Republic of Iraq
Ministry of Higher Education and
Scientific Research
University of Babylon
Information Technology College
Information Networks Department

A Developed Detection Method of DDoS Attacks in Cryptocurrency Network Services

A Dissertation Submitted to the Council of the College of Information Technology for Postgraduate Studies of the University of Babylon in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Information Technology / Information Networks

By

Amenah Abdulabbas Abdulameer Mehdi

Supervised by

Prof. Dr. Wesam S. Bhaya

2023 A.D. **1445 A.H.**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

﴿وَنُرِيدُ أَنْ نَمُنَّ عَلَى الَّذِينَ اسْتُضْعِفُوا فِي الْأَرْضِ وَنَجْعَلَهُمْ أَئِمَّةً وَنَجْعَلَهُمُ الْوَارِثِينَ﴾

صدق الله العلي العظيم

سورة القصص

الآية ٥

Supervisor Certification

I certify that the dissertation entitled (**A Developed Detection Method of DDoS Attacks in Cryptocurrency Network Services**) was prepared under my supervision at the department of Information Networks/ College of Information Technology/ University of Babylon as partial fulfillment of the requirements of the degree of Doctor of Philosophy in Information Technology/Information Networks.

Signature:
Supervisor Name: **Prof.Dr. Wesam S. Bhaya**
Date: / /2023

The Head of the Department Certification

In view of the available recommendations, I forward the dissertation entitled "**A Developed Detection Method of DDoS Attacks in Cryptocurrency Network Services**" for debate by the examination committee.

Signature:
Assistant Prof. Dr. Alharith A. Abdullah
Head of Information Networks Department
Date: / /2023

Certification of the Examination Committee

We hereby certify that we have studied the dissertation entitled (**A Developed Detection Method of DDoS Attacks in Cryptocurrency Network Services**) presented by the student (**Amenah Abdulabbas Abdulameer Mehdi Almamoori**) and examined him, in its content and what is related
to, and that, in our opinion, it is adequate with (**Very Good**) standing as a dissertation for the degree of Doctor of Philosophy in Information Technology /Information Networks.

Signature:
Name: **Dr. Mohammed Abdullah Naser**
Title: **Professor**
Date: / / 2023
(**Chairman**)

Signature:
Name: **Dr. Mahdi Nsaif Jasim**
Title: **Professor**
Date: / / 2023
(**Member**)

Signature:
Name: **Dr. Jumana Waleed Salih**
Title: **Assistant Professor**
Date: / / 2023
(**Member**)

Signature:
Name: **Dr. Ameer Kadhim Hadi**
Title: **Assistant Professor**
Date: / / 2023
(**Member**)

Signature:
Name: **Dr. Raaid N.Alubady**
Title: **Assistant Professor**
Date: / / 2023
(**Member**)

Signature:
Name: **Dr. Wesam S. Bhaya**
Title: **Professor**
Date: / / 2023
(**Member and Supervisor**)

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature:
Name: **Dr. Wesam S. Bhaya**
Title: **Professor**
Date: / / 2023
(**Dean of Collage of Information Technology**)

Dedication

To the memory of my most incredible mother who gave me and taught me a lot, whose death was my most significant loss.

To my father's martyr, I affectionately dedicate this modest work.

Amenah

Acknowledgments

All praise be to ALLAH the Almighty, who enable me to complete this task successfully, and our utmost respect be to His last Prophet Mohammed and his holy family.

My deepest gratitude is to my supervisor, for invaluable guidance, supervision, and untiring efforts during this work.

I would like to thank all my dear teachers in the College of Information Technology for their remarkable education and guidance during the study period.

Declaration Associated with this Dissertation

Some of the works presented in this dissertation that have been published are listed below.

1. **Hybrid Deep Learning Approach Utilizing RNN and LSTM for the Detection of DDoS Attacks Within the Bitcoin Ecosystem.**
   Ingénierie des Systèmes d'Information, 2023.
2. **Survey on cryptocurrency security attacks and detection mechanisms**
   Bulletin of Electrical Engineering and Informatics,2023

Abstract

The growing interest in blockchain technology, functioning as an unchangeable ledger facilitating distributed transactions, is remarkable. Nevertheless, Blockchain security is susceptible to a variety of risks, notably distributed denial of service (DDoS) attacks, which are progressively focusing on cryptocurrency network services. To address this, deep learning approaches have emerged as a potent solution to intricate challenges in information science.

