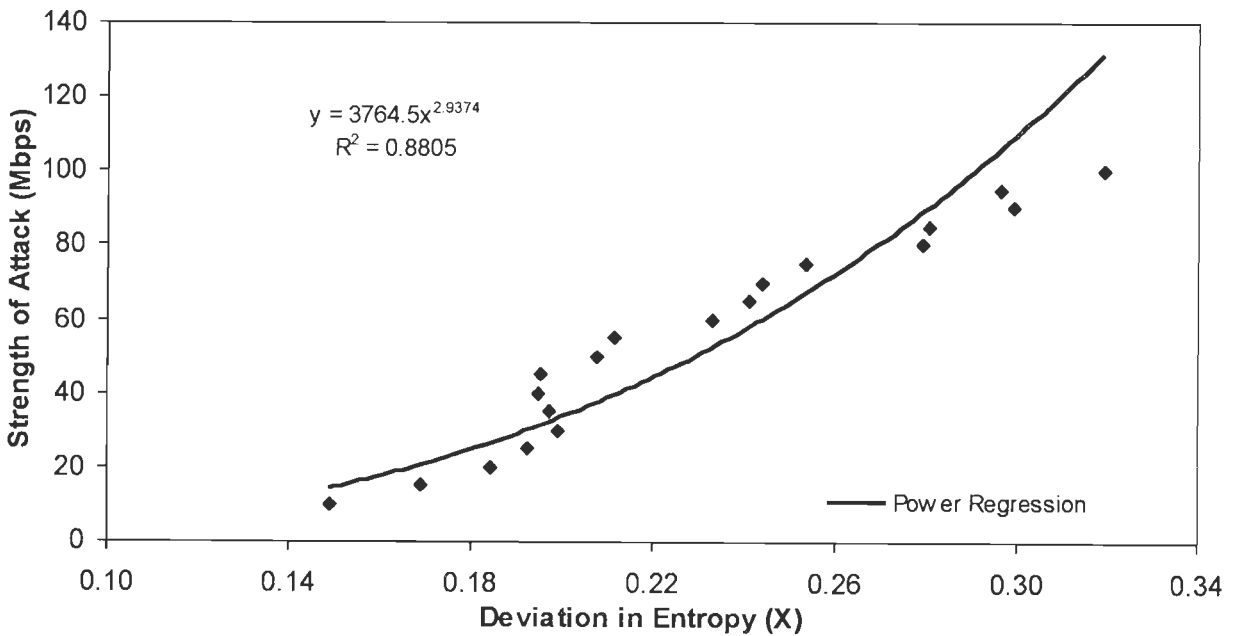


Figure 7.7. Regression equation and coefficient of determination for power regression based model M4

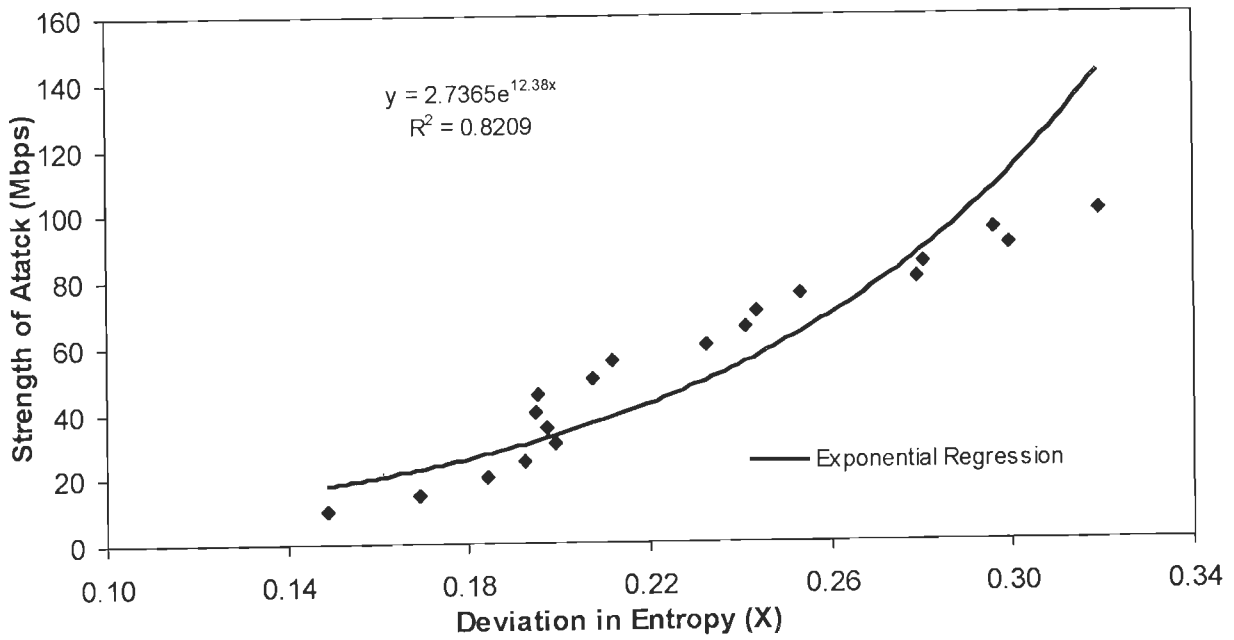


Figure 7.8. Regression equation and coefficient of determination for exponential regression based model M5

From the above figures, it can be inferred that among all basic regression models, the polynomial regression model has the best curve fitting. However, an exhaustive comparison of the suitability of both the basic and multiple regression models is discussed in the following section.

7.6 Results and Discussion

Here we give the results of the comparison of the simple and multiple regression models. For clarity of the presentation, first the simple regression models are separately compared and then the comparison of best found polynomial regression model with multiple regression is given.

A. Simple regression models

In this section, simulation results of models M1 to M5 given in section 7.5 are presented. The comparison between actual strength of DDoS attack and predicted strength of DDoS attack using various regression models is depicted in figures 7.9 to 7.13. Figure 7.14

shows comparison between actual strength of DDoS attack and predicted strength of DDoS attack using various regression models M1 to M5.

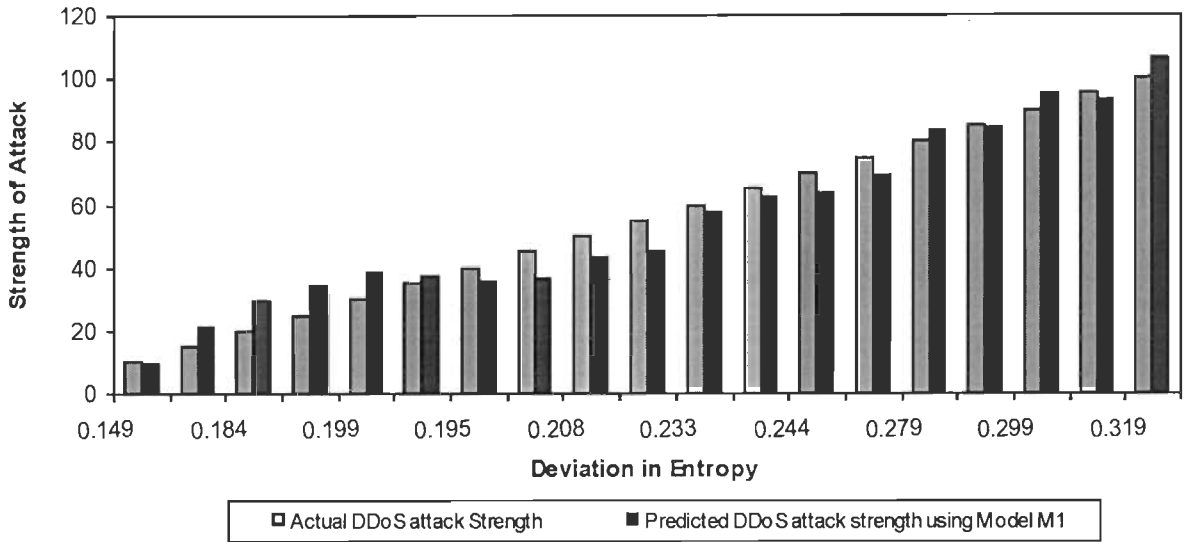


Figure 7.9. Comparison between actual Strength of DDoS attack and predicted Strength of DDoS attack using linear regression based model M1

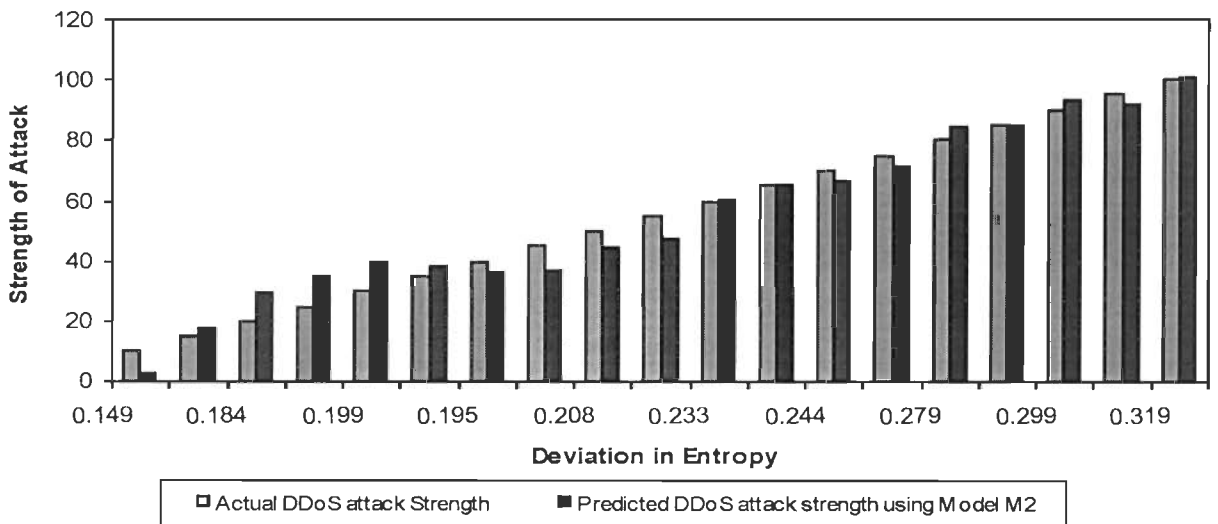


Figure 7.10. Comparison between actual Strength of DDoS attack and predicted Strength of DDoS attack using polynomial regression based model M2

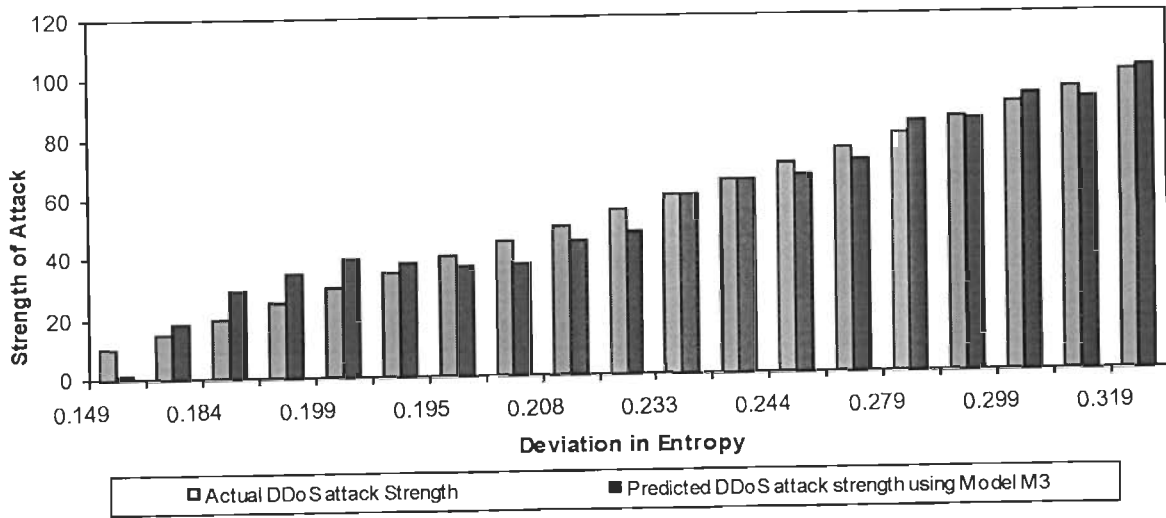


Figure 7.11. Comparison between actual Strength of DDoS attack and predicted Strength of DDoS attack using logarithmic regression based model M3

