**ABSTRACT**

The integration of interconnected Internet of Things (IoT) devices into the fabric of daily life has introduced significant security vulnerabilities and created a vast attack surface for cybercriminals, with botnets posing a significant threat. Cloud computing has revolutionized the way we access and manage computing resources. We propose a new framework that utilizes Long Short-Term Memory (LSTM) networks to analyse network traffic patterns and identify botnet activities on edge devices. The model benefits from fog computing infrastructure, enabling real-time detection and reducing reliance on centralized cloud resources. We evaluate our approach on a benchmark IoT botnet dataset, demonstrating its efficiency in detecting various botnet behaviors with more accuracy and low latency. This research contributes to enhancing the security and resilience of IoT networks by providing an effective and scalable botnet detection solution. A new approach to identifying botnet attacks in fog computing environments leverages the programmability of Software-Defined Networks (SDNs) to effectively neutralize these threats. This process has been evaluated using the latest data, diverse performance metrics, and modern deep-learning models. Cross-validation further confirms its effectiveness, demonstrating that it surpasses prior methods in accurately detecting 99.98% of complex and varied botnet attacks.

**LIST OF FIGURES**

**FIGURE NO FIGURE NAME PAGE NO**

Figure 3.2 Fundamental IoT Architecture 6

Figure 5.1 Proposed Architecture 9

Figure 5.3 Mirai Malware Detection Methodology 11

Figure 5.4 FOG Computing Architecture 12

Figure 5.6.3 Feature Selection 16

Figure 5.6.1 Operators of Python 20

Figure 5.7.5 LSTM with Gates 23

Figure 6.1 Data Flow Diagram 28

Figure 6.2 Flow chart Diagram 29

Figure 6.3 Sequence Diagram 30

Figure 6.4 Class Diagram 31

**LIST OF TABLES**

**TABLE NO TABLE NAME PAGE NO**

Table 5.6 Class Combinations 14

Table 9.1 Comparison work proposed with existing systems. 46

**LIST OF SCREENSHOTS**

**SCREENSHOT NO SCREENSHOT NAME PAGE NO**

Screenshot 9.1 GUI Screen 43

Screenshot 9.2 N\_BIoT 2023 dataset 44

Screenshot 9.3 Dataset Implementation 44

Screenshot 9.4 Accuracy,Precision, Recall and F1-Score 45

Screenshot 9.5 False Positive and True Positive rates 45

**TABLE OF CONTENTS**

ABSTRACT i

LIST OF FIGURES ii

LIST OF TABLES iii

LIST OF SCREENSHORTS iv

1. INTRODUCTION 1
2. LITERATURE SURVEY 3
3. SYSTEM ANALYSIS 5

3.1 Existing System 5

3.2 IOT 5

1. SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT 7

4.1 System Requirements 7

4.1.1 Hardware System Configuration 7

4.1.2 Software Requirements 7

4.2 Study of Feasibility 8

4.2.1 Economical Feasibility 8

4.2.2 Technical Feasibility 8

4.2.3 Social Feasibility 8

5. PROPOSED ARCHITECTURE 9

5.1 Proposed Architecture 9

5.2 Botnet Attacks 9

5.3 Botnet Detection Methods 10

5.4 Fog Computing Architecture 12

5.5 Modules 13

5.5.1 Service Provider for Authentication 13

5.5.2 Observe and Allow Users 13

5.6 Research Design and Research Methodology 13

5.6.1Data Set 13

5.6.2 Pre-Processing 14

5.6.3 Feature Selection 15

5.6.4 Selected Feature Selection Method for Research 16

5.6.5 Implementation 18

5.6.6Tools And Technologies Used 18

5.6.7 Classification 21

5.7 Leveraging Lstm Algorithms For Botnet Detection In

Deep Learning 21

5.7.1 Introduction 21

5.7.2 Understanding LSTM Networks 21

5.7.3 Application of LSTMs in Botnet Detection 22

5.7.4 Advantages of LSTM for Botnet Detection 22

5.7. 5 Challenges and Considerations 22

5.8 Utilizing RNN Algorithms for Botnet Detection in Deep Learning 23

5.8.1 Introduction 23

5.8.2 Understanding Recurrent Neural Networks 23

5.8.3 Application of RNNs in Botnet Detection 23

5.8.4 Advantages of RNNs for Botnet Detection 24

5.8.5 Challenges and Considerations 24

5.9 Evaluation and Decision 24

5.91 Classifier Evaluation 24

5.9.2 Decision Making 25

5.9.3 Incident Response 25

5.9.4 Continuous Improvement 25

5.9.5 Evaluation 25

5.9.6 Decision 26

6. SYSTEM DESIGN 28

6.1 Data Flow Diagram 28

6.2 Flow Chart Diagram 29

6.3 Sequence Diagram 30

6.4 Class Diagram 31

7. IMPLEMENTATION 32

8. MACHINE LEARNING /DEEP LEARNING MODEL

ACCURACY TECHNIQUES 42

9. RESULT AND DISCUSSION 43

10. SYSTEM TESTING 47

10.1 Introduction 47

10.2 Test Planning 48

10.3. Test Execution 49

10.4. Black-Box Testing 49

10.5. White-Box Testing 50

10.5.1. Types of White-Box Testing 51

11. CONCLUSION AND FUTURE WORK 52

12. BIBLIOGRAPHY 53

Appendix:

1. List of Publications
2. GitHub link