A DISTRIBUTED BLACKLISTING PROTOCOL FOR IOT DEVICE CLASSIFICATION USING THE HASHGRAPH CONSENSUS ALGORITHM

A Thesis

by

Ozan Tarlan

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Computer Science

in the
Department of Computer Science

Özyeğin University
December 2021

A DISTRIBUTED BLACKLISTING PROTOCOL FOR IOT DEVICE CLASSIFICATION USING THE HASHGRAPH CONSENSUS ALGORITHM

Approved by:

Assistant Professor Kübra Kalkan Çakmakçı, Advisor
Department of Computer Science
Özyeğin University

Associate Professor Hasan Sözer
Department of Computer Science
Özyeğin University

Professor Fatih Alagöz
Department of Computer Engineering
Boğaziçi University

Date Approved:

This thesis is dedicated to my late aunt Ülker, who wished to attend my graduation. May she rest in peace.

ABSTRACT

Industrial applications require highly reliable, secure, low-power and low-delay communications. However, wireless communication links in the industrial environment suffer from various channel impairments which can compromise these requirements. This thesis presents a new reliable blacklisting protocol for ensuring the Internet of Things (IoT) network security and mitigating the effects of interference caused by multipath Rayleigh fading using a distributed approach. The proposed blacklisting protocol is simulated over a distributed IoT network setup where flat Rayleigh fading disrupts Message Queuing Telemetry Transport (MQTT) communications. Distributed servers jointly decide in real-time whether to blacklist a device after individually performing anomaly detection and submitting their results to the hashgraph network. The IoT devices are classified by a device fingerprinting method using various machine learning (ML) algorithms that are trained with real-time packet capture data. The proposed blacklisting protocol is shown to increase the accuracy of blacklisting malignant devices from 42% to 82% as the number of servers increases from one to five for mixed attacks. It also achieves higher accuracies ranging between 47.2%-97.6% versus 47.4%-90.7% compared to the related work for Denial of Service (DoS) attacks. For this simulation with multiple servers, a novel solution for reducing the PER in a Rayleigh fading environment by searching optimal distributed server locations with a hybrid approach is also provided. The proposed protocol and the server location optimization algorithm are particularly suitable for the Industrial IoT (IIoT) in mitigating the effects of harsh communication environments in manufacturing facilities.

ÖZETÇE

Endüstriyel uygulamalar son derece güvenilir, güvenli, düşük güçte ve düşük gecikmeli iletişim gerektirmektedir. Ancak, endüstriyel uygulamaların bulunduğu ortamlarda kablosuz iletişim bağlantıları, bu gereksinimleri tehlikeye atabilecek çeşitli engellerle karşı karşıyadır. Bu tezde, Nesnelerin İnternetinde (IoT) ağ güvenliğini sağlayarak ve dağıtık sunucular kullanarak Rayleigh sönümlemesinin etkilerini azaltmak için yeni bir güvenilir kara listeleme protokolü bulunmaktadır. Önerilen kara listeleme protokolü, Flat Rayleigh sönümlemesini Message Queuing Telemetry Transport (MQTT) iletişimini engellediği dağıtık IoT ağı üzerinde simüle edilmektedir. Sunucular, anormallik tespiti gerçekleştirip sonuçlarını hashgraph ağına gönderdikten sonra, cihazların kara listeye alınmasına gerçek zamanlı olarak ortaklaşa karar verir. IoT cihazları, gerçek zamanlı pcap verileriyle eğitilmiş çeşitli makine öğrenmesi algoritmaları ile sınıflandırılır. Önerilen kara listeme protokolünün, çeşitli saldırılar için sunucu sayısı birden beşe çıkarken, cihazları kara listeye alma doğruluğunu %42'den %82'ye çıkardığı gösterilmiştir. Ayrıca, DoS saldırıları için ilgili çalışmaya kıyasla sunucu sayısına göre %47,2-%97.6 arasında değişen başarımlarla, ilgili çalışmadan (%47.4-%90.7) daha yüksek başarımlar elde etmektedir. Çoklu sunuculu bu simülasyon için, bir Rayleigh sönümleme ortamında optimal dağıtık sunucu konumlarını hibrit bir algoritma ile arayarak paket hata oranını azaltmak için yeni bir çözüm de sağlanmıştır. Önerilen protokol ve sunucu konumu optimizasyon algoritması, özellikle endüstriyel tesislerinin sönümlemeli iletişim ortamlarının etkilerini azaltmak için uygundur.

ACKNOWLEDGEMENTS

I would like to express my special thanks of gratitude to my Advisors; Kübra Kalkan Çakmakçı and Ilgın Şafak for their guidance and support. Their valuable feedback and constant motivation pushed me through the isolation and fear in the pandemic. The everlasting love that my family have given me kept me safe and warm. Throughout my life they have been helping me selflessly. And the "family nights" was a breath of fresh air that was a fun activity that kept me sane through the pandemic. My friends always found a way to cheer me up, even through a computer screen, and the idea that sometime in the near future we will continue to hang out was my the light at the end of the tunnel. My friend Jimbo helped me so much since our academic fields of interest were very similar. And finally, my cousin İrem who is a great psychiatrist kept me motivated with her words of wisdom.

TABLE OF CONTENTS

DEDICATION . . . . . **iii**

ABSTRACT . . . . . **iv**

ÖZETÇE . . . . . **v**

ACKNOWLEDGEMENTS . . . . . **vi**

LIST OF TABLES . . . . . **ix**

LIST OF FIGURES . . . . . **x**

I INTRODUCTION **1**

1.1 Motivation 1

1.2 Concepts 1

1.2.1 Blacklisting 2

1.2.2 MQTT 2

1.2.3 Multipath fading 3

1.2.4 Hashgraph Consensus Algorithm 4

1.2.5 Server Placement Optimization 6

1.3 Contributions 7

II LITERATURE REVIEW **9**

III BLACKLISTING PROTOCOL **11**

3.1 System Architecture 11

3.2 Flat Rayleigh Fading in the Network Simulation 13

3.3 The Blacklisting protocol: DiBLIoT 13

3.4 Simulation Setup 16

3.5 Server Location Optimization 17

3.5.1 IoT Device Setup 17

3.5.2 Optimization Metric 18

3.5.3 Genetic Algorithm 19

3.5.4 Simulated Annealing 20

3.5.5 K-Means Cluster Centroids 21

3.5.6 Novel Hybrid Approach 22

IV PERFORMANCE EVALUATION 23

4.1 Simulating Flat Rayleigh Fading 23

4.2 Datasets 24

4.3 Performance Evaluation of Classifiers 25

4.4 Distributed Server's Effect on Accuracy 26

4.5 Comparison of Server Location Optimization Algorithms 27

V CONCLUSION 29

APPENDIX A — FLAT RAYLEIGH FADING GRAPH 30

REFERENCES 31

VITA 33

LIST OF TABLES

1 Performance Metrics of the Classifiers 25

LIST OF FIGURES

1 MQTT Publish Subscribe communication 3
2 Multipath Fading 4
3 Hashgraph 5
4 System Architecture 12
5 Blacklisting protocol sequence 15
6 IoT device location representation 18
7 Optimization function 18
8 Crossover description 20
9 Elbow plot example 22
10 Average PER vs. SNR 24
11 Accuracy rate for the server count 27
12 Accuracy rate for the server count 28
13 SNR over time 30
14 PER over time 30

CHAPTER I

INTRODUCTION

Industry 4.0, the digital transformation of industry, which encompasses the IIoT and smart manufacturing, focuses heavily on inter-connectivity, automation, ML, big data and real-time communications. IIoT requires industrial devices such as sensors and robots to communicate autonomously, securely, robustly, and efficiently, and relies heavily on wireless communications. However, wireless communication links in the industrial environment, as in manufacturing lines, suffer from channel impairments due to path losses, shadowing and multipath fading due to obstacles on the communication path. The resulting random fluctuations in the received signal level will impair communications and end-to-end reliability [1].

1.1 Motivation

The motivation for this thesis is to secure IoT Network through blacklisting with machine learning anomaly detection. However blacklisting in a real IIoT network environments are affected by Multipath Fading. Which disrupts wireless communication accuracy. To protect the autonomous, robust and secure nature of the sensors in the industrial facility, multipath fading issue and malicious devices needs to be addressed. To mitigate this Multipath Fading, utilizing distributed servers and finding the optimal placements for these servers is crucial. The use of multiple servers can increase the blacklisting accuracy in an environment with multipath fading. However this blacklisting information must be shared securely and efficiently between the servers. Therefore we need to explore the fast and fault tolerant distributed ledgers like Hashgraph to share this blacklisting information.

1.2 Concepts

In this section, background information on the main topics of the thesis is provided.