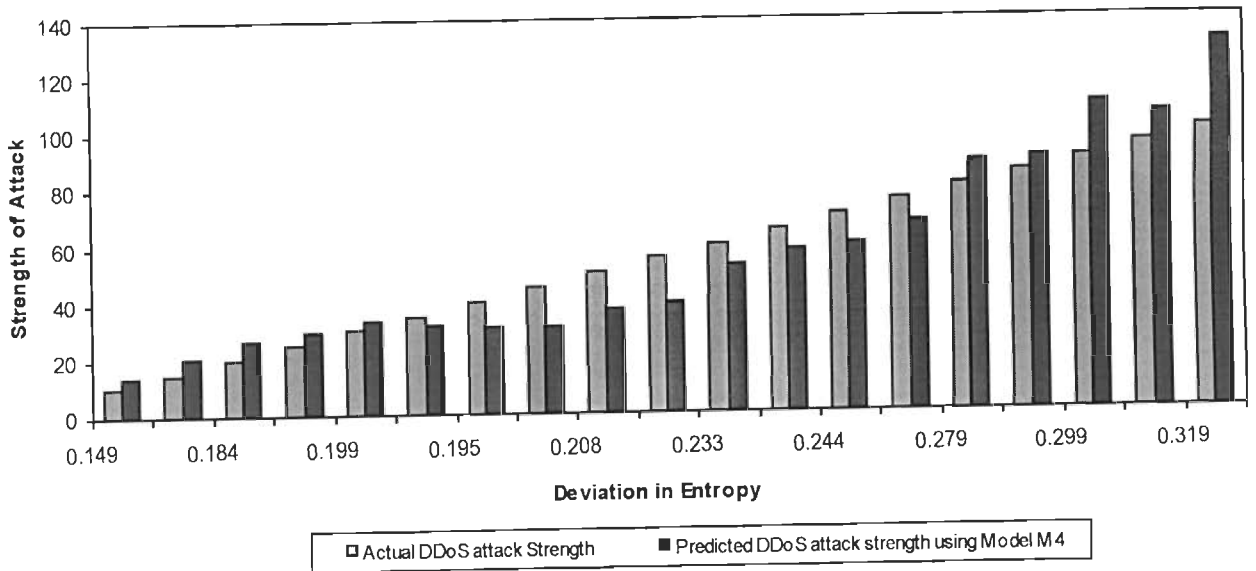


Figure 7.12. Comparison between actual Strength of DDoS attack and predicted Strength of DDoS attack using power regression based model M4

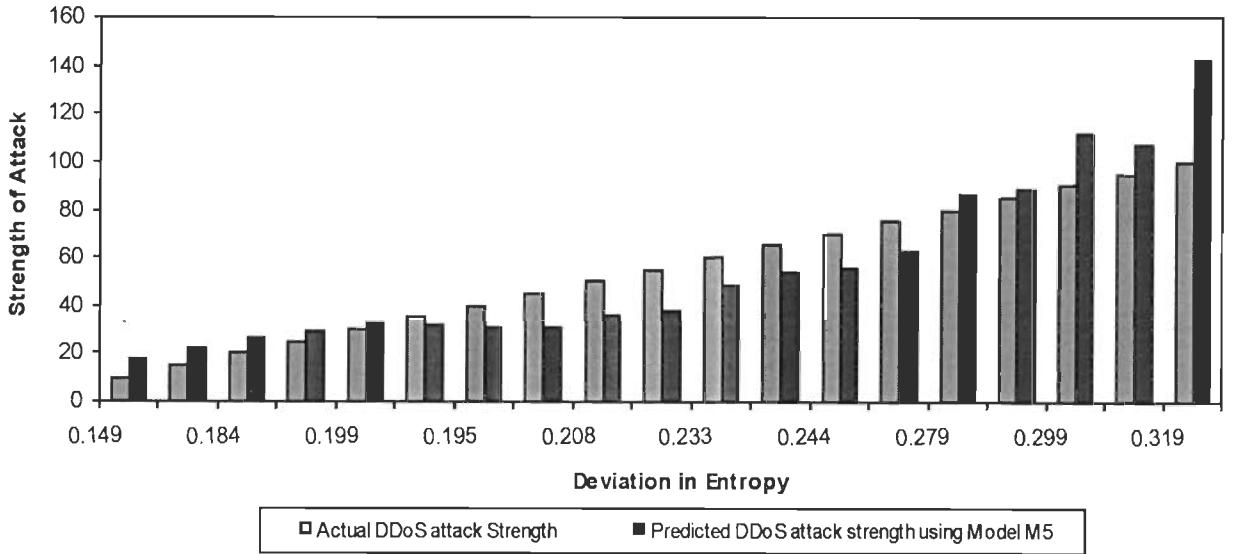


Figure 7.13. Comparison between actual Strength of DDoS attack and predicted Strength of DDoS attack using exponential regression based model M5

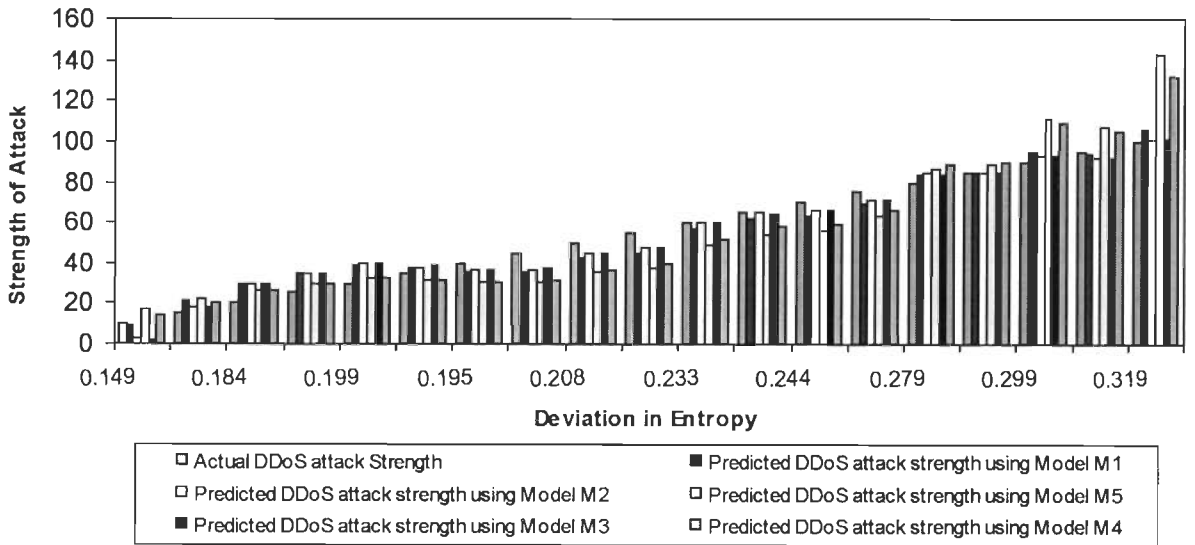


Figure 7.14. Comparison between actual Strength of DDoS attack and predicted Strength of DDoS attack using various regression models M1-M5

Table 7.4 shows residual error [118] for various regression models (M1 to M5). Figure 7.15 depicts summary of residual error in various regression models (M1 to M5).

Table 7.4 Summary of residual error for various regression models

| (X) Entropy Variation | Actual strength of DDoS attack (Y) | Residual error | | | | |
|-----------------------------|---|----------------|-------------|-------------|-------------|-------------|
| | | Model M1 | Model M2 | Model M3 | Model M4 | Model M5 |
| 0.149 | 10M | -0.37 | -7.37 | -8.40 | 3.99 | 7.29 |
| 0.169 | 15M | 6.10 | 3.14 | 3.22 | 5.32 | 7.20 |
| 0.184 | 20M | 9.77 | 9.17 | 9.52 | 6.20 | 6.82 |
| 0.192 | 25M | 9.28 | 9.67 | 10.03 | 4.65 | 4.59 |
| 0.199 | 30M | 8.30 | 9.44 | 9.76 | 2.98 | 2.30 |
| 0.197 | 35M | 2.20 | 3.15 | 3.48 | -2.95 | -3.47 |
| 0.195 | 40M | -4.23 | -3.54 | -3.19 | -9.14 | -9.43 |
| 0.195 | 45M | -8.90 | -8.15 | -7.80 | -13.87 | -14.21 |
| 0.208 | 50M | -6.96 | -5.11 | -4.88 | -12.80 | -14.19 |
| 0.212 | 55M | -9.70 | -7.57 | -7.41 | -15.67 | -17.38 |
| 0.233 | 60M | -2.69 | 0.24 | 0.04 | -7.98 | -11.13 |
| 0.241 | 65M | -2.90 | 0.03 | -0.31 | -7.25 | -10.75 |
| 0.244 | 70M | -6.49 | -3.60 | -3.97 | -10.48 | -14.06 |
| 0.253 | 75M | -6.02 | -3.41 | -3.90 | -8.30 | -11.97 |
| 0.279 | 80M | 3.62 | 4.29 | 3.76 | 8.65 | 6.72 |
| 0.280 | 85M | -0.61 | -0.09 | -0.60 | 4.93 | 3.19 |
| 0.299 | 90M | 5.04 | 3.01 | 2.85 | 18.74 | 21.21 |
| 0.296 | 95M | -1.73 | -3.27 | -3.52 | 10.45 | 12.00 |
| 0.319 | 100M | 6.32 | 0.61 | 1.25 | 31.34 | 42.21 |

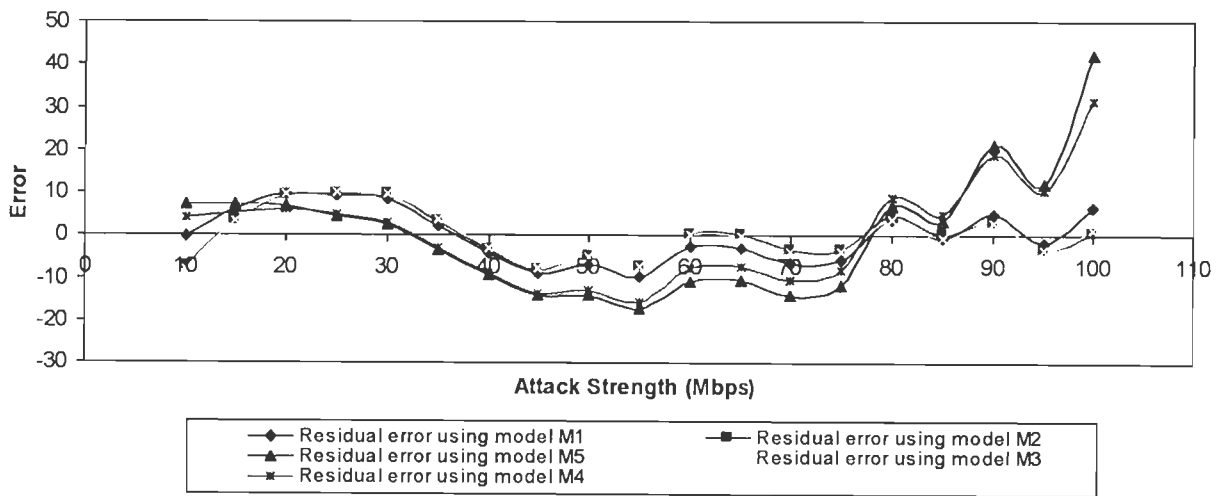


Figure 7.15. Summary of Residual error in various regression models

Table 7.5 shows summary of various performance measures calculated for simple regression models M1 to M5.

