A Project Report

On

**Machine learning based trust recommendation model**

**for ballot stuffing attack in fog networks.**

BY

[Timothy Zachariah Binesh](mailto:f20212978@hyderabad.bits-pilani.ac.in)

**2021A3PS2978H**

Under the supervision of

**Prof. G Geetha Kumari**

**SUBMITTED IN FULLFILLMENT OF THE REQUIREMENTS OF**

**CS F376: Design PROJECT**



**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)**

**HYDERABAD CAMPUS**

**(March 2024**)

**ACKNOWLEDGMENTS**

****

**Birla Institute of Technology and Science-Pilani,**

**Hyderabad Campus**

**Certificate**

This is to certify that the project report entitled “**Machine learning based trust recommendation model for ballot stuffing attack in fog networks.”** submitted by Mr. **Timothy Zachariah Binesh** (ID No. 2021A3PS2978H) in complete fulfillment of the requirements of the course CS F376, Design Project Course, embodies the work done by him/her under my supervision and guidance.

**Date: 12/3/24 (Prof. G Geetha Kumari)**

BITS- Pilani, Hyderabad Campus

**ABSTRACT**

The project simulates a trust-based communication network among fog nodes, assessing node reliability through iterative packet exchanges and subsequent trust matrix adjustments. Initially, fog nodes are distributed randomly within a defined area. Each node maintains a trust matrix, reflecting its trustworthiness in communicating with other nodes. Packet exchange success or failure influences trust matrix updates, with successful transmissions enhancing trust and failures diminishing it.

Subsequently, the project introduces malicious nodes, altering their trust behavior to undermine network integrity. These nodes manipulate trust matrices to deceive neighboring nodes, thereby compromising the network's reliability. The impact of malicious nodes is evaluated by observing changes in Borda ranks, indicating a significant disruption in node trustworthiness.

To detect malicious behavior, the project employs a strategy based on rank alterations. Nodes exhibiting rank increases or decreases are scrutinized, identifying potential accomplices or victims of malicious activity. This detection mechanism aims to isolate compromised nodes and mitigate their adverse effects on network performance.

Furthermore, the project implements a mechanism to identify nodes with significant rank alterations, highlighting potential targets for further investigation or remediation. By analyzing the frequency and magnitude of rank changes, it provides insights into the extent of malicious influence within the network.

Overall, the project offers a comprehensive framework for evaluating trust dynamics and detecting malicious behavior in fog computing environments. Through iterative simulations and analysis, it contributes to enhancing network resilience and security against adversarial threats, ensuring the integrity of communication processes in distributed systems.