1.2.1 Blacklisting

Blacklisting is the practice of rejecting devices of certain properties. In this thesis, malicious devices are blacklisted based on anomalous behavior of network traffic using ML techniques.

1.2.2 MQTT

MQTT [2] is a network protocol specifically designed for constrained and heterogeneous networks. This design allows various low complexity devices to communicate with the large-scale environment making this protocol a great solution for IoT device communication.

MQTT is a one to many communication protocol with publish-subscribe mechanism. MQTT protocol has clients and a broker. Clients can communicate, send or receive messages in the MQTT network. To send messages clients need to publish messages to some topic. As a message is published to a topic, the subscribers of the topic receives the message. To receive messages, clients can subscribe to some topic. As a client subscribes to some topic, the published messages of the topic are sent to the client. A topic might have many subscribers, therefore a published message of a topic must be sent to all of the subscribers. Instead of making the clients responsible for sending the same message to all subscribers and keeping track of the subscribers of the topic, the MQTT broker handles this process. MQTT Broker receives the published message once from the client and sends the message to all of the subscribers of that topic. This procedure is especially beneficial for low complexity devices since the broker reduces the complexity of the communication process. This communication is described in Fig 1. In Fig 1. with the help of the broker, publishing client is not responsible with communicating, keeping track of the subscribers or sending the message multiple times. The client just sends the publish message to the broker with the topic a. Broker handles the rest of the process.

The MQTT protocol used in this thesis as the IoT communication protocol for

the efficient properties described above.

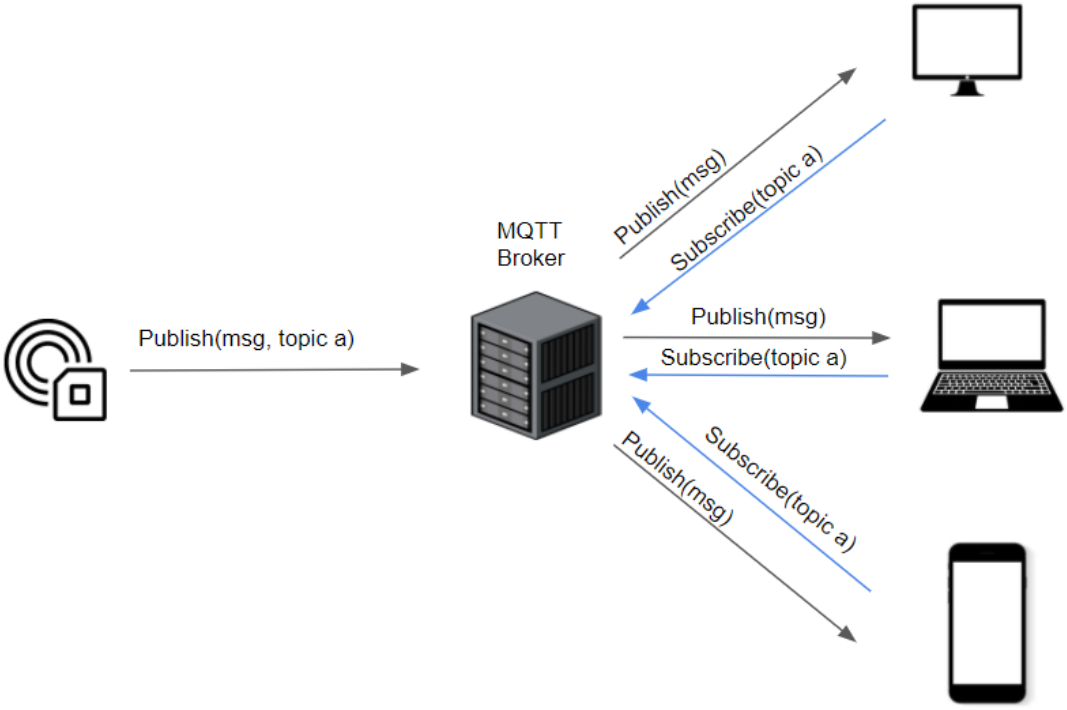

Figure 1: MQTT Publish Subscribe communication

1.2.3 Multipath fading

Multipath fading [3] is the phenomenon which radio frequency signals reflect from objects between the transmitter and the receiver in wireless channels. This reflection causes new paths that the signal can be transferred. These paths are added up on the receiver's end, causing the alteration of the signal strength this phenomena is described in fig 2. In a real-world IIoT scenario, multipath fading causes packet loss in wireless channels due to the fluctuating signal strength. Addressing this issue is crucial for the Industrial IoT device communication scenario. The impacts of flat Rayleigh fading on blacklisting has been studied in this thesis. As a mitigation technique, distributed blacklisting using multiple servers is proposed. It is shown that, as the number of servers in the network increases, the chance of receiving the packet in at least one of the servers increases.

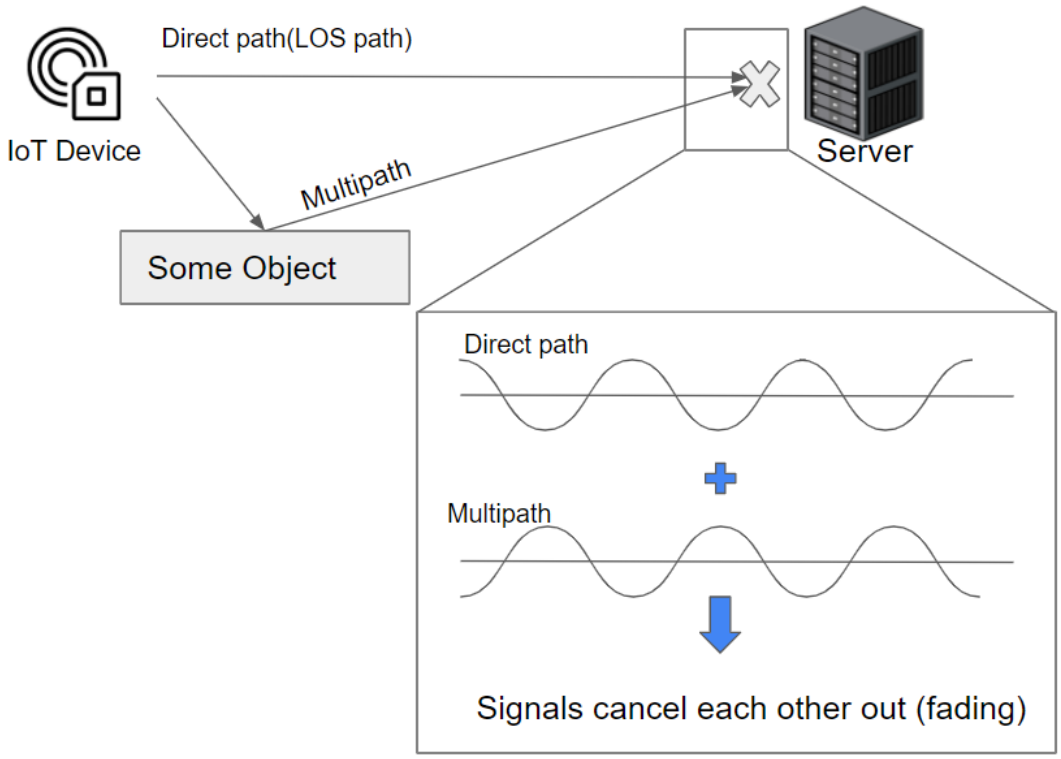

Figure 2: Multipath Fading

1.2.4 Hashgraph Consensus Algorithm

The Hashgraph consensus algorithm [4] of the Hashgraph network secures the data integrity in a network with multiple nodes. Hashgraph consensus algorithm provides fast, fair, and byzantine fault-tolerant transactions. Unlike the algorithm used by blockchain, the hashgraph algorithm is fast and requires a small communication overhead.

Hashgraph consensus algorithm allows multiple nodes in a network to jointly decide properties of data and its order without relying on trusting specific nodes in the network. This consensus algorithm is used for storing data through the independent nodes of the network instead of using a centralized database. This practice is also known as distributed ledger technology. In the hashgraph ledger, container of transactions do not require to be discarded like blockchain[5].

Hashgraph consensus algorithm uses gossip protocol[4] continuously to share every transaction known by a node to some random node to achieve agreement in

the network. The data structure that stores the gossip interactions is called the hashgraph. In Fig. 3 hashgraph data structure is presented. The lines shows the gossip interaction between nodes. Vertical lines displays the time of the gossip. Names on the top presents individual nodes in the network.