Table 7.5. Summary of various performance measures for simple regression model

| | Linear | Polynomial | Logarithmic | Power | Exponential |
|--------|---------------|-------------------|--------------------|--------------|--------------------|
| R^2 | 0.95 | 0.96 | 0.96 | 0.89 | 0.84 |
| CC | 0.97 | 0.98 | 0.98 | 0.94 | 0.92 |
| SSE | 708.13 | 566.31 | 596.96 | 2643.90 | 3995.70 |
| MSE | 37.27 | 29.81 | 31.42 | 139.15 | 210.30 |
| NMSE | 1.32 | 1.06 | 1.12 | 4.95 | 7.47 |
| η | 0.95 | 0.96 | 0.96 | 0.81 | 0.72 |

As described in chapter 5, coefficient of determination (R^2) is the proportion of variance in dependent variable which can be predicted from independent variable and CC is its square root. The Nash—Sutcliffe efficiency index (η) is a widely used and potentially reliable statistic for assessing the goodness of fit of models. On the other hand, values of SSE, MSE and NMSE quantify the error in the estimation using various regression models. Therefore, when comparing various regression models, we have to select a model with highest value of coefficient of determination, coefficient of correlation and η and lowest values of SSE, MSE and NMSE. Accordingly, it can be seen from table 7.5 that polynomial regression

based model M2 has highest value of coefficient of determination, coefficient of correlation and η and lowest values of SSE, MSE and NMSE. Thus, it can be concluded that it performs better than other models.

To compare the performance of the various regression models, the actual and estimated strength of attack are shown in figures 7.9 to 7.13. Observing the difference between the actual and predicted strength of attack in each figure, it can be verified that the polynomial regression based model M2 shown in figure 7.10 has the least difference between the actual and predicted strength of attack compared to other regression based models. The residual error for various regression models given in table 7.4 also verifies the same fact. Hence, we can conclude that, estimating strength of attack by polynomial regression model is closest to the actual strength of attack.

B. Multiple regression model

In this section, simulation results of the multiple regression model developed in section 7.5 are presented. The comparison between actual strength of DDoS attack and predicted strength of DDoS attack using multiple regression model is depicted in figures 7.16.

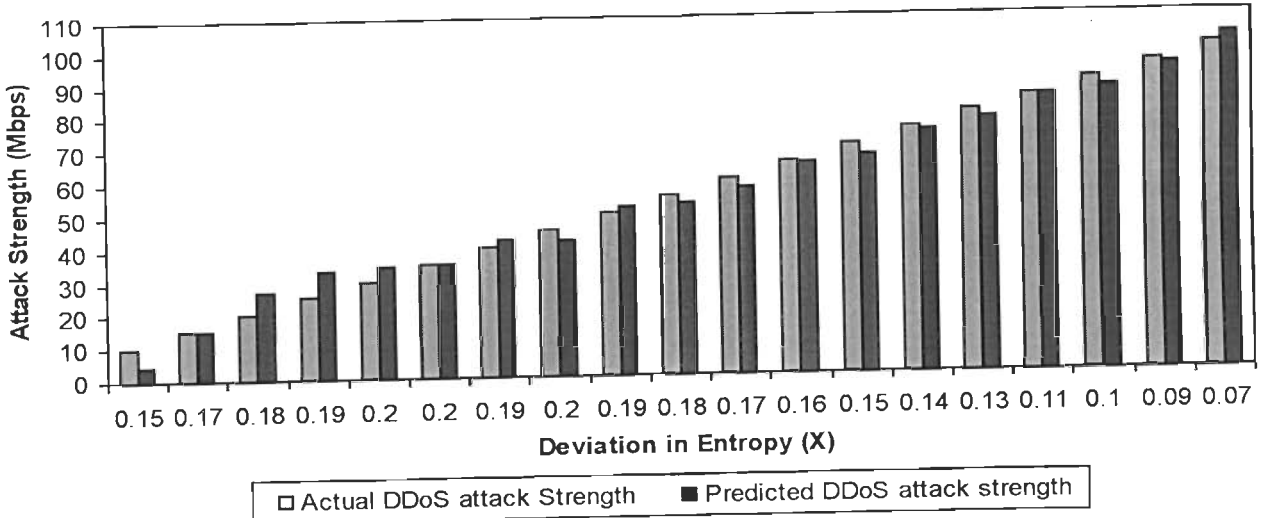


Figure 7.16. Comparison between actual DDoS attack strength and predicted DDoS attack strength using multiple regression model

Figure 7.17 represents residual error for multiple regression model. Table 7.6 shows values of various performance measures for multiple regression model.

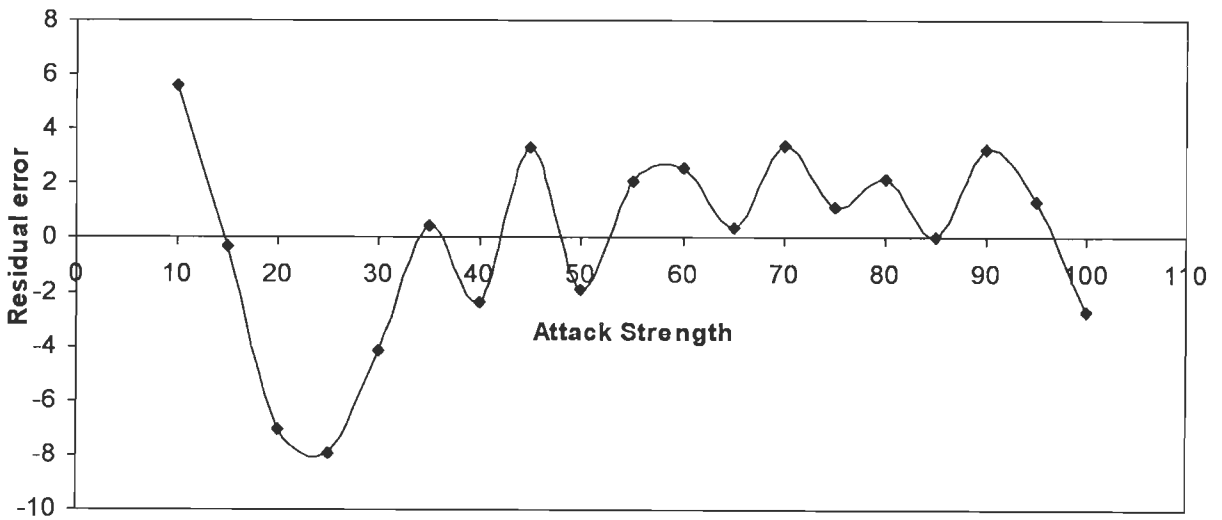


Figure 7.17. Residual error in multiple regression model

Table 7.6. Values of various performance measures for multiple regression model

| | |
|--------|--------|
| R^2 | 0.97 |
| CC | 0.98 |
| SSE | 341.02 |
| MSE | 17.94 |
| NMSE | 0.63 |
| η | 0.97 |

C. Comparison between Polynomial and multiple regression

Here performance between polynomial and multiple regression model is compared to estimate strength of attack. Polynomial regression is compared with multiple regression as it gives the best performance among the simple regression models discussed in section 7.2. It can be seen from table 7.7 that multiple regression model has higher value of coefficient of determination, coefficient of correlation and η and lower values of SSE, MSE and NMSE.

Thus, it can be concluded that it performs better than other models. It can also be verified from figure 7.18 that the difference between the actual strength and predicted strength of attack in multiple regression model is lower compared to that of the polynomial regression based model. Hence, we can conclude that, strength of attack estimated by multiple regression model is closest to the actual strength of attack.

Table 7.7. Summary of various performance measures for polynomial and multiple regression model

| | Polynomial | Multiple |
|--------|------------|----------|
| R^2 | 0.96 | 0.97 |
| CC | 0.98 | 0.98 |
| SSE | 566.31 | 341.02 |
| MSE | 29.81 | 17.94 |
| NMSE | 1.06 | 0.63 |
| η | 0.96 | 0.97 |

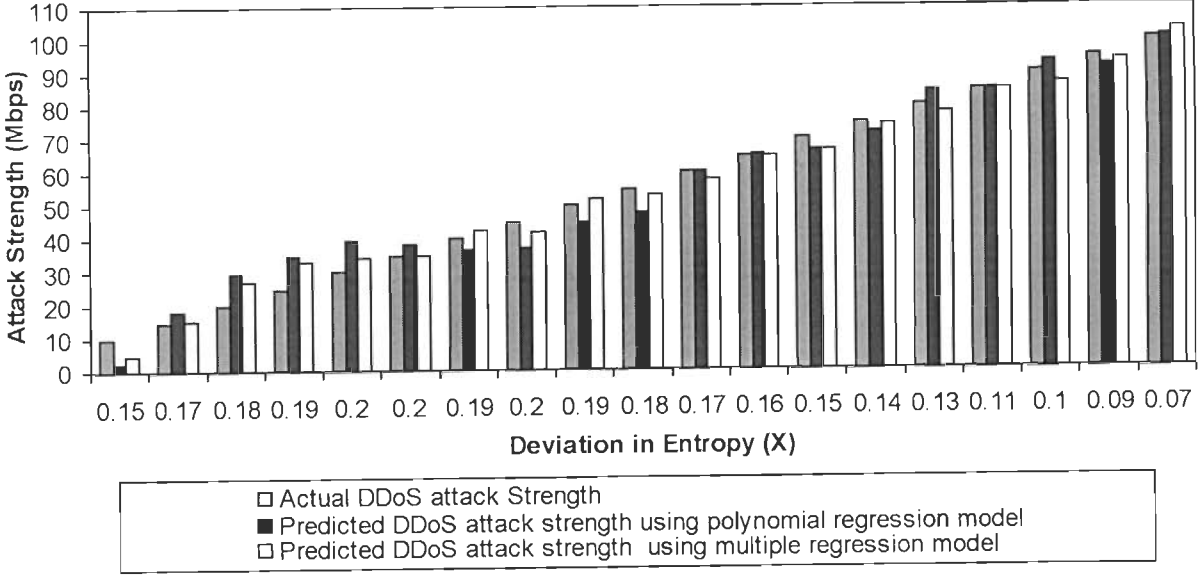


Figure 7.18. Comparison between actual strength of a DDoS attack and predicted strength of a DDoS attack using polynomial and multiple regression model

7.7 Chapter Summary

This chapter investigated suitability of various regression models to estimate strength of DDoS attack from deviation in sample entropy when simple regression models are used and deviation in volume and flow when multiple regression model is used, respectively. In order to estimate strength of DDoS attack, several models are developed using various regression techniques. For each regression model, we have calculated several statistical performance measures. Based on the statistical measures, it was found that multiple regression based model perform better than any other model explored in this study. Predicted strength of DDoS attack using multiple regression is close to actual strength of DDoS attack. It has also been observed that there is a relation between number of zombies and strength of attack. When number of zombie increases, strength of attack also increases.

CHAPTER 8

CONCLUSION AND FUTURE WORK

DDoS attacks are an impending threat to Internet related applications. This chapter summarizes the major contributions of the research described in this thesis and discusses further directions for research. Section 8.1 highlights the key research contributions and section 8.2 discusses avenues for future research.

8.1 Contributions of the Thesis

As the survey shows, DDoS attacks are currently one of the major threats in computer network security. Even if some tools like IDSs can help to defend systems against some DDoS attacks, there is no effective solution yet. In this research, we developed the various efficient approaches to defend against variety of DDoS attacks. The major contributions of our work can be summarized as follows:

- We presented a comprehensive study of a wide range of DDoS attacks and defense methods proposed to combat them and analyzed strengths and weaknesses of existing defense mechanisms.
- We developed and evaluated a real time defense mechanism for variety of flooding DDoS attacks i.e. low rate degrading, high rate disruptive, mixed rate, etc, that detects attacks by the constant monitoring of abrupt traffic changes inside ISP network. For this, a newly designed flow-volume based approach (FVBA) is used to construct profile of the traffic normally seen in the network and identify anomalies whenever traffic goes out of profile. Consideration of varying tolerance factors make proposed detection system scalable to the changeable network conditions and attack loads in real time. Proposed scheme is evaluated through extensive simulations using NS-2 network simulator on Linux platform. Performance of proposed scheme is compared with existing volume based approaches. The results show that proposed scheme gives 10-30% improvement in detection rate over earlier volume based schemes. For validating

performance of proposed scheme, KDD 99, a publicly available benchmark dataset is used.