The dissertation suggests the integration of these approaches into hybrid model to address complex cybersecurity challenges. Initially, A hybrid deep learning model used for binary classification, which combines Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) algorithms, was introduced for detecting DDoS attacks on cryptocurrency network services.

Subsequently, a proposal is made to build a Graphical User Interface (GUI) for monitoring network traffic and identifying abnormal patterns or behaviors. This plan involves implementing measures to prevent intrusions and automatically respond to suspicious activities. Two models are established, one utilizing the Agglomerative Hierarchical Clustering (AHC) method and the other employing the Gaussian Mixture Model (GMM) for identifying DDOS attacks on cryptocurrency network services.

The suggested model is assessed using the real dataset that represents data associated with Bitcoin services attacked by DDoS. The suggested model outperforms standard deep learning implementations, achieving an impressive level of accuracy. When evaluated with the Mt.Gox dataset, the model demonstrated an accuracy of 95.84%. Furthermore, it underwent validation using widely recognized datasets, specifically CIC-IDS2017 and CSE-CIC-IDS2018, achieving accuracies of 95.40% and

99.77% respectively. Therefore, this dissertation introduces a promising strategy for mitigating DDoS attacks within the cryptocurrency ecosystem.

Table of Contents

Table of Tables

Table of Figures

Table of Abbreviations

Abbreviations	Full Name
AHC	Agglomerative Hierarchical Clustering
AML	Anti-Money Laundering
AWS	Amazon Web Services
BGP	Border Gateway Protocol Hijacking
BTC	Bitcoin
CDNs	Content Delivery Networks
CICIDS2017	Canadian Institute for Cybersecurity Intrusion Detection System 2017
CSE-CIC-IDS2018	Communications Security Establishment and Canadian Institute for Cybersecurity Intrusion Detection System 2018
CTGAN	Conditional Tabular Generative Adversarial Network
DAG	Distributed Acyclic Graph
DApps	Decentralized applications
DDoS	Distributed Denial of Service
DeFi	Decentralized Finance
DHC	Divisive Hierarchical Clustering
DLT	Distributed Ledger Technology
DNS	Domain Name System
DPoS	Delegated Proof of Stake
EHRs	Electronic Health Records
ETH	Ethereum
EVS	Electronic Voting Systems
FN	False Negative
FP	False Positive
GMM	Gaussian Mixture Model
GUI	Graphical User Interface
HYIP	High Yield Investment Program
ICMP	Internet Control Message Protocol
ICOs	Initial Coin Offerings
IDS	Intrusion Detection Systems
IoT	Internet of Things
ISE	International Securities Exchange

Abbreviations	Full Name
ISP	Internet Service Provider
KYC	know-Your-Customer
LSTM	Long Short Term Memory
LTC	Litecoin
ML	Machine Learning
MLP	Multi-Layer Perceptron
NTP	Network Time Protocol
P&D	Pump &Dump attack
PoA	Proof of Authority
PoD	Ping of Death attacks
PoS	Proof of Stake
PoW	Proof of Work
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
STOs	Security Token Offerings
tanh	hyperbolic tangent
TN	True Negative
TP	True Positive
UDP	User Datagram Protocol
UI	User Interface
XRP	Ripple

Chapter One
Introduction

1.1 Introduction

Cryptocurrencies are digital currencies that require blockchain technology to function. Cryptocurrency exchanges sprung up to satisfy the demands of investors looking to benefit from the unexpected spike in the value of digital currencies. These cryptocurrency exchanges use online platforms to authorize their clients to store, buy and sell cryptocurrencies. Cryptocurrency examples include Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC), Ripple (XRP) and Libra[7][8] . Moreover, blockchain is a data structure and DLT that records transactions using an immutable cryptographic signature known as a hash. Distributed Ledger Technology (DLT) participants who manage the decentralized database are widely perceived. Among the main functions of blockchain is to store cryptocurrency transaction data. Blockchain needs a peer-to-peer network that ensures that each user has the information of all the transactions made in that grid. It is composed of five key concepts: database, block, hash, miner, transaction and consensus mechanism[9]. Recently, as the blockchain ecosystem matures and new use cases emerge, organizations across all industries face a complex and potentially contentious set of challenges, as well as novel dependencies. Bitcoin continues to have the most trading volume of any cryptocurrency.