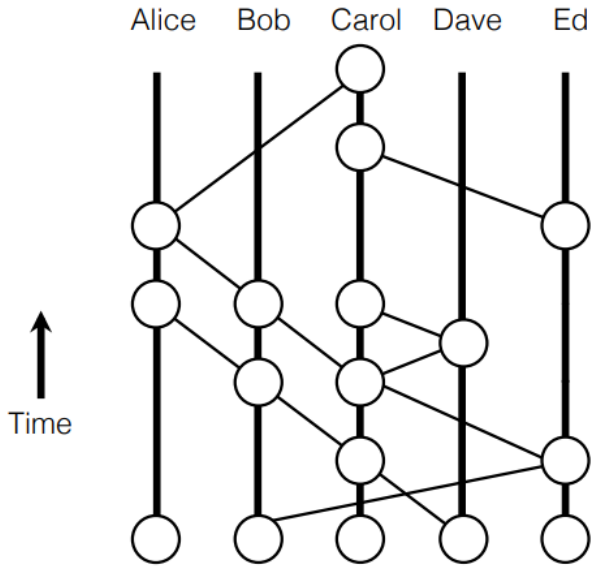

Figure 3: Hashgraph

By adding every transaction to the ledger, hashgraph network claims to provide stronger mathematical guarantees[5] than blockchain. These mathematical guarantees are Byzantine agreement and fairness. Byzantine agreement is the feature of distributed systems where distributed nodes reach to an agreement on some data property even when there is some nodes that fail to provide the correct data information. In this context, fairness means a distributed node does not have the privilege to assign a timestamp to a transaction. The timestamps are assigned through voting of the nodes. This also provides a fair transaction order. A node also cannot prevent other node's transactions. The transactions are added to the ledger through the gossip protocol of hashgraph consensus to achieve fairness. These mathematical guarantees are claimed with the assumption of 2/3 of the nodes are correctly following the hashgraph consensus procedure.

Unlike blockchain, Hashgraph network transactions[5] do not need any mechanism to slow down the process to maintain the ledger's integrity. This property of Hashgraph consensus algorithm provides faster and energy efficient transactions.

In this thesis, distributed blacklisting is performed using the hashgraph network. It is shown that sharing the blacklisted clients through the hashgraph network increases the blacklisting accuracy of the network.

1.2.5 Server Placement Optimization

In this thesis, the MQTT server locations are selected according to their proximity to the IoT devices by minimizing the packet error rate (PER) in a flat Rayleigh fading channel. The MQTT server location selection is decided by using the genetic algorithm, simulated annealing, K means clustering and a novel algorithm.

Genetic algorithm [6] is a heuristic search algorithm that is inspired by the process of natural selection and genetics. In the genetic algorithm process of natural selection is simulated for finding an optimal solution. The better performing solutions in the set get to pass their data to the next generation set through performing crossover. The next set of solutions has a chance of altering the data by the process called mutation. Mutations that yield better solutions can pass their data to the next set of solutions. This process produces better results for each iteration.

Simulated annealing [7] is a stochastic search algorithm used for finding an optimal solution. Since the solution is searched for multiple 3D coordinates and each coordinate can be real numbers, ordered search for the optimal solution is very expensive. Simulated annealing algorithm searches the optimal solution randomly and has a probability of accepting sub-optimum solutions that escape local optimas in the function.

K-means [8] is an ML algorithm used to group data entries with similar properties without the need of any label or supervision. In this thesis, the k-means algorithm is used on 3D IoT device coordinates to obtain server locations using the cluster centroids of each cluster.

We present a novel hybrid approach, which combines the k-means algorithm and genetic algorithm. To the best of our knowledge, we are first to combine k-means clustering and the genetic algorithm to find an optimal server placement using theoretical packet error rate (PER). This approach is much faster than the generic algorithm used to solve this optimization problem. Also, a hybrid approach provides a better mean theoretical PER result than k-means clustering used to find a server placement.

1.3 Contributions

To the best of our knowledge, the distributed blacklisting protocol proposed in this paper is the first to address the impacts of Rayleigh fading using Hashgraphs in distributed IoT networks. This paper also presents a novel algorithm that combines the k-means and genetic algorithm in selecting optimal distributed server locations. The contributions of this paper are as follows:

1. A new distributed blacklisting protocol is presented for mitigating the effects of flat Rayleigh fading in a distributed IoT network by performing anomaly detection.

2. Distributed network simulation is performed using the MQTT protocol in a flat Rayleigh fading channel to test the effectiveness of the protocol.

3. A real-time blacklisting accuracy comparison with existing work over a Flat Rayleigh Fading channel is provided. To the best of our knowledge, this thesis is the first to evaluate the impacts of Rayleigh fading in a blacklisting protocol.

4. A detailed anomaly detection performance analysis of the Random Forest, Decision Tree, Naive Bayes, and Logistic Regression classifiers is provided.

5. A novel server placement optimization algorithm presented and compared with k-means, genetic algorithm and simulated annealing solutions.

The remainder of this thesis is structured as follows; Chapter II presents the existing approaches and related work. Chapter III, presents the system model, blacklisting protocol and the network setup. Chapter IV presents the experimental work performed namely the network simulation and its results. Finally, Chapter V presents the conclusions.

CHAPTER II

LITERATURE REVIEW

In this section, a brief summary of related work that focuses on blacklisting in IoT is provided.

In [9], a secure IoT communication framework for Bluetooth low energy-based (BLE) networks that guards against device spoofing is proposed. Device fingerprints, which are generated using ML techniques, are employed for device authentication both before the connection setup and during session establishment. A whitelisting and blacklisting approach is used in distinguishing trusted and malicious devices in the network. However, the paper neither studies the effects of fading, nor utilizes a distributed approach in blacklisting.

In [10], a DNS (Domain Name System)-based profiling scheme over real datasets of Mirai-alike botnet activity captured on globally distributed honeypots is provided. The developed feature set over various ML classifiers are shown to significantly reduce botnet detection time whilst simultaneously maintaining high levels of accuracy (99% on average) for the Random Forest classifier. However, the effects of multipath fading on threat detection is not studied.

In [11], an ML-based IoT distributed denial of service (DDoS) attack detection system using feature extraction is proposed. The system is shown to have to high DDoS attack detection accuracy. A software defined network (SDN) controller blocks malicious devices using blacklists generated by the IoT DDoS attack detection system. However, impacts of multipath fading is not considered.

In [12] a new protocol for the authentication of IoT devices and CPANs (Personal Area Network Coordinators) in an IIoT environment is proposed. A Blockchain-based authentication approach is used in eliminating the need for

blacklisting by introducing a new way of generating IIoT device identifiers. However, all authenticated devices are assumed to be benign, and anomaly detection is not performed to identify malicious device behavior. Additionally, the impact of multipath fading is not addressed.

In [13], a method is proposed for the identification of DDoS attacks by investigating the Packet Reception Ratio. A rate monitoring algorithm examines the incoming traffic rate from connected nodes and blacklists the ones with increased rates due to malicious behavior. Other types of security attacks are not considered, and the impacts of multipath fading or distributed blacklisting are not addressed.

In [14], an intrusion detection mechanism in IoT for identifying Blackhole attack on Routing Protocol for Low-Power and Lossy Networks is proposed. The approach filters out suspicious nodes first and then verifies the behavior of those nodes only. It is shown that the Packet Delivery Ratio (PDR) of the system is improved by blacklisting of malicious Blackhole nodes. However, other security threats, impacts of multipath fading or distributed blacklisting are not studied.

In [15], a new anomaly detection protocol is proposed for IoT networks that utilize sparse autoencoders and neural networks in order to identify anomalies from bidirectional TCP communication flows. Detected anomalies are blacklisted with the mitigation module described in the paper. However, the effects of multipath fading and distributed server architecture are not considered.

CHAPTER III

BLACKLISTING PROTOCOL

This chapter describes the blacklisting protocol and the network simulation in detail. Presented blacklisting protocol and the server location optimization is designed for distributed servers where fading disrupts the communication of IoT devices. To mitigate the effects of flat Rayleigh fading, protocol is designed to work with distributed servers. Using distributed servers increases the chance of receiving packets because of the fading causes disruption for each server uniquely and independently. The blacklisting information of the successfully received packets will be sent through hashgraph network to inform the other servers that did not receive the packet. This protocol is aimed to be suitable for industrial facilities.

The network simulation is prepared for evaluating the presented blacklisting protocol. The system architecture with distributed MQTT servers, the mixture of malicious and benign clients, numerous attack types, signal that is affected by flat Rayleigh Fading environment, specific locations of IoT devices are all constructed to assess the blacklisting protocol.

In this chapter, first section describes the actors in the network, second chapter explains flat Rayleigh fading phenomena and why using distributed network would increase reliability of the connection, third section describes the blacklisting protocol in detail. Fourth section explains the MQTT protocol properties and the and the anomaly detection. And Finally fifth section describes optimal server placement methods for the given IoT device locations.

3.1 System Architecture

The system architecture is shown in Fig. 4, where clients communicate with the server (MQTT broker or broker) using MQTT communications. The communication link between the clients and broker is assumed to undergo Rayleigh fading.