- We analyzed DDoS attack using time series analysis and found important results. In our analysis, we made use of one of the most popular statistical nonlinear time series modeling technique, Generalized Autoregressive Conditional Heteroskedastic (GARCH) model, for detecting flooding DDoS attacks. Simulation data is generated using NS2 network simulator and MATLAB routines are used for heteroskedasticity tests and implementation of the GARCH model for detection. Proposed scheme detects flooding DDoS attacks, even if they exist over a very short time interval. Our studies show that this non linear volatility model gives 4 to 5.5% improvement in detection performance from earlier models like linear prediction. Detection performance of GARCH model based detection scheme is also compared with FVBA scheme. Results show that GARCH model based detection scheme shows marginal improvement in detection rate over FVBA.
- We used various regression models i.e. linear, polynomial, exponential, power, logarithmic, multiple etc to find relationship between number of zombies involved in a DDoS attack and deviation in traffic from detection threshold. Various statistical performance measures are used to evaluate the performance of these regression models. Network topologies similar to Internet are used for simulation and are generated using Transit-Stub model of GT-ITM topology generator. NS-2 network simulator on Linux platform is used for launching DDoS attacks with varied number of zombies. A comparative study of different regression models for predicting number of zombies is performed. The simulation results show that multiple regression model performs best.
- We employed ANN based scheme to predict number of zombies involved in a DDoS attack. The sample data used to train the feed forward neural networks is generated using NS-2 network simulator running on Linux platform. Mean square error (MSE) is used to compare the performance of various feed forward neural networks. For the prediction of the number of zombies in a DDOS attack, feed forward neural networks of different sizes have been tested. Various sizes of feed forward networks are compared for their estimation performance. The simulation results show that feed forward neural networks with 10 neurons in hidden layer performs best and is able to predict number of zombies in a DDoS attack efficiently. Prediction performance of ANN based scheme is

compared with regression based scheme. It can be concluded that the selection of ANN and regression based scheme is based on the severity of attack. When attack is more severe, ANN based scheme performed better than regression based scheme and if attack is not much severe, regression based scheme performed better than ANN based scheme.

- Regression analysis is used to investigate suitability of various regression models i.e. linear, polynomial, exponential, power, logarithmic and multiple to estimate strength of a DDoS attack. NS-2 network simulator on Linux platform is used for launching DDoS attacks with varied strength of attacks. A comparative study is performed using different regression models for estimating strength of DDoS attack. The simulation results show that multiple regression model performs best to estimate strength of a DDoS attack.

8.2 Scope for Future Work

Though our research provides efficient solutions for defending against DDoS attacks, but at the same time there are number of research issues which spring up from our work which need to be addressed. Some of them are as follows:

- The long-term challenge for defense against DDoS attacks is to find a technical and economic model to achieve cooperation between ISPs, in order to combat a wide range of DDoS attacks collaboratively.
- A large number of simulation based schemes are proposed in the literature but an effective analytical solution to defend against DDoS attacks is still a pending issue.
- Better hashing and flow classification techniques should be designed to reduce packet handling overheads, thus enabling DDoS defense to handle higher packet rates in a better manner.
- The mammoth volume generated by DDoS attacks pose the biggest challenge in terms of memory and computational overheads as far as monitoring and analysis of traffic at single point connecting the victim is concerned. To address this problem, an effective distributed cooperative technique is required to be proposed that distributes memory and computational overheads to many points e.g. all edge routers for detecting a wide range of DDoS attacks at early stage.
- Currently, proposed defense scheme is limited to single ISP domain, but it can be extended to multiple ISP domains with the help of trusted entities acting as interfaces

between two ISPs so that two ISPs can share their information and thus more effectively stop the attack.

- Simulation experiments in NS-2 testbed have been used for validation, as a proof of concept and for evaluation of the proposed schemes, but deployment and investigations using real time testbeds or real time attack traces will be more useful.
- In the present work, KDD 99 attack data has been used. Moreover, more recent attack data as and when available can be used in future.

Appendix-A

KDD 99 Dataset Description

MIT Lincoln Lab's DARPA intrusion detection evaluation datasets have been employed to design and test intrusion detection systems [208]. In 1999, Stolfo et. al. [176] summarized recorded network traffic from the DARPA 98 Lincoln Lab dataset into network connections with 41-features per connection [142, 176]. This formed the KDD 99 intrusion detection benchmark dataset that is most popular dataset used to test and evaluate a large number of IDSs. KDD dataset covers following four major categories of attacks:

- Denial of Service (DoS) attacks (deny legitimate requests to a system), e.g. ping-of-death, SYN flood
- Probing attacks (information gathering attacks), e.g. Port scanning
- Remote-to-Local (R2L) attacks (unauthorized local access from a remote machine), e.g. guessing password
- User-to-Root (U2R) attacks (unauthorized access to local super-user or root), e.g. various buffer overflow attacks

KDD dataset is divided into labeled and unlabeled records. Each labeled record consisted of 41 features and one target value. KDD dataset contains several data files, from which two files are chosen: `kddcup.data_10_percent.gz` and `corrected.gz`. In `kddcup.data_10_percent.gz`, there are around 5 million (494021) records and it was used for training and validating DDoS detection system. In `corrected.gz`, there are around 3 million (311029) records and it was used for testing DDoS detection system.

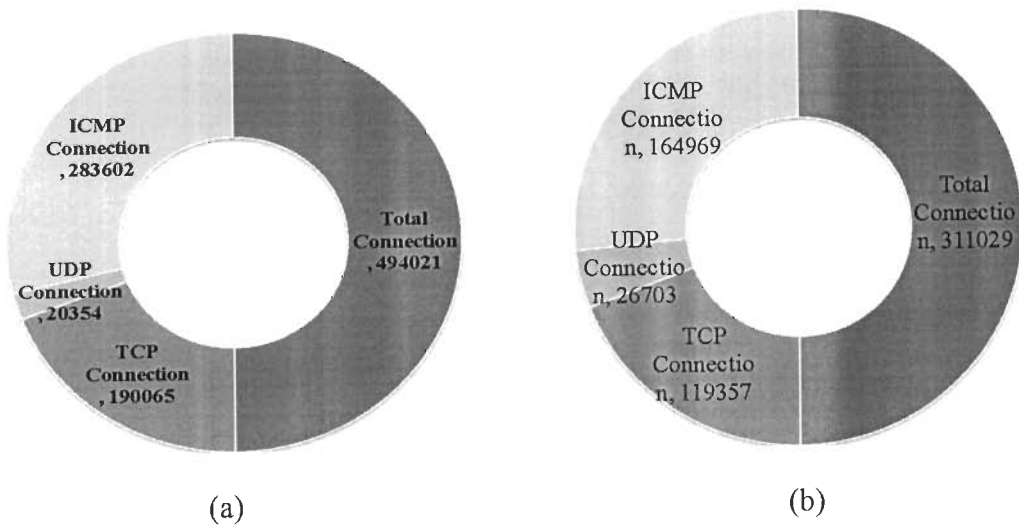


Figure A.1 Distribution of TCP, UDP and ICMP connections in (a) training dataset, (b) testing dataset

Table A.1. Distribution of TCP, UDP and ICMP connections in normal connections in (a) training dataset, (b) testing dataset

| (a) | | (b) | |
|--------------------------|-----------|--------------------------|-----------|
| Connections | Instances | Connections | Instances |
| Total Normal Connections | 97278 | Total Normal Connections | 60593 |
| TCP Normal Connections | 76813 | TCP Normal Connections | 44118 |
| UDP Normal Connections | 19177 | UDP Normal Connections | 16097 |
| ICMP Normal Connections | 1288 | ICMP Normal Connections | 378 |

Table A.2. Distribution of TCP, UDP and ICMP connections in attack connections in (a) training dataset, (b) testing dataset

| (a) | | (b) | |
|--------------------------|-----------|--------------------------|-----------|
| Connections | Instances | Connections | Instances |
| Total Attack Connections | 396743 | Total Attack Connections | 250436 |
| TCP Attack Connections | 113252 | TCP Attack Connections | 75239 |
| UDP Attack Connections | 1177 | UDP Attack Connections | 10606 |
| ICMP Attack Connections | 282314 | ICMP Attack Connections | 164591 |

Table A.3 Distribution of TCP, UDP and ICMP connections in DoS attack connections in (a) training dataset, (b) testing dataset

| (a) | | (b) | |
|------------------------------|-----------|------------------------------|-----------|
| Connections | Instances | Connections | Instances |
| Total DoS Attack Connections | 391458 | Total DoS Attack Connections | 229853 |
| TCP DoS Attack Connections | 109425 | TCP DoS Attack Connections | 65661 |
| UDP DoS Attack Connections | 979 | UDP DoS Attack Connections | 14 |
| ICMP DoS Attack Connections | 281054 | ICMP DoS Attack Connections | 164178 |

Three types of connections are there in KDD dataset: TCP connections, UDP connections and ICMP connections. Distribution of these connections in both training and testing datasets is shown in figure A.1. Table A.1, table A.2 and table A.3 show distribution of normal, attack and DoS attack connections, respectively, in training and testing datasets.

Distribution of DoS and other types of attack connections are shown in figure A.2. Total 22 and 37 attack types are there in training and testing datasets respectively. Table A.4 list the DoS attack types, protocol categories and instances in both training and testing datasets.

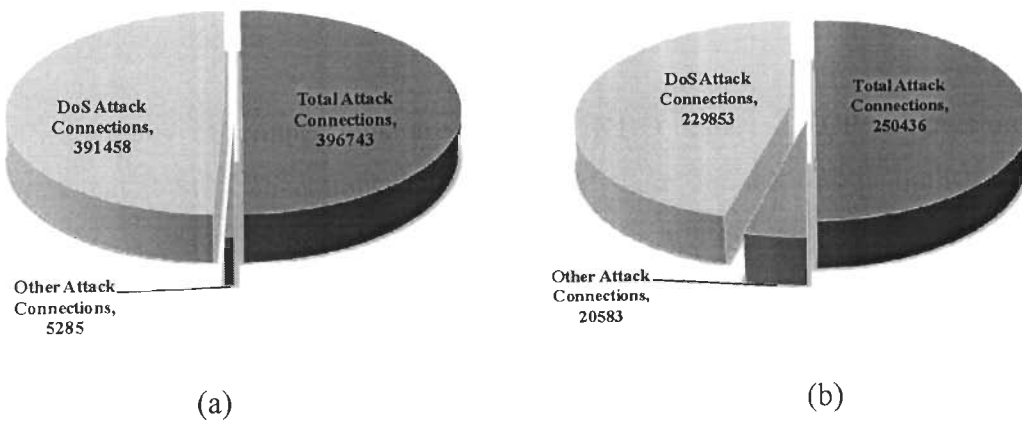


Figure A.2 Distribution of total DoS and other attack connections in (a) training dataset, (b) testing dataset

We can see that there are 6 and 10 different types of DoS attacks in training and testing dataset respectively. After analyzing Tables A.1 to A.4 and Figures A.1 to A.2, we infer the following results:

- Most of the Internet traffic is based on TCP.
- Among all attack types, DoS attack is most serious one as maximum part of attack traffic is of DoS type.

Each record in dataset has 41 extracted features, in which 38 features are continuous and others are symbolic. There are four categories of derived features, which are 9 intrinsic features, 13 content features, 9 traffic features and 10 host features.