**CONTENTS**

Title page………………………………………………………………………..1

Acknowledgements……………………………………………………………...2

Certificate……………………………………………………………..................3

Abstract……………………………………………………………….................4

1.Modelling of Fog Network…………………………….…………...................6

2.Borda Scoring and Voting Mechanism………………………………………....7

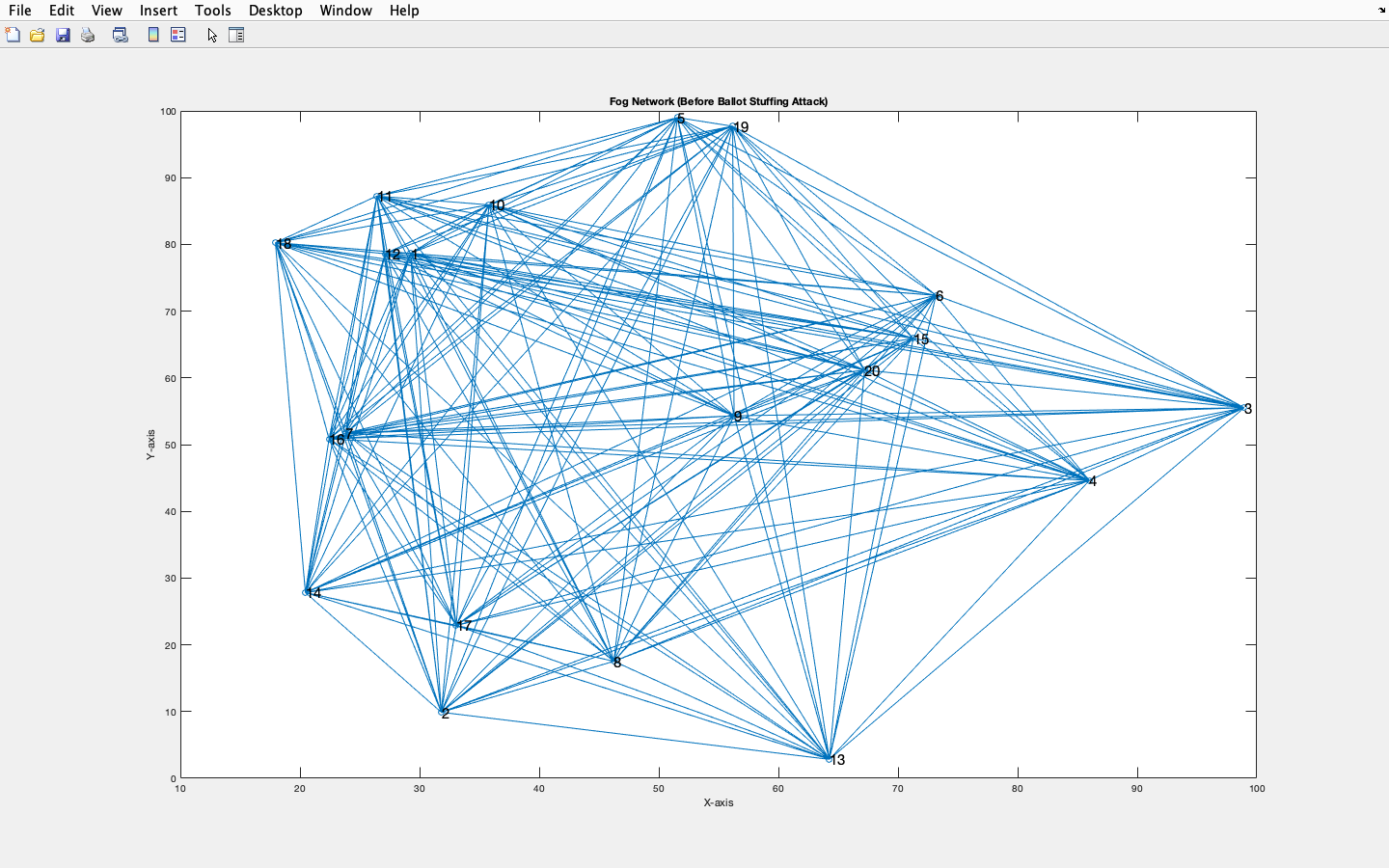
3.Ballot Stuffing Attack…………………………………………………………10

4.Malicious Nodes and Detection..……………………………………………...11

References……………………………………………………………………….13

Modelling Of Fog Networks

This MATLAB code defines a homogeneous fog network with randomly placed fog nodes in a 100x100 area. It initializes a trust matrix representing the trust level between nodes. Then, it simulates communication between nodes over multiple iterations, adjusting trust based on packet success or failure. Finally, it normalizes the trust matrix between 0 and 1. This code serves as a simulation of trust dynamics in a fog computing environment.



Borda Scoring And Voting Mechanism

The introduced MATLAB code initializes and populates a voting matrix based on the trust matrix previously defined. Each row of the voting matrix represents the voting preferences of a node, with ranks assigned to other nodes based on their trust levels. It employs a Borda Count method, where each rank corresponds to a node's relative position in the trust rankings.

The voting matrix calculation involves sorting the trust matrix columns, assigning unique ranks to nodes based on their sorted positions, and ensuring no rank is repeated within a node's preferences. The resulting matrix reflects each node's ordered preference list of other nodes in the network.

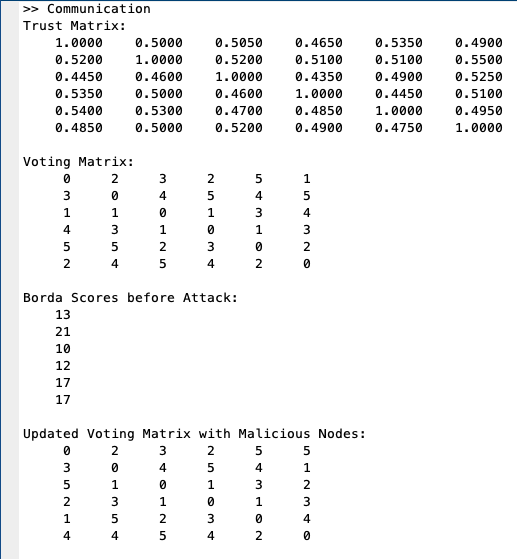
The Borda scores, obtained by summing the ranks in each row, provide a measure of overall preference or trustworthiness for each node in the network. These scores offer valuable insights into collective decision-making processes within fog computing environments, aiding in tasks such as resource allocation, task scheduling, or consensus formation among distributed nodes.

**Ballot Stuffing Attack**

This extended MATLAB code introduces malicious nodes into the previously established fog network simulation. Initially, it defines the number of malicious nodes, determining it as 30% of the total nodes. These malicious nodes are randomly selected from the pool of all nodes.