Blacklisting is performed across the brokers via the Hashgraph network. It is assumed that there exists no authentication or authorization process within the network, therefore any client can join the network at any time. The system is secured from various attacks by blacklisting the devices that send malicious packets. A description of each system component is provided below.

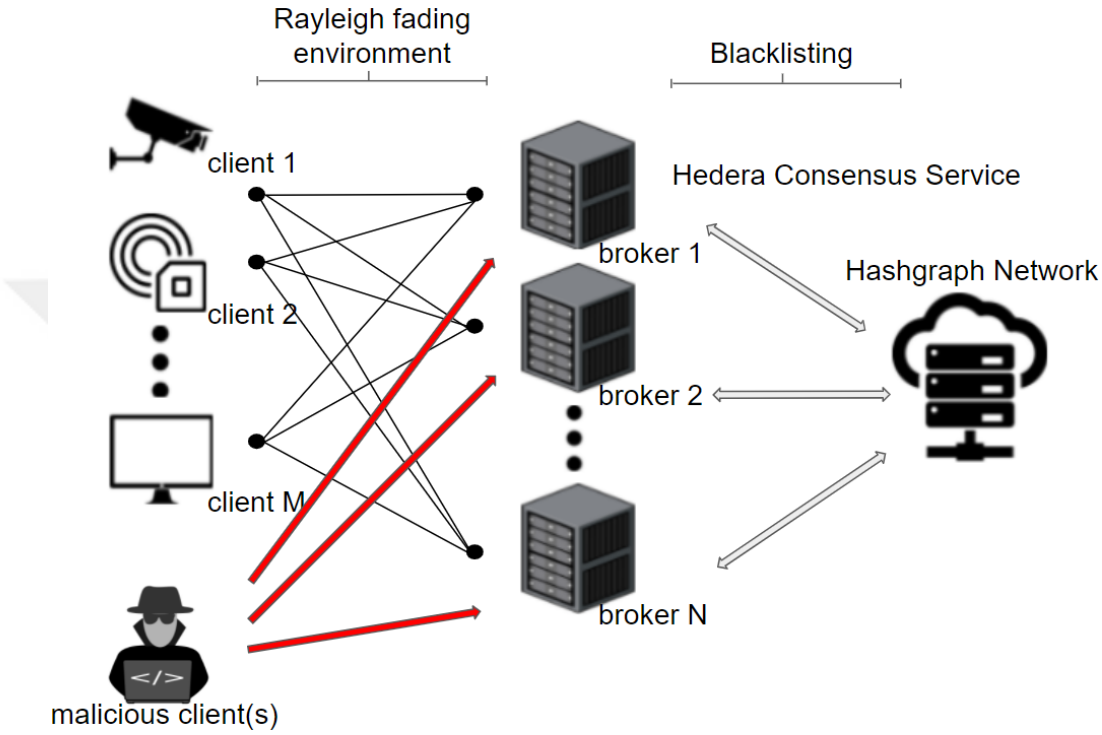

Figure 4: System Architecture

Client: Clients are the users' IoT devices in the network that use the MQTT messaging service provided by the distributed servers. Clients are assumed to consist of legitimate and malicious devices. The complexity of the clients varies from high complexity devices such as computers and smartphones to low complexity devices such as basic sensors.

Broker (Server): In the network simulation setup, servers are MQTT brokers that manage the MQTT messages for the clients as well as performing anomaly detection, blacklisting via the hashgraph and device fingerprinting for ensuring the security of the network.

Hashgraph Network: The distributed ledger is used to share the blacklist information among the MQTT servers using the Hedera Hashgraph Consensus Service. Hashgraph consensus service provides fast, fair, and byzantine fault-tolerant communications and transactions in the network.

3.2 Flat Rayleigh Fading in the Network Simulation

In a real wireless channel, transmitted radio frequency (RF) signals reach the receiver via multiple paths due to scattering by multiple objects located in the propagation path. These multipath signals are added up at the receiver. Depending on the out-of-phase or in-phase addition of the multipath signals at the receiver, we observe flares and fades in the received signal strength. This phenomenon is called multipath fading [3]. If the delay spread of the multipath signal components is a small fraction of the symbol duration, then fading is assumed to be flat. In this thesis, we assume that fading is flat and is described by Rayleigh statistics. In our network simulation, the MQTT servers are assumed to have different locations. This implies a unique flat Rayleigh fading channel profile for each server. Due to independence of Rayleigh fading, a packet is received correctly at some servers, while it may be received erroneously at some others. Diversity reception with multiple servers introduces diversity, i.e., higher probability of correct reception of the transmitted data. When the number of brokers in the network increases, one achieves a higher probability for correctly receiving the transmitted packets. Therefore, use of a distributed architecture in a flat Rayleigh fading channel leads to increased reliability in IoT communications. To the best of our knowledge, the setup proposed in this thesis is the first to address the impacts of Rayleigh fading in distributed IoT networks.

3.3 The Blacklisting protocol: DiBLIoT

The Hedera Hashgraph Consensus service is used for the DiBLIoT blacklisting protocol. The consensus algorithm [4] is a procedure across distributed processes for achieving agreement over the properties of a certain data. Consensus algorithms

protect the data integrity in a network with unreliable nodes. Hashgraph consensus algorithm is a new method used in distributed ledger technology that offers fast, fair, and byzantine fault-tolerant transactions. Unlike Blockchain, Hashgraph is fast, requires very little communication overhead beyond the transactions, and does not require excessive computation that consumes electricity. The reliability, security, and efficiency of Hashgraph proves advantageous for the distributed blacklisting protocol. Hedera [16] provides the same functionality of the hashgraph consensus algorithm as a cloud service. This offloads hashgraph consensus computations in the distributed network to the Hashgraph cloud. Distributed servers are authorized to use the pre-built hashgraph consensus service at the start of the simulation. Server instances are all subscribed to the preshared topic that is created for blacklist records. Published messages to the blacklist topic will be directed to the topic's subscribers. Detected anomalies are shared by each server via the hashgraph network to make the information visible for all the servers in the distributed IoT network. Servers blacklist the malicious clients through the Hashgraph, which is an immutable and byzantine fault-tolerant distributed ledger. The use of distributed servers provides diversity, as it makes it possible to achieve more consistent communication and more accurate results in blacklisting, specifically when MQTT QoS is 0 and flat Rayleigh fading causes high packet error rates (PER). Every detected anomaly is shared through the hashgraph network to inform distributed servers experiencing packet drop in that anomaly instance. This process leads to higher accuracies in all distributed brokers without a centralized database. By using a distributed ledger for the blacklist, detected anomaly is blacklisted by all distributed servers. Feature composition methodology is used in evaluating the performance of the anomaly classifier.

The blacklisting protocol presented in this paper is compared to [10], which both use real-time anomaly detection using ML and blacklisting. Also, the acquired features in our work and [10] are obtained from the network traffic. The proposed blacklisting protocol (see Fig. 5) is as follows:

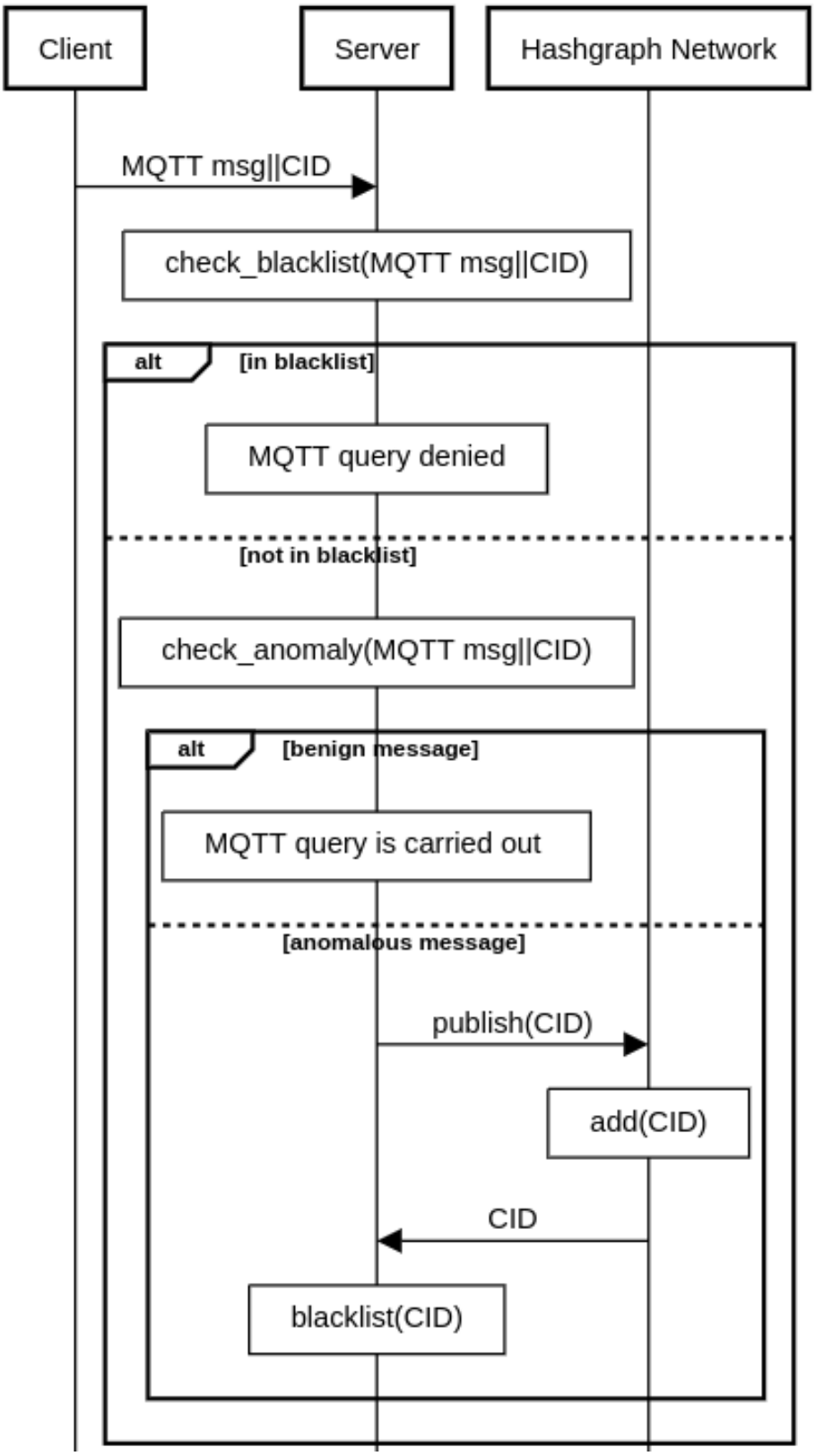

Figure 5: Blacklisting protocol sequence

1. Check the client identifier (ID) against the blacklist. Blacklisted devices

cannot connect, publish, or subscribe messages in the network. If the device is blacklisted, terminate the connection. Otherwise continue to the next step.

2. For each MQTT message sent by any client, perform device fingerprinting and anomaly detection using ML. If the device displays an anomaly, terminate the connection, otherwise continue to next step.

3. Whenever any form of anomaly is classified by the ML algorithm, the MQTT server sends the device ID of the malicious device with a publish message to the Hedera consensus service. By using the hashgraph network, nodes store blacklisted client information without a centralized database. Since all MQTT servers are subscribed to this topic from the beginning of the simulation, every server receives the published message.

4. The MQTT server uses the hashgraph service to check the ID against the ledger. The MQTT server blacklists the device by adding its ID to the blacklist.

3.4 Simulation Setup

There may be one to five MQTT servers in the distributed IoT network jointly performing blacklisting in the presence of flat Rayleigh fading. An increase in accuracy is observed when the MQTT QoS is set to "0". QoS 0 in MQTT means that each message will be sent at most once with no guarantee of delivery. In a wireless network with multipath fading, sending messages at most once may result in packet loss. However, this packet loss is mitigated by connecting clients to the servers by joint blacklisting. In the network, the clients are assumed to be connected to every server and can publish and subscribe to the MQTT topics. Anomaly detection is performed by the distributed servers using ML techniques. The ML algorithms that are trained with real MQTT packet capture data are used in identifying the various attack types and differentiating between IoT devices from non-IoT devices.

To get the most accurate classifier for these identification processes, different ML classifiers are tested and the best performing ML algorithms are used in real-time anomaly detection in the network simulation. The results are provided in Section IV. Device fingerprinting is used in classifying a device based on its data. In the network simulation, device fingerprinting is carried out by classifying devices as IoT and non-IoT. The classification task is done by an ML algorithm. The ML algorithm is trained by real-world packet capture data that consists of both IoT and non-IoT devices. This approach can be useful for providing specific services for devices of different complexity. The experimental network setup is comprised of five legitimate, five malicious devices and distributed servers ranging between one and five. The clients are subscribed to the same topic for messaging. Each client is connected to all of the distributed servers. Clients generate the network traffic by publishing messages to the subscribed topic with a Poisson distribution, where $\lambda = 1$, meaning the rate of publish message occurrence throughout the simulation time-frame is 1 on average. MQTT QoS is set to zero, which implies that every message is sent at most once with no guarantee of arrival.

3.5 Server Location Optimization

Our assumption for the previous iteration of the simulation setup was that the distributed servers are equidistant to all IoT devices. For an IIoT scenario, this is not very realistic. As well as it is infeasible from a three-dimensional standpoint. As a follow up for the conference paper, we tried to optimize the distributed server locations over the 3D IoT device locations.

3.5.1 IoT Device Setup

75 % of the IoT devices are placed with the multivariate normal distribution on some arbitrary point for clusters. This means that the device locations are set as clusters with varying densities. The remaining 25% of the devices are placed randomly within the range -100 and 100. Example setup with three clusters is shown in Fig 6.

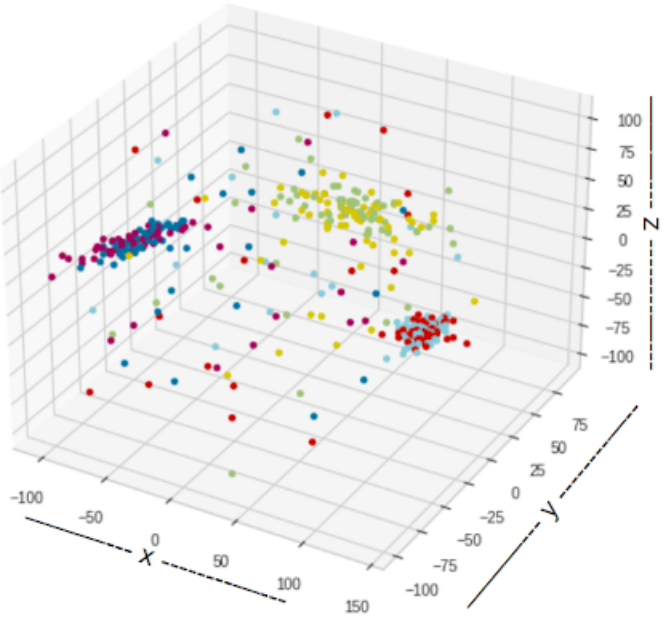

Figure 6: IoT device location representation

3.5.2 Optimization Metric

This method is used to find the optimal placement for each server in the simulation. This method does that by taking the average of theoretical PER values for the IoT devices to the closest server. This method is used in simulated annealing, genetic algorithms and our hybrid approach. The pseudocode is shown in Fig 7.

Algorithm 1 Optimization Function

```
1: procedure MEAN BER(ServerCoordinates, IoTCoordinates)
2:     total bit error rate ← 0
3:     for each IoT device do
4:         closest server = find closest server(IoT device)
5:         min distance += distance(closest server, IoT device)
6:         total bit error rate += theoretical bit error rate(min distance)
7:     end for
8:     return return total bit error rate / IoT device count
9: end procedure
```

Figure 7: Optimization function

3.5.3 Genetic Algorithm

3.5.3.1 Initialization of The Genetic Algorithm

First generation of server coordinates are initialized randomly. Each generation consists of 50 solutions. Each solution is a list of 3D coordinates.

3.5.3.2 Mutation

The mutation function alters the coordinates randomly within the bounds of the facility. This alteration is done by randomly generating values and adding those values to the coordinate set if the resulting coordinate set is in range of the boundaries. Otherwise, a new set of random numbers are generated until a coordinate set within the bounds of the simulation is obtained. Random values for the coordinate set are picked between the range of 5% of the simulation boundaries to reduce the probability of generating coordinates that are out of bounds.

3.5.3.3 Tournament

The purpose of the tournament method is to select successful solutions. The tournament method takes a list of server coordinates, and tournament size as parameters. The method randomly puts server coordinates into a new empty list from the list of server coordinates until the tournament size value is reached. The new list generated is searched for the best and the second-best coordinate values. The best coordinates are determined with the optimization function.

3.5.3.4 Crossover

The crossover method takes two server coordinate values as parameters. It flattens these two-dimensional (2D) lists into a one dimensional list. Then it randomly chooses two different indices and the values between those indices are swapped. The flat lists are then resized to 2D lists and crossover method returns the modified server coordinate lists. Crossover depiction is shown in Fig 8.