Table A.4 Attacks Distribution in (a) training dataset, (b) testing dataset

| (a) | | | | (b) | | | |
|------------|--------|-------------------|-----------|--------------|--------|-------------------|-----------|
| DDoS Types | Attack | Protocol Category | Instances | DDoS Types | Attack | Protocol Category | Instances |
| back | | TCP | 2203 | apache2 | | TCP | 794 |
| land | | TCP | 21 | back | | TCP | 1098 |
| neptune | | TCP | 107201 | land | | TCP | 9 |
| pod | | ICMP | 264 | mailbomb | | TCP | 5000 |
| smurf | | ICMP | 280790 | neptune | | TCP | 58001 |
| teardrop | | UDP | 979 | pod | | ICMP | 87 |
| | | | | processtable | | TCP | 759 |
| | | | | smurf | | ICMP | 164091 |
| | | | | teardrop | | UDP | 12 |
| | | | | udpstorm | | UDP | 2 |

References

- [1] “Akaike Information Criteria.” <http://modelselection.org/aic/>.
- [2] “Bayesian Information Criteria.” <http://modelselection.org/bic/>.
- [3] “DDoS attacks on Yahoo, Buy.com, eBay, Amazon, E*Trade,” CNN Headline News, Feb. 7–11, 2000.
- [4] “GARCH Toolbox.” <http://math.bu.edu/misc/DOCSEVER/raw/garch.pdf>.
- [5] “Principle of Parsimony.” <http://www.philosophyprofessot.com/philosophies/parsimony-principle.php>.
- [6] A. Adas, “Traffic Models in broadband networks,” *Communications Magazine, IEEE*, vol. 35, no. 7, pp. 82-89, 1997.
- [7] A. B. Kulkarni, S. F. Bush, and S. C. Evans, “Detecting Distributed Denial-of-Service Attacks Using Kolmogorov Complexity Metrics,” Tech. Report, GE Research & Development Center, 2001CRD176, December 2001.
- [8] A. Belenky, and N. Ansari, “IP Traceback with deterministic packet marking,” *IEEE communication letter*, 7(4), pp. 162-164, 2003.
- [9] A. C. Snoeren et al., “Single Packet IP Traceback,” *IEEE/ACM Trans. Net.*, vol. 10, Dec. 2002, pp. 721–34.
- [10] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer, “Hash-Based IP Traceback,” in *Proceedings of ACM SIGCOMM 2001*, San Diego, CA, USA, pp. 3–14 August 2001.
- [11] A. D. Keromytis, V. Misra, and D. Rubenstein, “SOS: Secure Overlay Services,” in the *Proceedings of ACM SIGCOMM*, pp. 61-72, 2002.
- [12] A. Demers, S. Keshav, and S. Shenker, “Analysis and Simulation of a Fair Queuing Algorithm,” *Journal of Internetworking Research and Experience*, Volume 1, Issue 1, pp. 3-26, 1990.
- [13] A. Juels and J. Branard, “Client puzzles: A cryptographic countermeasure against connection depletion attacks,” in *Proc. NDSS’99*, pp. 151-165, 1999.
- [14] A. Lakhina, M. Crovella, and C. Diot, “Mining Anomalies Using Traffic Feature Distributions,” *ACM SIGCOMM Computer Communication Review*, Volume 35, Issue 4, 217-228, 2005.

- [15] A. Mankin and K. Ramakrishnan, "Gateway Congestion Control Survey," IETF RFC 1254, Aug. 1991. Available at: <http://www.rfc-editor.org/rfc.html>.
- [16] A. S. Sairam, G. Barua, "Effective Bandwidth Utilization in Multi homing Networks," In First international conference on communication system Software and Middleware (COMSWARE), pp. 1-8, Jan. 2006.
- [17] A. Sang and S. Li, "A Predictability Analysis of Network Traffic," Computer Networks, pp. 329-345, 2002.
- [18] Abdel El-Shaarawi, Román Viveros-Aguilera, "Logarithmic Regression," Encyclopedia of Environmetrics, DOI: 10.1002/9780470057339.val015, 2006.
- [19] Abennett, "CERT hit by DDoS attack for a third day," May 2001. Available at <http://www.itworld.com/IDG010524CERT2>.
- [20] Ajith Abraham, Crina Grosan and Carlos Martin-Vide, "Evolutionary Design of Intrusion Detection Programs," International Journal of Network Security, Vol.4, No.3, pp. 328-339, 2007.
- [21] Akamai, "Press release: Akamai provides insight into Internet denial of service attack," June 2004. Available at: <http://www.akamai.com/en/html/about/press/press459.html>.
- [22] Anandhi A, Srinivas VV, Kumar DN, Nanjundiah RS., "Role of predictors in downscaling surface temperature to river basin in India for IPCC SRES scenarios using support vector machine," International Journal of Climatology , 29: 583–603, 2009.
- [23] Anjali sardana, "Honeypot Framework for Networks under Distributed Denial of Service Attacks," PhD thesis, IIT Roorkee, 2009.
- [24] Ao Tao, Jantao Wang, "Understanding CHOKe: Throughput and Spatial Characteristics," IEEE/ACM Transactions on Networking, Vol. 12(4) (2004) 694–709.
- [25] Arbor Networks, "The Peak flow Platform". Available at: <http://www.arbornetworks.com>.
- [26] AusCERT, "2005 Australian computer crime and security survey," Tech. Report, Australian Computer Emergency Response Team, 2005. Available at: <http://www.auscert.org.au/crimesurvey>.

- [27] B. Bencsath, I. Vajda, "Protection against DDoS Attacks Based on Traffic Level Measurements," in Proceedings of the Western Simulation Multi Conference. San Diego, California, pp. 22-28, 2004.
- [28] B. Hancock, "Trinity v3, a DDoS tool," hits the streets, *Computers Security* 19(7), pp. 574, 2000.
- [29] B. M. Leiner, V. G. Cerf, et. al., "A Brief History of the Internet. Internet Society," 2003. Available at: <http://www.isoc.org>.
- [30] B. Wang, H. Schulzrinne, "Analysis of Denial-of-Service Attacks on Denial-of-Service Defensive Measures," *GLOBECOM* 2003, pp. 1339-43.
- [31] B. Yegnanarayana, "Artificial Neural Networks," Prentice-Hall, New Delhi, 1999.
- [32] B. Zhou, D. He, and Z. Sun, "Traffic Predictability based on ARIMA/GARCH model," in next generation Internet Design and Engineering, 2006. *NGI'06*, pp. 207-215, 2006.
- [33] Bai, Y. Kobayashi, H., "Intrusion Detection System: Technology and Development," in the Proceedings of the 17th International Conference on Advanced Information Networking and Applications (AINA), pp. 710-715, march, 2003.
- [34] Baker, F., "Requirements for IP version 4 routers," RFC 1812, Internet Engineering Task Force (IETF). Go online to www.ietf.org.
- [35] Bo Zhao, Caixia Chi, Wei Gao, Sencun Zhu, Guohong Cao, "A Chain Reaction DoS Attack on 3G Networks: Analysis and Defenses," *INFOCOM*, pp. 2455-2463, 2009.
- [36] Bysin, "Knight.c sourcecode," [PacketStormSecurity.nl](http://packetstormsecurity.nl), July 11, 2001. Available at: <http://packetstormsecurity.nl/distributed/knight.c>.
- [37] C. Douligeris, A. Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art," *Computer Networks*, Volume 44, Issue 5, pp. 643-666, April 2004.
- [38] C. Douligeris, A. Mitrokotsa, "DDoS attacks and defense mechanisms: classification," in Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (ISSPIT 03), Darmstadt, Germany, pp. 190-193, Dec. 14-17, 2003.
- [39] C. Gonsalves, "Akamai DDoS attack whacks Web traffic," June 2004. Available at: <http://www.eweek.com/article2/0,1895,1612739,00.asp>.

- [40] C. Jin, H. Wang, and K. G. Shin, "Hop-count Filtering: An Effective Defense Against Spoofed DDoS Traffic," Proceedings of the 10th ACM Conference on Computer and Communication Security, (CCS 2003), Washington D.C., USA, pp.30-41, October 27-31, 2003;
- [41] C. M. Cheng, H. T. Kung, K. S. Tan, "Use of spectral analysis in defense against DoS attacks," in Proceedings of IEEE GLOBECOM 2002, Taipei, Taiwan, pp. 2143-2148, 2002.
- [42] C. Papadopoulos, R. Lindell, J. Mehringer, A. Hussain, R. Govindan, "COSSACK: Coordinated Suppression of Simultaneous Attacks," in Proceedings of the DARPA Information Survivability Conference and Exposition, Vol. 1, pp. 2-13, April 2003.
- [43] C. Sangpachatanaruk, S. M. Khattab, T. Znati, r. Melhem, and Mosse, " Design and analysis of a replicated elusive server scheme for mitigating denial of service attack," the journal of System and Software, Vol. 73, pp. 15-29, 2004.
- [44] C. Schuba, I. Krsul, M. Kuhn, G. Spafford, A. Sundaram, and D. Zamboni, "Analysis of a denial of service attack on TCP," in Proceedings of the 1997 IEEE Symposium on Security and Privacy, May 1997.
- [45] CERT Coordination Center, "Denial of Service Tools". Available at: <http://www.cert.org/advisories/CA-1999-17.html>.
- [46] CERT Coordination Center, Carnegie Mellon Software Engineering Institute, CERT Advisory CA-2001-20, "Continuing threats to home users," 23, 2001. Available at: <http://www.cert.org/advisories/CA-2001-20.html>.
- [47] CERT Coordination Center," Mail bomb attack". Available at: http://www.cert.org/tech_tips/email_bombing_spamming.html.
- [48] CERT Coordination Center. "Trends in Denial of Service Attack Technology, October 2001". Available at: http://www.cert.org/archive/pdf/DoS_trends.pdf.
- [49] CERT statistics. Available at: http://www.cert.org/stats/cert_stats.html.
- [50] CERT, "CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks," September 1996.
- [51] CGI request attack. Available at: <http://cpan.uwinnipeg.ca/htdocs/CGI.pm/CGI.html>.
- [52] Chandan Singh Negi, "Using Network Management System to detect Distributed Denial of Service Attacks," Master's thesis, Naval Postgraduate School Monterey, CA, September 2001.

- [53] Craig A Huegen, "The Latest in Denial of Service Attacks: 'SMURFING' Description and Information to Minimize Effects," 2000. Available at: <http://www.pentics.net/denial-of-service/white-papers/smurf.txt>.
- [54] Craig Morris, "40,000 euros offered for identities of online blackmailers," www.heise.de/english/newsticker/news/63238, posted on 25 August, 2005.
- [55] Cs3. Inc. "MANAnet DDoS", White Papers. Available at: <http://www.cs3-inc.com/mananet.html>.
- [56] D. Dean, M. Franklin and A. Stubblefield, "An algebraic approach to IP traceback," *ACM Trans. Inform. System Security*, Volume 5, Issue 2, pp. 119–137, 2002.
- [57] D. Dittrich, "The DoS Project's Trinoo Distributed Denial of Service attack tool," University of Washington, October 21, 1999. Available at: <http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt>.
- [58] D. Dittrich, "The Stacheldraht Distributed Denial of Service attack tool," University of Washington, December 1999. Available at: <http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt>.
- [59] D. Dittrich, "The Tribe Flood Network Distributed Denial of Service attack tool," University of Washington, October 21, 1999. Available at: <http://staff.washington.edu/dittrich/misc/tfn.analysis.txt>.
- [60] D. Dittrich, G. Weaver, S. Dietrich, and N. Long, "The "Mstream" distributed denial of service attack tool," May 2000. Available at: <http://staff.washington.edu/dittrich/misc/mstream.analysis.txt>.
- [61] D. Foster and E. George, "The risk inflation criterion for multiple regression. *Annals of Statistics*," 22, 1994 pp. 1947-1975.
- [62] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient Overlay Networks," In *Proceedings of 18th ACM SOSP*, October 2001.
- [63] D. Lin, and R. Morris, "Dynamics of Random Early Detection," In *Proceeding of ACM SIGCOMM*, pp.127-137, New York, 1997.
- [64] D. Mankins, R. Krishnan, c. Boyd, J. Zao, M. Frenzt, and B. B. N. Technologies, "mitigating Distributed Denial of Service Attacks with Dynamic Resource Pricing," in *Proceedings of 17th annual Computer Security applications conference*, pp. 411-421, 2001.