Subsequently, the voting matrix population is adjusted to reflect the presence of malicious nodes. When a node is identified as malicious, its voting behavior is manipulated to act against its original trust rankings. Specifically, the ranks assigned to nodes by the malicious entity are reversed, with nodes that were originally trusted receiving lower ranks. This alteration aims to disrupt the trust-based decision-making process within the network.

The updated voting matrix, incorporating the influence of malicious nodes, is displayed. Additionally, the Borda scores, calculated considering the altered voting preferences due to malicious behavior, are presented. These scores offer insights into the impact of malicious entities on collective decision-making processes, highlighting vulnerabilities and potential distortions introduced by adversarial actions within fog computing environments. Understanding such dynamics is crucial for devising robust strategies to mitigate the effects of malicious activities and enhance the resilience of distributed systems.



**Malicious Node Detection**

The project conducts a comprehensive analysis of node ranks and their potential vulnerabilities to attacks within a network, utilizing Borda scores as a measure of node importance. Here's a detailed explanation:

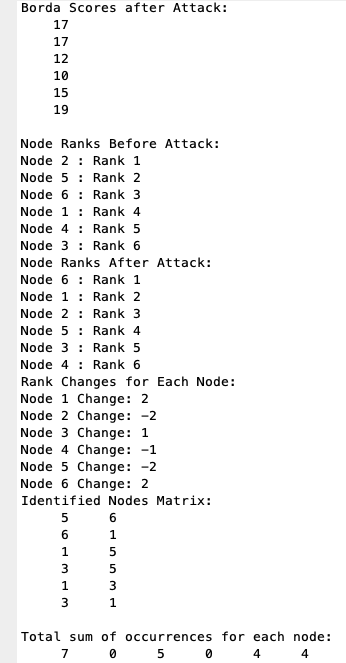
1. Ranking Nodes: Initially, the code sorts the nodes based on their Borda scores, both before and after a simulated attack. This process establishes a baseline understanding of node importance within the network.

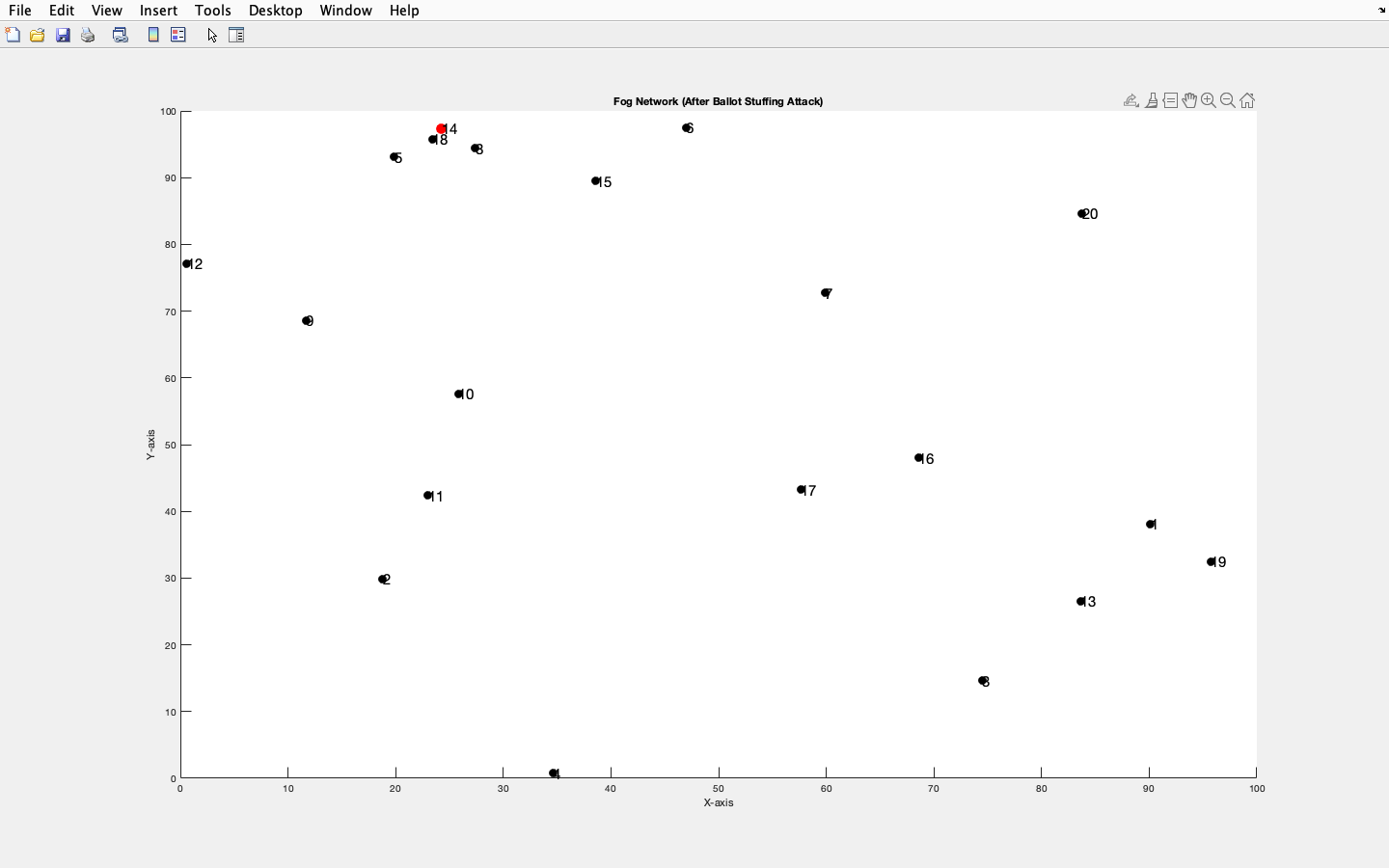
2. Calculating Rank Changes: By computing the differences between the initial and post-attack ranks, the code determines how each node's position within the network has been affected. This step is crucial for identifying nodes that have experienced significant rank fluctuations, indicating potential vulnerabilities or targeted attacks.