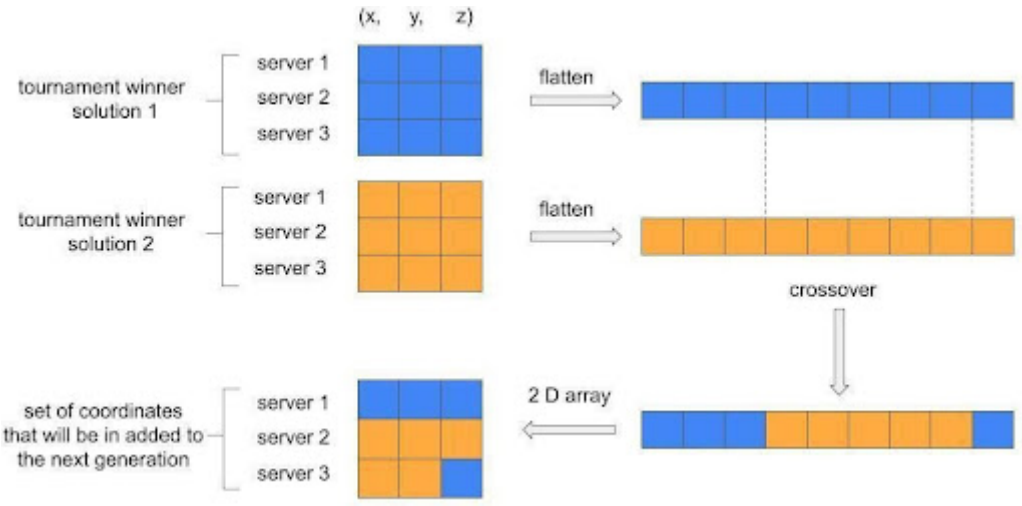

Figure 8: Crossover description

3.5.3.5 Evolve

The Evolve method takes the server coordinate list as the parameter and generates the next steps server coordinate list. The method calls the tournament method and gets two server coordinate values. The method uses these two server coordinate values as the parameters for the crossover method. The crossover method returns a server coordinate value. Then this resulting value is mutated with the mutation probability. The mutation method is called for some instances and the server coordinate values are altered. The resulting server coordinate values are put into the next steps coordinate list. After the next step's coordinate list is successfully filled with the values, the list is returned. The evolve method is run for a fixed number of times and for the final iterations best server coordinate values are set as the solution to the problem.

3.5.4 Simulated Annealing

Simulated annealing takes boundary value, iteration count, step size and temp as method parameters. Simulated annealing starts with a random coordinate list. Then the optimization score for that list is calculated. The random coordinate list and its optimization score is set as the current list and current score. Then the loop that runs for the iteration count starts. Inside that loop, a candidate

value is generated by adding a random coordinate list multiplied by the step size parameter to the current list. If the resulting coordinate set is out of the bounds of the simulation then the candidate value generation process is repeated until a value within the bounds is obtained. Afterwards, the optimization score of that coordinate is set. If the candidate coordinate list has a better optimization score, the best value and the best coordinate list is updated. The difference between the current optimization score and the best optimization score is calculated, which is used on the metropolis acceptance criterion. This criterion is used as a probability to accept a candidate coordinate list. Metropolis acceptance criterion is used for leaving a local minima in the earlier iterations of simulated annealing algorithm.

3.5.5 K-Means Cluster Centroids

In the k-means approach, IoT coordinates are clustered by their 3D locations. k-means works for the set of coordinates and takes the k amount of cluster centers. The algorithm starts with assigning some arbitrary three dimensional coordinates. For these 3D coordinates, IoT coordinates are assigned in a cluster set. The closest arbitrary 3D coordinate generated is the deciding factor for which set an IoT coordinate will be assigned to. After every IoT coordinate assignment is complete, the arbitrary coordinates are updated as the mean of assigned IoT coordinates. The updated mean coordinates are used for re-assigning IoT coordinates to the cluster set. The mean value update and the cluster assignment is repeated until convergence. The mean of each cluster is used as the server coordinate.

The k-means algorithm is also utilized in deciding the number of servers to use for the given IoT coordinates. By using the k elbow visualizer of the Yellowbrick library in Python, elbow value decision is done automatically. K elbow visualizer takes the k-means as a parameter in the elbow value decision. The distributed server amount is set as the elbow value for every algorithm used. K elbow visualizer is shown in Fig 9.

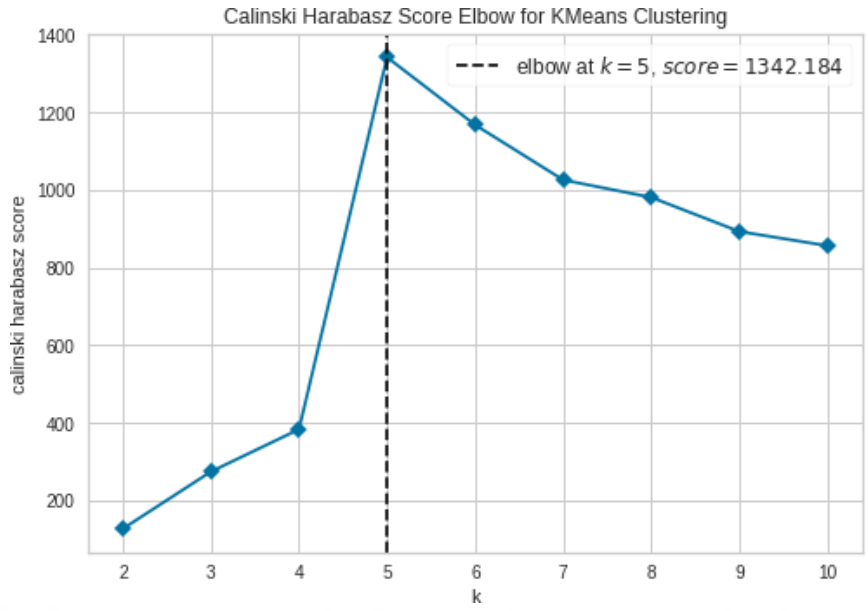

Figure 9: Elbow plot example

3.5.6 Novel Hybrid Approach

The hybrid approach starts with the same k-means algorithm used to find a server placement. The resulting cluster centroids are modified to generate an initial generation for the genetic algorithm. Cluster centroid coordinates are modified by adding a small random value within the range of -0.001 and 0.001. Starting with this initial generation has proven to give the desired results much faster than the genetic algorithm that is initialized with arbitrary coordinates. The mutation function is also altered to obtain a better result faster. The random value addition in the mutation function for the hybrid approach uses a range between -0.001 and 0.001.

CHAPTER IV

PERFORMANCE EVALUATION

4.1 Simulating Flat Rayleigh Fading

Flat Rayleigh fading is simulated to fluctuate with time per packet, i.e., the duration of a fade extends over one packet, as per the flat Rayleigh fading simulation guide [3]. The Rayleigh fading simulation is carried out with the Clarke's "sum-of sinusoids" formula. Received signal strength of -60 dB is stated to be adequate for wireless networks [17], The noise power is set to -86.02 dB in the network simulation. This implies an average signal to noise ratio (SNR) of -60-(-86.02) = 26.02 dB. Based on this, the instantaneous received fading signal is generated. The average PER is determined as follows [18]:

$$PER = 1 - (1 - BER)^n \tag{1}$$

where BER denotes the Bit Error Rate; n is the number of bits per packet, which is taken as 1024 bits. The simulation results of average PER per SNR (dB) is shown in Fig 10. The analytical Rayleigh PER represents the estimated PER for the SNR value [19]. Also, in Fig 10, the analytical AGWN is also compared to the flat Rayleigh fading results using the BER computation in AWGN as follows [19]:

$$P_b = (1/2)erfc(\sqrt{E_b/N_o}) \tag{2}$$

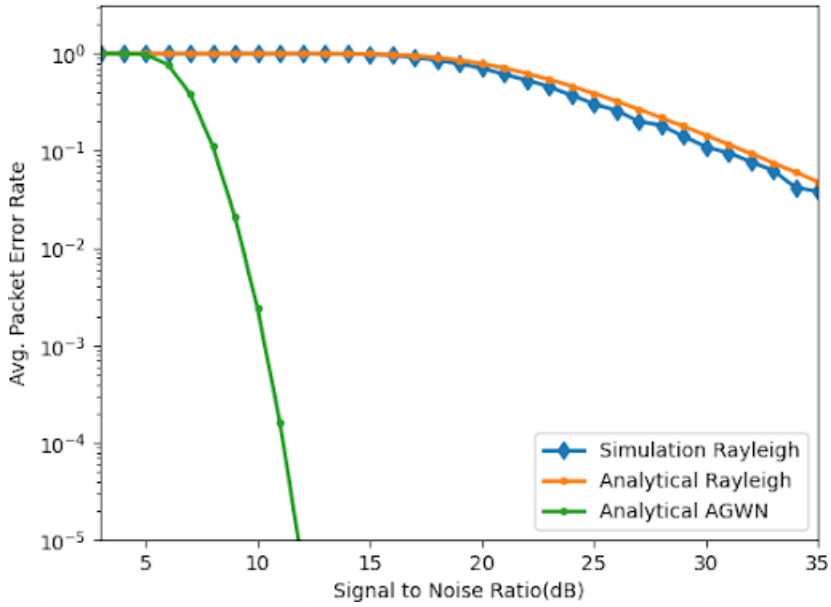

Figure 10: Average PER vs. SNR

An example fluctuating signal to noise ratio over time due to flat Rayleigh Fading and its resulting packet error rate over time graph can be seen in Appendix A.