- [65] D. Moore, C. Shannon, D. J. Brown, G. Voelker, S. Savage, "Inferring Internet Denial-of-Service Activity," *ACM Transactions on Computer Systems*, 24 (2), 115-139, (2006).
- [66] D. Moore, "The spread of the code red worm (crv2)". Available at: http://www.caida.org/analysis/security/code-red/coderedv2_analysis.xml.
- [67] D. X. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," in *IEEE INFOCOM 2001*.
- [68] D. Yau, J. Lui, F. Liang, and Y. Yam, "Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles," *IEEE/ACM Transaction on Networking*, vol. 13, 29-42, 2005.
- [69] David A. Freedman, "Statistical Models: Theory and Practice," Cambridge University Press, 2005.
- [70] Debar H, Dacier M, Wespi A, "Towards a taxonomy of intrusion detection systems," *Computer Networks*, Vol. 31, 1999.
- [71] Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining, "Introduction to Linear Regression Analysis (3rd ed.)," New York: Wiley, 2001, xv + 641 pp., ISBN: 0-471-31565-6.
- [72] Elinor Mills, "Radio Free Europe DDOS attack latest by activists," Online at http://news.cnet.com/8301-10784_3-9933746-7.html, CNET News, May. 2008.
- [73] F. Kargl, J. Maier, and M. Weber, "Protecting web servers from Distributed Denial of Service Attacks," in *Proceedings of the Tenth International conference on World wide Web*, pp. 514-524, Hong Kong, 2001.
- [74] F. SCALZO, "Recent DNS Reflector Attacks," VeriSign, 2006. Available at: <http://www.nanog.org/mtg-0606/pdf/frank-scalzo.pdf>.
- [75] Felix C. Freiling, Thorsten Holz, and Georg Wicherski, "Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks," *Laboratory for Dependable Distributed Systems, RWTH Aachen University*, Springer-Verlag Berlin Heidelberg 2005.
- [76] Felix Lau, Rubin H. Stuart, Smith H. Michael, and et al., "Distributed Denial of Service Attacks," in *Proceedings of 2000 IEEE International Conference on Systems, Man, and Cybernetics*, Nashville, TN, Vol.3, pp.2275-2280, 2000.

- [77] Fonseca, B., "Yahoo outage raises Web concerns," 2000. Available at: <http://www.nwfusion.com/news/2000/0209yahoo2.html>.
- [78] G Sivakumar, "Cryptographic Protocols and Network Security," 2004. Available at: www.cse.iitb.ac.in/~siva/talks/crypto.pdf
- [79] G. Carl, G. Kesidis, R. R. Brooks, and S. Rai, "Denial-of-service attack-detection techniques," *IEEE Internet computing*, vol. 10, pp. 82-89, 2006.
- [80] Gang Wang, Jinxing Hao, Jian Ma, Lihua Huang, "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering," *Expert Systems with Applications*, vol. 37, 2010, pp. 6225–6232.
- [81] GL Baskerville, "Use of logarithmic regression in the estimation of plant biomass," *Canadian Journal of Forest Research*, NRC Research Press, 1972.
- [82] Global Incident analysis Center-Special Notice-"Egress filtering v0.2". Available at: <http://www.sans.org/y2k/egress.htm>.
- [83] Gwyn Aneuryn-Evans and Angus Deaton, "Testing Linear versus Logarithmic Regression Models," *The Review of Economic Studies*, vol. 47, no. 1, Econometrics Issue, 1980, pp. 275-291.
- [84] H Dette, IM Lopez, IMO Rodriguez, A Pepelyshev, "Maximum efficient design of experiment for exponential regression models," *Journal of Statistical Planning and Inference*, vol. 136, no. 12, pp. 4397–4418, 2006.
- [85] Hal Burch and Bill Cheswick, "Tracing Anonymous Packets to Their Approximate Source," in *Proceedings of the 14th Systems Administration Conference*, New Orleans, Louisiana, U.S.A., December 2000.
- [86] Hill, T., Lewicki, P, "Statistics Methods and Applications," StatSoft, Tulsa, OK, 2007.
- [87] <http://d.root-servers.org/october21.txt>
- [88] <http://lists.jammed.com/ISN/2009/03/0039.html>
- [89] <http://staff.washington.edu/dittrich/misc/ddos/register.com-unisog.txt>
- [90] <http://www.bksmf.com/index.php?topic=119.0%3Bwap2>
- [91] http://www.circleid.com/posts/attack_internet_root_servers/
- [92] <http://www.thesmarttechie.com/magazine/fullstory.php/WJJE491776651>
- [93] I. Stoica, S. Shenker, and H. Zhang, "Core-stateless fair Queuing: Achieving approximately Fair bandwidth allocations in high speed Networks," in *Proceedings of ACM SIGCOMM*, New York, 1998.

- [94] Iftikhar Ahmad, Azween B Abdullah, Abdullah S Alghamdi, "Application of Artificial Neural Network in Detection of DOS Attacks," In proceedings of International Conference on Security of Information and Networks (SIN 2009), October 6–10, 2009, North Cyprus, Turkey. pp. 229-234.
- [95] Internet World Stats, Internet User Statistics–The Big Picture: World Internet Users and Population Stats, <http://www.internetworldstats.com/stats.htm>
- [96] Internet: www.cisco.com/debugging.htm.
- [97] J. Howard, "An analysis of Security Incidents on the Internet 1989-1995," PhD thesis, Carnegie Mellon University, Aug, 1998.
- [98] J. Joannidis, S. Bellovin, "Implementing pushback: Router-based defense against DDoS attacks," in Proc. Network and distributed System Security Symposium, pp. 1-12, 2002.
- [99] J. Barlow, W. Thrower, "TFN2K- An Analysis," Axent Security Team. February 10, 2000. Available at: http://security.royans.net/info/posts/bugtraq_ddos2.shtml.
- [100] J. Evers, "Bot Herders May Have Controlled 1.5 Million PCs," ZDNet News, Oct. 21, 2005, http://news.zdnet.com/2100-1009_22-5906896.html.
- [101] J. Jung, b. Krishnamurthy, M. Rabinovich, "Flash Crowds and Denial of Service Attacks: Characterization and Applications for CDNS and Websites," In Proceedings of International World Web Wide conference, ACM Press, pp. 252-262, 2002.
- [102] J. K. Kishore, L. M. Patnaik, v. Mani and V. K. Agrawal, "Application of generic programming for multicategory pattern classification," IEEE Transactions on Evolutionary Computation, vol. 4. pp. 242-258, 2000.
- [103] J. Leyden, "Scottish ISP floored as DDoS attacks escalates," Apr. 2002. Available at: http://www.theregister.co.uk/2002/04/09/scottish_isp_floored_as_ddos/.
- [104] J. Li, J. Mirkovic, M. Wang, P. Reiher and L. Zhang, "SAVE: Source Address Validity Enforcement Protocol," in Proceedings of IEEE INFOCOM, pp. 1557-1566, 2002.
- [105] J. Makhoul, "Linear Prediction: A Tutorial Review," in proc. Of the IEEE, vol. 63(4), pp. 561-580, 1975.
- [106] J. Mirkovic, G. Prier, and P. Reiher. "Attacking DDoS at the Source," in Proceedings of the ICNP 2002, November 2002.

- [107] J. Mirkovic, M. Robinson, and P. Reiher, "Forming Alliance for DDoS Defenses," in Proceedings of New Security Paradigms Workshop (NSPW 2003), ACM Press, New York, NY, 11-18, Aug. 2003.
- [108] J. Mirkovic, P. Reiher, "A Taxonomy of DDoS Attack and DDoS defense Mechanisms," ACM SIGCOMM Computer Communications Review, Volume 34, Issue 2, pp. 39-53, April 2004.
- [109] J. Molsa, "Mitigating denial of service attacks: A tutorial," Journal of computer security, 13, pp. 807-837, IOS Press, 2005.
- [110] J. Yan, S. Early, and R. Anderson. "The XenoService – A Distributed Defeat for Distributed Denial of Service," in Proceedings of ISW 2000, October 2000.
- [111] J-063: Domain Name System (DNS) Denial of Service (DoS) Attacks, 1999. Available at: <http://www.securityfocus.com/advisories/1727>.
- [112] James R. Eagan, John Stasko and Ellen Zegura, "Interacting with Transit-Stub Network Visualizations," InfoVis, 2003. Available: <http://www.cc.gatech.edu/gvu/ii/netviz/eaganIV2k3poster.pdf>
- [113] Jane Wakefield, "Online service foils ransom plot", <http://news.bbc.co.uk/1/hi/technology/4579623.stm>, posted on 31 May, 2005.
- [114] Javvin network management & security, "UDP Flood attack". Available at: <http://www.javvin.com/networkSecurity/UDPFloodAttack.html>.
- [115] Jianqing Fan, Qiwei Yao, "Nonlinear Time Series: Nonparametric and Parametric Methods," Springer-Verlag New York, 2003.
- [116] Joao B. D. Cabrera, Lundy Lewis, Xinzhou Qin, Wenke Lee, Ravi K. Prasanth, B. Ravichandran and Raman K. Mehra, "Proactive Detection of Distributed Denial of Service Attacks using MIB Traffic Variables -A Feasibility Study", in Proceedings of the 7th IFIP/IEEE International Symposium on Integrated Network Management, Seattle, WA - May 14-18, 2001.
- [117] K. Hwang, M. Cai, Y. Chen, M. Qin, "Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes," IEEE Transaction on Dependable and Secure Computing, Volume 4, Issue 1, 41-55, 2007.
- [118] K. Kumar, "Protection from Distributed Denial of Service (DDoS) Attacks in ISP Domain," PhD thesis, IIT Roorkee, 2006.

- [119] K. Kumar, K. Singh, R. C. Joshi, "Distributed Approach to Detect DDoS attacks in ISP Domain using Entropy," in the proceedings of International Conference on Advanced Communication System (ICACS-2007), India, pp. 101-108, 2007.
- [120] K. Kumar, R. C. Joshi and K. Singh, "An Integrated Approach for Defending against Distributed Denial-of-Service (DDoS) Attacks," IRISS, 2006, IIT Madras. Available at: www.cs.iitm.ernet.in/~iriss06/iitr_krishan.pdf.
- [121] K. Kumar, R. C. Joshi, and K. Singh, "Predicting Number of Attackers using Regression Analysis," In Proceedings of IEEE International conference on Information and Communication Technology, pp. 319-322, Dhaka, Bangladesh, March 2007.
- [122] K. Park and H. Lee, "On the effectiveness of the route-based packet filtering for distributed DoS attack prevention in power-law internets," ACM SIGCOMM Computer Communication Review, vol. 31, pp. 15-26, 2001.
- [123] K. Sohn and D. Kim, "Statistical Model for Forecasting Link Travel Time Variability," Journal of Transportation Engineering, vol. 135, no. 7, pp. 440-453, 2009.
- [124] Kirchgassner, Gebhard, Wolters, and Jurgen, "Introduction to Modern Time Series Analysis," Springer, 2007.
- [125] L. A. Gordon, M. P. Loeb, W. Lucyshyn, R. Richardson, "2005 CSI/FBI computer crime and security survey," Tech. Report, Computer Security Institute, 2005. Available at: www.GCSI.com.
- [126] L. Feinstein, D. Schnackenberg, R. Balupari, D. Kindred, "Statistical Approaches to DDoS Attack Detection and Response," in Proceedings of DISCEX'03, Washington, DC, USA, Vol. 1, pp. 303-314, 2003.
- [127] L. Garber, "Denial-of-service attacks rip the Internet," IEEE Computer, Volume 33, Issue 4, pp. 12-17, Apr. 2000.
- [128] L. Santhanam, A. Kumar, and D. P. Agrawal, "Taxonomy of IP Traceback," Journal of information assurance and security, vol. 1, pp. 79-94, 2006.
- [129] Leiner, B. M., Cerf, V. G., et. al. (2003), "A Brief History of the Internet," Internet Society. Available at: <http://www.isoc.org>.
- [130] Lindley, D.V. (1987). "Regression and correlation analysis," New Palgrave: A Dictionary of Economics, vol. 4, pp. 120-23.