3. Identifying Influential Nodes: The code then proceeds to identify neighboring nodes that contributed to rank changes. For nodes experiencing rank increases, it identifies other nodes that positively influenced their positions, and for nodes with decreased ranks, it identifies nodes that may have negatively impacted them. This step helps uncover potential patterns of influence or manipulation within the network.

4. Quantifying Node Vulnerabilities: Through the process of counting occurrences, the code calculates the total sum of occurrences for each identified node. This metric provides insight into the collective impact of neighboring nodes on a particular node's rank change. Nodes with higher occurrence sums may be more susceptible to manipulation or exploitation within the network.

5. Results Presentation: Finally, the code presents its findings, including the matrix of identified nodes and the total sum of occurrences for each node. This information offers network administrators valuable insights into the network's structure, vulnerabilities, and potential points of exploitation, enabling them to take proactive measures to enhance security and resilience.

Overall, this code serves as a powerful tool for analyzing network dynamics and identifying potential security threats, facilitating informed decision-making and proactive risk mitigation strategies.



References:

### [A trust management system for fog computing services](https://www.sciencedirect.com/science/article/pii/S2542660521000263)

[SO Ogundoyin](https://scholar.google.com/citations?user=OftupnYAAAAJ&hl=en&oi=sra), [IA Kamil](https://scholar.google.com/citations?user=RTlScjMAAAAJ&hl=en&oi=sra) - Internet of Things, 2021 - Elsevier

### [A comprehensive study of simulation frameworks and research directions in fog computing](https://www.sciencedirect.com/science/article/pii/S1574013721000319)

M Gill, [D Singh](https://scholar.google.com/citations?user=YBZOirkAAAAJ&hl=en&oi=sra) - Computer Science Review, 2021 - Elsevier

### [HTML] [A survey and taxonomy of simulation environments modelling fog computing](https://www.sciencedirect.com/science/article/pii/S1569190X1930173X)

[A Markus](https://scholar.google.com/citations?user=a8Yw-dsAAAAJ&hl=en&oi=sra), [A Kertesz](https://scholar.google.com/citations?user=-ouO5T8AAAAJ&hl=en&oi=sra) - Simulation Modelling Practice and Theory, 2020 - Elsevier

### [HTML] [A survey and taxonomy of simulation environments modelling fog computing](https://www.sciencedirect.com/science/article/pii/S1569190X1930173X)

[A Markus](https://scholar.google.com/citations?user=a8Yw-dsAAAAJ&hl=en&oi=sra), [A Kertesz](https://scholar.google.com/citations?user=-ouO5T8AAAAJ&hl=en&oi=sra) - Simulation Modelling Practice and Theory, 2020 - Elsevier

### [PDF] [Graph model of Fog Computing system](https://ceur-ws.org/Vol-2850/paper2.pdf)

AV Ryabko, OV **Zaika**, RP Kukharchuk, [TA **Vakaliuk**](https://scholar.google.com/citations?user=Ka98KhMAAAAJ&hl=en&oi=sra)

Ceur workshop proceedings, 2021•ceur-ws.org

### [Towards detecting and measuring ballot stuffing](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1965735)

[D Vorobyev](https://scholar.google.com/citations?user=IX8GoVwAAAAJ&hl=en&oi=sra)