4.2 Datasets

MQTTset [20] is a dataset specifically generated for training ML algorithms that can detect and classify cyber-attacks over the MQTT network. Labels of the dataset are; Bruteforce, Denial of Service (DoS), Flood, Malformed, Slow DoS against Internet of Things Environments (SlowITe), and Legitimate. The 1.4 million random instances were used in this paper for training and testing ML algorithms. In order to distinguish IoT devices from non-IoT devices, Yourthings [21] raw packet capture (pcap) data is used. The raw packet capture data presented is processed with cicflowmeter tool to generate 81 statistical features and labeled as "IoT" and "non-IoT" following the IP address-device name mapping. 317039 instances were generated with 81 features to train and test the device identification model. In conclusion, two ML classifiers are used, one for classifying the device type, and the other for detecting Bruteforce, Denial of Service (DoS), Flood, Malformed, Slow DoS against Internet of Things Environments (SlowITe)

attacks in real-time. The dataset for [10] is collected using globally distributed Mirai honeypots. This dataset is unfortunately not publicly available. Dataset generated in [22] is used to evaluate the performance and applicability of the classifier trained with MQTTset [20]. The pcap data with both attack and benign instances, is used for obtaining the information required to compose the features. After filtering the instances, the rows with the DNS protocol made up 2% of the original data.

4.3 Performance Evaluation of Classifiers

In order to validate MQTTset, and to find the best performing ML algorithm, the Random Forest, Decision Tree, Naive Bayes, and Logistic Regression classifiers were trained with 1,050,000 random instances from the balanced MQTTset data. The training data makes up %75 of the data used, leaving the remaining 25% for testing. In the network simulation, 7% (=1,400,000 instances) of the MQTTset was used for testing and training in total. The accuracy, precision, recall, F1-score, training time and testing time of the classifiers are provided in Table I.

Table 1: Performance Metrics of the Classifiers

Classifier	**Accuracy**	**Precision**	**Recall**	**f1**	**Training Time(s)**	**Testing Time(s)**
Decision Tree	0.83	0.84	0.84	0.83	1.68	0.28
Random Forest	0.68	0.79	0.68	0.62	20.86	3.06
Logistic Regression	0.64	0.67	0.64	0.54	1.27	0.35
Naive Bayes	0.65	0.74	0.65	0.58	112.65	0.61

The Decision Tree classifier is shown to have the highest accuracy of 83%, and the second shortest training time. Therefore, one may conclude that the Decision

Tree Classifier is the best option for anomaly detection. Generated data with cicflowmeter [23] tool from YourThings raw packet captures shows great accuracy results with the Decision Tree classifier. The accuracy and F1-score in weighted averages of identifying IoT devices is 99.99%. However, the macro average of the F1-score is 93%. The lower accuracy metrics in macro average F1-score can be explained by the imbalance of class instances. Extracted pcap data for identifying device type shows lower numbers of non-IoT device instances.

4.4 Distributed Server's Effect on Accuracy

The effect of the number of MQTT servers on blacklisting accuracy in a flat Rayleigh fading channel is shown in Fig 11. (indicated as DibLIoT). The usage of five servers is observed to result in an accuracy close to that achieved in a nonfading environment. Increasing the number of servers from one to five significantly improves the blacklisting accuracy from 42% up to 82%. The 82% accuracy achieved is less than the 83% accuracy observed in the simulation without flat fading. The increase in server count after 5 shown to increase with diminishing returns. Also introducing more distributed servers increases the complexity by adding new MQTT connection and communication processes for the low power IoT devices. In instances where fading causes packet loss for a MQTT server, the lost blacklisting information is communicated through hashgraph network by the servers that successfully receive the packet. To evaluate the anomaly detection performance in DiBLIoT, the same DNS feature composition techniques and the ML classifier in [10] are used. To simulate the same procedure, data from [22] is used when simulating the distributed server blacklisting accuracy over a flat Rayleigh fading channel. Feature composition procedure utilizes external IP addresses to acquire autonomous system (AS) diversity information and geographical statistics from a publicly available database. The dataset used in comparison only provided local IP addresses. Therefore, composing these features was not possible for this case. However, the paper shows that these features scored lower

than 0.1 in feature importance. The other features were successfully composed for evaluation. The blacklisting accuracy comparison over Rayleigh Fading channel in Fig 11 shows that the distributed blacklisting protocol presented achieves higher accuracies ranging between 47.2%-97.6% in DoS and benign labels versus 47.4%-90.7% accuracy with DNS feature composition RF classifier (indicated as DNS in Fig.11).

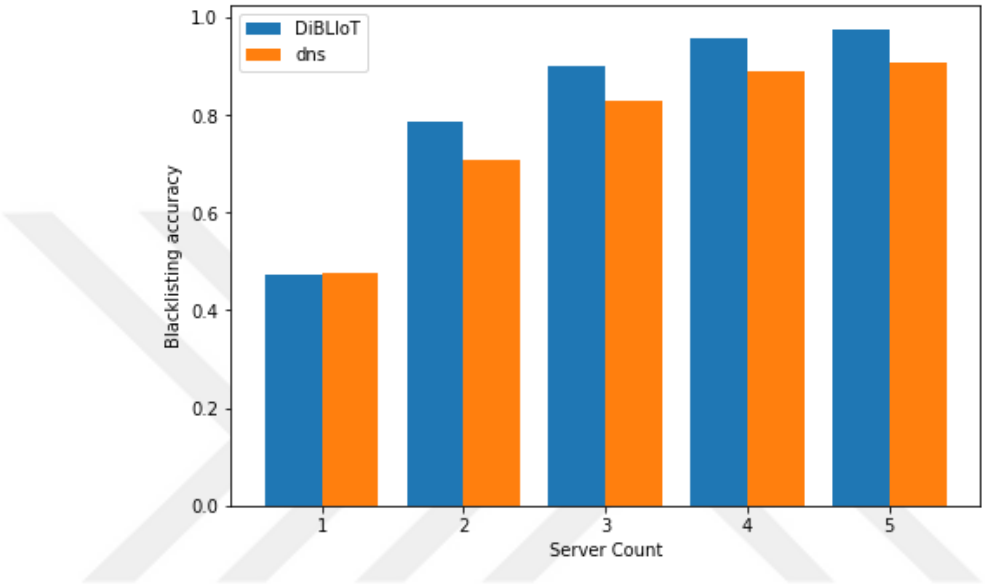

Figure 11: Accuracy rate for the server count

4.5 Comparison of Server Location Optimization Algorithms

The mean theoretical PER performances of the optimization algorithms are shown in Fig 12. The execution time for all the algorithms is 700 seconds. The genetic algorithm, k-means and hybrid approach yields close results, and the simulated annealing algorithm is shown to perform the worst. For each instance, our hybrid approach is shown to yield the best optimization score. The k-means algorithm is a close second, which converged in 0.5 seconds, making it the fastest algorithm . The combination of k-means and the genetic algorithm in our hybrid approach has consistently shown to achieve better results than all the other algorithms used, and also shown to yield more optimal results faster than the genetic algorithm and simulated annealing.