- [131] M. Handley, "Internet Architecture WG: DoS-resistant Internet subgroup report," 2005. Available at: <http://www.communications.net/object/download/1543/doc/mjh-dos-summary.pdf>.
- [132] M. Harish, N. Anandavelu, N. Anbalagan, G. S. Mahalakshmi, T. V. Geetha, "Design and Analysis of a Game Theoretic Model for P2P Trust Management," ICDCIT 2007: 110-115.
- [133] M. K. Goyal, C.S. P. Ojha, "Evaluation of Various Linear Regression Methods for Downscaling of Mean Monthly Precipitation in Arid Pichola Watershed," Natural Resources, Vol. 1, pp. 11-18, 2010. doi:10.4236/nr.2010.11002.
- [134] M. Kenney, "Ping of Death attack". Available at: <http://insecure.org/splouts/ping-o-death.html>.
- [135] M. Li, Ming Li, X. Jiang, "DDoS Attacks Detection Model and its Applications," WSEAS Transactions on Computers, 7(8), pp. 1159-1168, 2008.
- [136] M. Marchesseau, "Trinity-Distributed Denial of Service Attack Tool," 11 Sept, 2000. Available at: http://www.giac.org/certified_professionals/practicals/gsec/0123.php.
- [137] M. Robinson, J. Mirkovic, M. Schnaider, S Michel, and P. Reiher, "Challenges and principles of DDoS defense," SIGCOMM, 2003.
- [138] M. Roesch, "Snort-Lightweight Intrusion Detection for Networks," in Proceedings of the USENIX Systems Administration Conference (LISA '99), pp. 229-238, Nov.1999.
- [139] Mazu Networks. Mazu Technical White Papers. Available at: http://www.mazunetworks.com/white_papers/.
- [140] McAfee, "Personal Firewall". Available at: http://www.mcafee.com/myapps/firewall/ov_firewall.asp.
- [141] Michael Gribskov and Nina L. Robinson, "Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching," Computers & Chemistry, Volume 20, Issue 1, March 1996, Pages 25-33.
- [142] MIT Lincoln Laboratory, "DARPA Intrusion Detection Evaluation," 1999, MA, USA. Available at: <http://www.ll.mit.edu/IST/ideval/index.html>.
- [143] N. Weaver. Warhol Worm. <http://www.cs.berkeley.edu/nweaver/worms.pdf>.
- [144] N. Weiler, "Honeypots for Distributed Denial of Service Attacks," in Proceedings of the Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure

for Collaborative Enterprises (WETICE'02), Pittsburgh, USA, pp. 109-114, June 2002.

- [145] Nash JE, Sutcliffe JV., "River flow forecasting through conceptual models," part I – a discussion of principles. *Journal of Hydrology* 10:282–290, 1970.
- [146] Nikhil Ranjan, "GARCH Models for TCP based denial of service attacks," MS Thesis, IIT Madras, 2010.
- [147] P. Ferguson, and D. Senie, "Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing," RFC 2267, the Internet Engineering Task Force (IETF), 1998.
- [148] P. Galli, "DoS attack brings down Sun Grid demo," March 2006. Available at: <http://www.eweek.com/article2/0,1895,1941574,00.asp>.
- [149] P. Mckenny, "Stochastic Fairness Queuing," in *Proceeding of IEEE INFOCOM*, Piscataway, N.J., pp. 733-740, 1990.
- [150] P. R. Hansen and a. Lunde, "A Forecast comparison of volatility Models: Does anything beat a GARCH(1,1)?," in *Journal of Applied Econometric*, vol. 20, pp. 873-889, 2005.
- [151] Paul D. Ezhilchelvan, Santosh K. Shrivastava, "A Characterization of Faults in Systems," *Symposium on Reliability in Distributed Software and Database Systems*, 1986: 215-222.
- [152] Power Regression. http://help.ixellence.com/dataplore/dp_manual182.html.
- [153] R. Azrina, R. Othman, "Understanding the Various Types of Denial of Service Attack". Available at: www.niser.org.my/resources/dos_attack.pdf.
- [154] R. B. Blazek, H. Kim, B. Rozovskii, A. Tartakovsky, "A novel approach to detection of denial-of-service attacks via adaptive sequential and batch sequential change-point detection methods," in *Proceedings of IEEE Systems, Man and Cybernetics Information Assurance Workshop*, pp. 220-226, 2001.
- [155] R. B. Lee, "Taxonomies of Distributed Denial of Service networks, attacks, tools and countermeasures," Princeton University, 2003. Available at: <http://www.ee.princeton.edu/~rblee>.
- [156] R. Bush, D. Karrenberg, M. Kusters and R. Plzak, "Root name server operational requirements," RFC2870, Internet: <http://www.ietf.org/rfc/rfc2870.txt>, 2000.

- [157] R. Farrow, "TCP SYN Flooding attacks and Remedies," Network Computing Unix World. Available at: <http://www.networkcomputing.com/unixworld/security/004/004.txt.html>.
- [158] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Aggregate-based congestion control," ACM SIGCOMM Computer Communication Review, vol. 32, p. 69, 2002.
- [159] R. Mahajan, S. Bellovin, S. Floyd, V. Paxson, and S. Shenker. "Controlling high bandwidth aggregates in the network," ACM Computer Communications Review, 32(3), July 2002.
- [160] R. Mahajan, S. Floyd, and D. Wetherall, "Controlling high-bandwidth flows at the congested router," in Proc. 9th International conference on network protocols, p. 192, 2001.
- [161] R. Oppliger, "Internet Security: firewall and beyond," Communications of the ACM, Volume 40, Issue 5, pp. 92-102, 1997.
- [162] R. Stone, "Centertrack: An IP Overlay Network for Tracking DOS Floods," In Proceedings of the 9th USENIX Security Symposium, Denver, Colorado, U.S.A., October 1999.
- [163] R. Thomas, B. Mark, T. Johnson, and J. Croall, "NetBouncer: Client-legitimacy based High-performance DDoS Filtering," DARPA Information Survivability Conference and Exposition III, Vol. 1, pp.14-25, April 22-24, 2003;
- [164] Robert Vamosi, "Study: DDoS attacks threaten ISP infrastructure," Online at http://news.cnet.com/8301-1009_3-10093699-83.html, CNET News, Nov. 2008.
- [165] Ruey S. Tsay, "Analysis of Financial Time Series," 3rd Edition, John Wiley & Sons, Inc., Publication, New Jersey, 2010.
- [166] S. Hazelhurst, "Algorithms for Analyzing Firewall and Router Access Lists", In proceedings of workshop on dependable IP systems and platforms (ICDSN), 2000.
- [167] S. Bellovin, M. Blaze, R. Canetti, J. Ioannidis, A.D. Keromytis, O. Reingold, "Efficient, DoS-resistant, secure key exchange for Internet protocols," In Proceedings of the 2001 Security Protocols International Workshop, April 2001, Cambridge, England.
- [168] S. Bellovin, M. Leech, and T. Taylor, "ICMP Traceback Messages," Internet draft: draft-ietf-itrf-01.txt, work in progress, October 2001.

- [169] S. Blake, D. Black, m. Carlson, et al, "an Architecture of Differentiated Services," IETF, RFC 2475, 1998.
- [170] S. Cheung, "Denial of Service against the Domain Name System," IEEE Security & Privacy, Volume. 4, Issue 1, pp. 40-45, Jan/Feb. 2006.
- [171] S. Dietrich, N. Long, D. Dittrich, "Analyzing Distributed Denial of Service tools: The Shaft Case," in Proceedings of the 14th Systems Administration Conference (LISA 2000), New Orleans, LA, USA, pp. 329-339, December 3-8, 2000.
- [172] S. Floyd and K. Fall, "Promoting the use of End-to-End Congestion Control in the Internet," IEEE/ACM Trans. on Networking, Volume 7, Issue 4, pp. 458-472, August 1999.
- [173] S. Floyd and V. Jacobon "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM Transaction on Networking, vol. 1, no. 4, pp. 397-413, 1993.
- [174] S. Floyd, S. Bellovin, J. Loannidis, K. Kompella, R. Mahajan, V. Paxson, "Pushback Messages for Controlling Aggregates in the Network," draft-floyd-pushback-messages-00.txt, 2001.
- [175] S. Gibson, "The Strange Tale of the Attacks Against GRC.COM," March 2002. Available at <http://grc.com/dos/grcdos.htm>.
- [176] S. J. Stolfo, W. Lee, W. Fan, A. Prodromidis, P. K. Chan, "Cost-based Modeling for Fraud and Intrusion Detection: Results from the JAM Project," DARPA Information Survivability Conference 2000.
- [177] S. K. Gupta, V. Bhatnagar, and S. K. Wasan, "On mining of Data," IETE journal of Research, special issue on Data and Knowledge Engineering, vol. 47, no. 1, pp. 5-18, 2001.
- [178] S. M. Khattab, C. Sangpachatanaruk, R. Melhem, D. Mosse, T. Znati, "Proactive server roaming for mitigating Denial of Service attacks," in Proceedings of the 1st International Conference on International Technology: Research and Education (ITRE 03), Newark, NJ, pp. 500-504, August 2003.
- [179] S. M. Stigler, "Optimal Experimental Design for Polynomial Regression," Journal of American Statistical Association, 1971, vol. 66, num. 334, pp. 311-318, 1971.
- [180] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network support for IP traceback," Proc. IEWACM Transaction on Networking vol. 9: (3), pp. 226-237, June 2001.

- [181] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical Network Support for IP Traceback," in Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden, pp. 295-306, August 2000.
- [182] S. Templeton and K. Levitt, "Detecting Spoofed Packets", DARPA Information Survivability Conference and Exposition, April 22-24, 2003;
- [183] Shravan K Rayanchu, Gautam Barua, "Tracing Attackers with Deterministic Edge Router Marking (DERM)", In proceedings of Distributed Computing and Internet Technology (ICDCIT) vol. 3347 of LNCS, pp. 400-409, 2004.
- [184] Sreenivasa Rao Jammalamadaka, "Introduction to Linear Regression Analysis," The American Statistician. February 1, 2003, 57(1): 67-67.
- [185] Stefan Seufert and Darragh O'Brien, "Machine Learning for Automatic Defense against Distributed Denial of Service Attacks," In proceedings of IEEE International Conference on Communications, 2007. ICC '07, 24-28 June 2007 Page(s):1217 – 1222.
- [186] Suhail Mohiuddin, shlomo Hershkop, Rahul Bhan, & Sal Stolfo, "Defending Against large scale Denial of Service attacks," proceedings of the 2002 IEEE workshop on information Assurance & Security, west point, NY. June 2002.
- [187] T. Anderson, T. Roscoe, D. Wetherall, "Preventing Internet Denial-of-Service with Capabilities," In ACM SIGCOMM Computer Communication Review, Volume 34, issue 1, January 2004, pp. 39-44.
- [188] T. Arura, P. Nikander, and J. Leiwo, "DOS-resistant authentication with client puzzles," In Lecture notes in Computer science, vol. 2133, B. Christianson, b. Crispo, J. A. Malcolm, and M. Roe, Eds.: Springer, pp. 170-177, 2001.
- [189] T. Bollerslev, "Generalized Autoregressive conditional Heteroskedasticity," Journal of Econometrics, vol. 31, pp. 307-327, 1986.
- [190] T. Darmohray, R. Oliver, "Hot spares for DDoS attacks," 2000. Available at: <http://www.usenix.org/publications/login/2000-7/apropos.html>.
- [191] T. J. Ott, T. V. Lakshman, and L. H. Wong, "SRED: Stabilized RED," In proceedings of IEEE INFOCOM, New York, USA, pp. 1346-1355, March, 1999.
- [192] T. M. Gil, M. Poletto, "Multops: a data-structure for bandwidth attack detection," in Proceedings of the 10th USENIX Security Symposium, Washington, DC, USA, pp. 23-38, 2001.