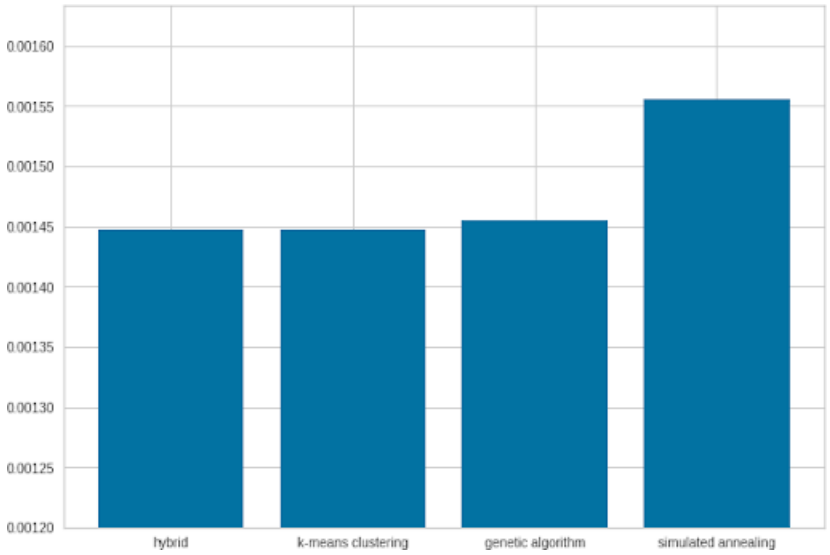

Figure 12: Accuracy rate for the server count

CHAPTER V

CONCLUSION

This paper presents a novel blacklisting protocol, where distributed MQTT servers jointly decide in real-time whether to blacklist a device by performing anomaly detection and using the hashgraph consensus algorithm. The IoT devices are classified by a device fingerprinting method using various ML algorithms that are trained with real-time packet capture data. The proposed blacklisting protocol is shown to mitigate the effects of a Rayleigh fading environment, where the number of MQTT servers provide diversity and increase the blacklisting accuracy from 42% for one server to 82% with five servers. For this simulation with multiple servers, a novel solution for reducing the PER in a Rayleigh fading environment by searching optimal distributed server locations with a hybrid approach is provided. The hybrid approach has shown to outperform k-means, genetic algorithm and the simulated annealing in a time constraint. Also, the MQTTset trained RF classifier achieves higher accuracies for the distributed network setup in comparison to related work. The proposed protocol is particularly suitable in mitigating the effects of multipath fading in an IIoT setting.

APPENDIX A

FLAT RAYLEIGH FADING GRAPH

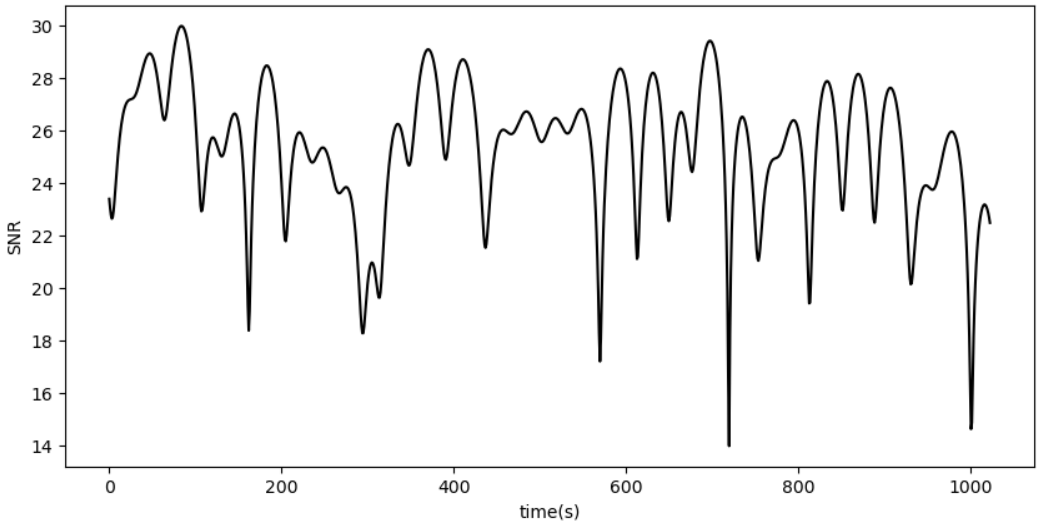

Figure 13: SNR over time

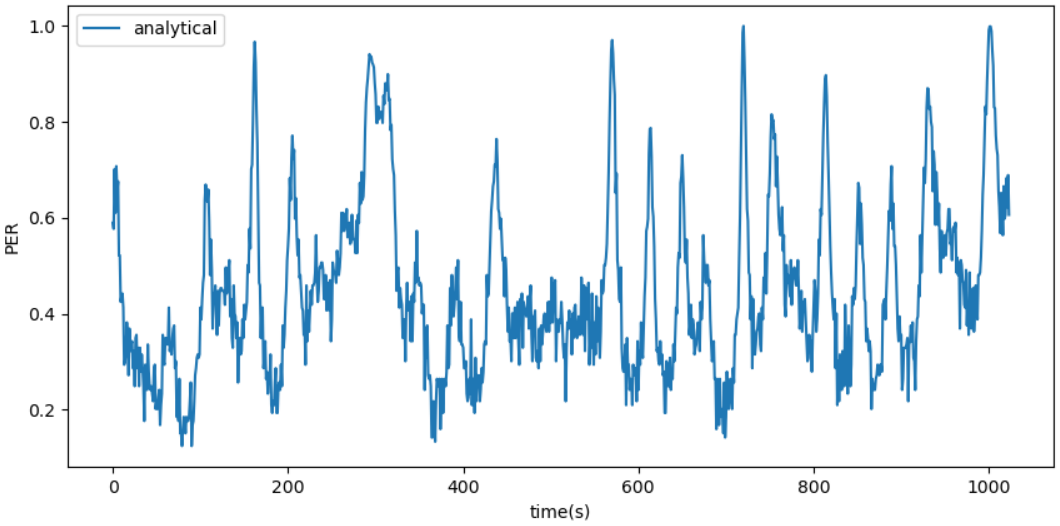

Figure 14: PER over time

Bibliography

[1] M. Tamizan, "Latency issues in internet of things: A review of literature and solution," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, pp. 83–91, 06 2020.

[2] "Oasis message queuing telemetry transport (mqtt) technical committee." https://www.oasis-open.org/committees/mqtt/charter.phpitem-2 (accessed: July 2021).

[3] "Multipath fading." https://pysdr.org/content/multipath$_f ading.html(accessed:July$2021).

[4] L. Baird, "Hashgraph consensus: Fair, fast, byzantine fault tolerance," tech. rep., Swirlds, May 2016.

[5] "What is hashgraph consensus?." https://hedera.com/learning/what-is-hashgraph-consensus (accessed: December 2021).

[6] "Genetic algorithm." https://www.geeksforgeeks.org/genetic-algorithms/ (accessed: December 2021).

[7] "Simulated annealing." https://machinelearningmastery.com/simulated-annealing-from-scratch-in-python/ (accessed: December 2021).

[8] "K-means." https://www.geeksforgeeks.org/k-means-clustering-introduction/ (accessed: December 2021).

[9] T. Gu and P. Mohapatra, "Bf-iot: Securing the iot networks via fingerprinting-based device authentication," pp. 254–262, 2018.

[10] O. P. Dwyer, A. K. Marnerides, V. Giotsas, and T. Mursch, "Profiling iot-based botnet traffic using dns," pp. 1–6, 2019.

[11] Y.-W. Chen, J.-P. Sheu, Y.-C. Kuo, and N. Van Cuong, "Design and implementation of iot ddos attacks detection system based on machine learning," pp. 122–127, 2020.

[12] A. AlAbdullatif, K. AlAjaji, N. Al-Serhani, R. Zagrouba, and M. AlDossary, "Improving an identity authentication management protocol in iiot," pp. 1–6, 05 2019.

[13] K. Skoufas, E. Spyrou, and D. Mitrakos, "Identifying ddos attacks from fluctuations in wireless traffic in an intelligent iot road network," pp. 451–456, 06 2020.

[14] H. B. Patel and D. C. Jinwala, "Blackhole detection in 6lowpan based internet of things: An anomaly based approach," pp. 947–954, 2019.

[15] M. R. Shahid, G. Blanc, Z. Zhang, and H. Debar, "Anomalous communications detection in iot networks using sparse autoencoders," *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*, Sep 2019.

[16] "Hedera consensus service." https://hedera.com/consensus-service (accessed: July 2021).

[17] "Wi-fi signal strength: What is a good signal and how do you measure it." https://eyenetworks.no/en/wifi-signal-strength/ (accessed: July 2021).

[18] "Packet error rate." https://www.sciencedirect.com/topics/computer-science/packet-error-rate (accessed: July 2021).

[19] "Ber for bpsk in rayleigh channel." http://www.dsplog.com/2008/08/10/ber-bpsk-rayleigh-channel/ (accessed: July 2021).

[20] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "Mqttset, a new dataset for machine learning techniques on mqtt," *Sensors*, vol. 20, no. 22, 2020.

[21] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "Sok: Security evaluation of home-based iot deployments," pp. 1362–1380, 2019.

[22] A. Hamza, H. H. Gharakheili, T. A. Benson, and V. Sivaraman, "Detecting volumetric attacks on lot devices via sdn-based monitoring of mud activity," p. 36–48, 2019.

[23] "Cicflowmeter." https://www.unb.ca/cic/research/applications.html (accessed: July 2021).

VITA

Ozan Tarlan

EDUCATION

Bachelor of Computer Science (June 2020) Özyeğin University, Istanbul, Turkey

Master's of Computer Science (Ongoing) Özyeğin University, Istanbul, Turkey