- [193] T. Peng, C. Leckie, and K. Ramamohanarao, "Protection from distributed denial of service attacks using history-based IP filtering," in IEEE International conference on Communications, pp. 482-486, 2003.
- [194] T. W. Anderson, "The Choice of the Degree of a Polynomial Regression as a Multiple Decision Problem," *The Annals of Mathematical Statistics*, Vol. 33, No. 1 (Mar., 1962), pp. 255-265.
- [195] Tao Peng, Christopher Leckie, Kotagiri Ramamohanarao, "Detecting Distributed Denial of Service Attacks Using Source IP Address Monitoring," In Proceedings of the Third International IFIP-TC6 Networking Conference (Networking 2004), 771-782.
- [196] TCP Reset. Available at: http://www.cisco.com/univercd/cc/td/doc/Product/voice/c_callmg/sec_vir/secup/tcpreset.htm.
- [197] Teardrop attacks. Available at: <http://www.physnet.uni-hamburg.de/physnet/security/vulnerability/teardrop.html>.
- [198] The ISC Internet Domain Survey. <https://www.isc.org/solutions/survey>.
- [199] U. K. Tupakula and V. Vardharajan, "A practical method to counteract denial of service attacks," in Proc. 26th Australasian computer science conference pp. 275-284, 2003.
- [200] Udith E. Dayhoff, James M. DeLeo, "Artificial neural networks," *Cancer*, American Cancer Society, Volume 91 Issue S8, Pages 1615 – 1635, 2001.
- [201] U. K. Tupakula, V. Varadharajan, "Analysis of Traceback Techniques", In Proc. Of 2006 Australasian Workshops on Grid computing and e-research, volume 54, pp. 115-124, 2006.
- [202] U. K. Tupakula, V. Varadharajan, "A controller agent model to counteract DoS attacks in multiple domains", in proceedings of IFIP/IEEE Eighth International Symposium on Integrated Network Management, pp. 113-116, 2003.
- [203] V. Gowariker, V. Thapliyal, S. M. Kulshrestha, MANDAL et. al., "A power regression model for long range forecast of southwest monsoon rainfall over India," *Mausam*, 1991, vol. 42, num. 2, pp. 125-130.
- [204] V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," *ACM SIGCOMM Computer Communications Review (CCR)*, Volume 31, Issue 3, pp. 38-47, 2001.

- [205] V. Paxson, "Bro: A System for Detecting Network Intruders in Real-Time," *International Journal of Computer and Telecommunication Networking*, Volume 31, Issue 24, pp. 2435-2463, 1999.
- [206] V.B. Melas, "Optimal designs for exponential regression," *Statistics*, Volume 9, Issue 1, 1978, pages 45 – 59.
- [207] W. Lee and S. J. Stolfo, "Data mining approaches for intrusion detection," 7th USENIX Security Symposium, San Antonio, TX, pp 79- 93, January 1998.
- [208] W. Lee, S. J. Stolfo, K. W. Mok, "Mining in a data-flow environment: experience in network intrusion detection," in *Proceedings of the 5th ACM SIGKDD*, San diego, CA, pp. 114-124, 1999.
- [209] W. Zhao, D. Olshefski, and H. Schulzrinne, "Internet quality of Service: an Overview," *Columbia Technical Report CUCS-003-00*, 2000.
- [210] Wang, H., Zhang, D., and Shin, K. G., "Detecting SYN flooding attacks," In *Proceedings of IEEE INFOCOM*, 2002. 1530–1539.
- [211] Williams, M., "EBay, Amazon, Buy.com hit by attacks," 2000, <http://www.nwfusion.com/news/2000/0209attack.html>.
- [212] X. f. Wang and M. K. Peiter, "Defending against denial-of-service attacks with puzzle auctions," in *proc. Symposium on Security and Privacy*, pp. 78-92, 2003.
- [213] X. Geng, A.B. Whinston, "Defeating Distributed Denial of Service attacks," *IEEE IT Professional* 2 (4) (2000) 36–42.
- [214] Y. Chen, K. Hwang, W. Ku, "Collaborative Detection of DDoS Attacks over Multiple Network Domains," *IEEE Transaction on Parallel and Distributed Systems*, TPDS-0228-0806, 18 (12), Dec. 2007.
- [215] Y. Huang, J. M Pullen, "Countering Denial of-Service attacks Using Congestion Triggered Packet Sampling and Filtering," *Proc. IO' ICCCN*, Anzona, USA, Oct. 2001.
- [216] Y. Xiang, W. Zhou, and M. Chowdhury, "A Survey of Active and Passive Defense Mechanisms against DDoS Attacks," *Technical Report, TR C04/02*, School of Information Technology, Deakin University, Australia, 2004.
- [217] Y. Xiong, S. Liu, and P. Sun, "On the defense of the distributed denial of the service attacks: An On-Off Feedback Control Approach," *IEEE Transactions on system, man and cybernetics-part A: systems and humans*, vol. 31, no. 4, pp. 282-293, 2001.

- [218] Z. Xiaobo, W. Jianbin, C. Xu, "Quality-of-service differentiation on the Internet: A Taxonomy," *Journal of Network and computer Applications*, Vol. 30, No. 1, pp. 354-383, Jan, 2007.
- [219] Zhang B, Govindaraju RS., "Prediction of watershed runoff using Bayesian concepts and modular neural network," *Water Resources Research* 36(3): 753–762, 2000.
- [220] Zhiqiang Gao and Nirwan Ansari, "Tracing Cyber Attacks from the Practical Perspective", *IEEE Communications Magazine*, vol. 43, No. 5, pp.123-131, may 2005.

Publications out of the work

International Journals:

1. B. B. Gupta, R. C. Joshi, Manoj Misra, "Defending against Distributed Denial of Service Attacks: Issues and Challenges," *Information Security Journal: A Global Perspective*, Taylor & Francis, UK, pp. 224-247, 2009.
2. B. B. Gupta, R. C. Joshi, Manoj Misra, "ANN Based Scheme to Predict Number of Zombies involved in a DDoS Attack," *International Journal of Network Security (IJNS)*, vol. 13, no. 3, ISSN: 1816-3548, pp. 216-225, 2011.
3. B. B. Gupta, Manoj Misra, R. C. Joshi, "An ISP level Solution to Combat DDoS attacks using Combined Statistical Based Approach," *International Journal of Information Assurance and Security (JIAS)*, vol. 3, issue 2, ISSN: 1554-1010, Dynamic Publishers Inc., USA, pp. 102-110, 2008.
4. B. B. Gupta, R. C. Joshi, Manoj Misra, "Prediction of Number of Zombies in DDoS Attack using Polynomial Regression Model," *Journal of Advances in Information Technology (JAIT)*, vol. 2, no. 1, ISSN : 1798-2340, Academy publisher, pp. 57-62, 2011.
5. B. B. Gupta, R. C. Joshi, Manoj Misra, "Distributed Denial of Service Prevention Techniques," *International Journal of Computer and Electrical Engineering (IJCEE)*, vol. 2, number 2, ISSN: 1793-8198, Singapore, pp. 268-276, 2010.
6. B. B. Gupta, R. C. Joshi, Manoj Misra, "Dynamic and Auto Responsive Solution for Distributed Denial-of-Service Attacks Detection in ISP Network," *International Journal of Computer Theory and Engineering (IJCTE)*, vol. 1, number 1, ISSN: 1793-821X, Singapore, pp. 71-80, 2009.

7. B. B. Gupta, R. C. Joshi, Manoj Misra, "Estimating Strength of DDoS Attack using Various Regression Models," *International Journal of Multimedia Intelligence and Security (IJMIS)*, ISSN: 2042-3470, Inderscience Publisher, 2011. (To appear)
8. B. B. Gupta, R. C. Joshi, Manoj Misra, "Isotonic Regression and Correlation Analysis for Predicting Number of Zombies in DDoS Attacks," *Security and Communication Networks*, ISSN: 1939-0122, John Wiley & Sons, Inc., 2011. (To appear)
9. B. B. Gupta, R. C. Joshi, Manoj Misra, "Estimating Number of Zombies using Linear Regression Model," *Information Security Journal: A Global Perspective*, Taylor & Francis, UK, 2011. (Accepted with minor modification)
10. B. B. Gupta, R. C. Joshi and Manoj Misra, Nadeem Jamali, "Detecting Distributed Denial of Service Attacks using GARCH Model," *Computer Communication*, Elsevier, 2011. (Communicated)

Book Chapters:

1. B. B. Gupta, R. C. Joshi, Manoj Misra et. al., "Estimating Strength of a DDoS Attack using Multiple Regression Analysis," N. Meghanathan et al. (Eds.), Book on CCSIT, part III, Communications in Computer and Information Science (CCIS 133), LNCS, Springer-Verlag Berlin Heidelberg, pp. 280–289, 2010.
2. B. B. Gupta, R. C. Joshi and Manoj Misra et. al., "Predicting Number of Zombies in a DDoS Attack using ANN Based Scheme," V.V. Das, G. Thomas, and F. Lumban Gaol (Eds.), Book on AIM-2011, Communications in Computer and Information Science (CCIS 147), LNCS, Springer-Verlag Berlin Heidelberg, pp. 117-122, 2011.

International Conferences:

1. B. B. Gupta, R. C. Joshi, M. Misra, et. al., "Detecting a Wide Range of Flooding DDoS Attacks using Linear Prediction Model," *in the proceedings of 2nd IEEE International Conference on Information and Multimedia Technology (ICIMT 2010)*, pp. 535-539, December 28-30, 2010, Hong Kong, China.
2. B. B. Gupta, R. C. Joshi, Manoj Misra, "On Approximating Number of Zombies in a DDoS Attack using Linear Regression Model," *in the proceedings of IEEE International Conference on Intelligent Network and Computing (ICINC-2010)*, pp. 150-154, Nov. 26-28, 2010, Malaysia.
3. B. B. Gupta, R. C. Joshi, Manoj Misra, "Pace Regression Model for Predicting Number of Zombies in a DDoS Attack," *in proceedings of IEEE INDICON*, 2010, India.
4. B. B. Gupta, M. Misra, R. C. Joshi, "FVBA: A Combined Statistical Approach for Low Rate Degrading and High Bandwidth Disruptive DDoS Attacks Detection in ISP Domain," *in the proceedings of 16th IEEE International Conference on Networks (ICON-2008)*, DOI: 10.1109/ICON.2008.4772654, Dec. 12-14, 2008, New Delhi, India.
5. B. B. Gupta, R. C. Joshi, M. Misra, "An Efficient Analytical Solution to Thwart DDoS Attacks in Public Domain," *in the proceedings of ACM International Conference on Advances in Computer, Communication and Computing (ICAC3-2008)*, pp. 503-509, Jan. 23-24, 2009, Mumbai, India.
6. B. B. Gupta, M. Misra, R. C. Joshi, "An Integrated Approach to Counter Flooding Based DDoS Attacks in ISP Domain," *in the proceedings of IEEE International Conference on Emerging Trends in Computing (ICETIC-2009)*, pp. 115-120, Jan. 8-10, 2009, India.

7. B. B. Gupta, R. C. Joshi, M. Misra, "Defense Mechanisms against Distributed Denial of Service (DDoS) Attacks," *PhD/ECR Forum, Fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing ISSNIP 2009*, University of Melbourne, Victoria, 2009, Australia.
8. B. B. Gupta, K. Kumar, R. C. Joshi, M. Misra, "Tackling Aggressive Flows using Queue Management Techniques," *in the proceedings of International Conference on Recent advancement and Applications of Computer in Electrical engineering (RACE 2007)*, March 23-25, 2007, India.
9. B. B. Gupta, R. C. Joshi, Manoj Misra, et. al., "Estimating Number of Zombies in DDoS Attacks using Multiple Regression Analysis," *IEEE International Conference on Intelligent Information Networks (ICIIN 2011)*, 2011, Dubai, UAE. (Accepted)