Several vulnerabilities and attacks, such as DDoS attacks, are investigated as the Bitcoin market grows[10]. DDoS is a network threat that immediately compromises the resources of its victims. A DDoS attack floods the target with bogus requests directed at its network infrastructure by the attacker. As a result of this attack, the website becomes difficult to access for the intended users[11]. Perhaps a critical issue in DDoS attack detection techniques is the lack of consideration of the limitations of small datasets. A comprehensive understanding of the

application of new technologies is necessary to improve efficiency. To the best of our knowledge, researchers concentrate on other DDoS detection techniques or included some blockchain-based research for a specific domain by using the same datasets available in research engines.

Machine learning algorithms are used for a variety of tasks, including detecting malicious traffic and addressing network problems. Building a model to reflect complex interactions between input and target variables is one of the main goals of machine learning. It serves as a framework for reliable behavior in the network domain. Supervised, unsupervised, and reinforcement learning are the three main categories of machine learning algorithms[12].

1.2 The Related Works

Many researchers propose different methodologies that focus on the detection of DDoS attacks. Nevertheless, the lack of consideration of the size of the dataset, the limitation of complexity, and real-time problems are critical problems in DDoS detection methodologies. These related works are as follows:

1. Vasek et al. (2014) conducted an empirical investigation into DDoS attacks in the Bitcoin ecosystem. Mt. Gox was used as an illustration of a DDoS case study on currency exchange. Between May 2011 and October 2013, the authors analyzed and collected several posts mentioning 'DDoS' on the common Bitcoin forum named bitcointalk.org. By using the precision, recall accuracy, and confusion matrix measures, they developed a simple word based classifier to detect DDoS attack threads. Precision was 54%, recall was 74% and accuracy was 75%. They also discovered that DDoS attacks had been launched against 7.4% of Bitcoin related services[13].

2. On this basis, Feder et al. (2018) studied the influence of DDoS on cryptocurrency exchanges, specifically the Mt. Gox exchange. Mt. Gox had been frequently targeted by DDoS attacks and was forced to shut down as a critical breach that resulted in stolen funds. The authors built a series of regressions to assess the effects of shocks on transaction volume. They employed a regression model, and the results were powerful for different options. They demonstrated a reduction in high-volume trades as a DDoS attack. They also declared that other types of security breaches had an identical influence [14].

3. On the basis of a new transaction history summarization, Toyoda et al. (2018) suggested the implementation of a multi-class categorization system for identifying services associated with Bitcoin addresses. Adopting a multi-class categorical scheme offers several benefits, notably in enhancing fraud detection capabilities. The authors proposed two methods for retrieving transaction history: address- and owner-based methods. They used random forest as a detection algorithm and achieved an accuracy of 72% using the owner-based scheme, and 70% using the address-based scheme.[15].

4. Sergy (2018) studied the size of the effect of a DDoS attack on a cryptocurrency exchange using the Bitfinex business model. This exchange gets money by charging a fee for every trade that occurs on the platform. A large amount of money can be forgotten if a DDoS attack is successful and the exchange's service is denied. The study employed statistical techniques to assess the effects of 18 DDoS attacks on the Bitfinex exchange. By using the proposed prediction model for the number of trades for the event study with diverse estimation and event windows, no substantial effect on the cryptocurrency exchange was found. That is, trading on the exchange was not reduced significantly in the days following a DDoS attack [16].

5. Abhishta et al. (2019) examined the effects of DDoS attacks on Bitfinex in 2016, 2017 and 2018. They forecasted the average Bitcoin volume traded on the exchange using an additive model. To find the better estimation model, they compared the goodness of fit of two models, linear ordinary least squares and quadratic ordinary least squares. According to research, in most cases, this cryptocurrency exchange can recover from the effect of a DDoS attack within a single day. However, a long standing DDoS attack can considerably affect the exchange's revenues [7].

6. Beak et al. (2019) examined and identified DDoS attacks under the premise that there is an association between Bitcoin's service and network-level data. The researchers collected and scrutinized data from actual DDoS attacks targeting services related to Bitcoin. They put forth an approach for delineating the data retrievable from the Bitcoin network, along with the statistical data pertaining to blocks. Consequently, their focus was on detecting DDoS attacks at the Bitcoin service level, employing deep learning methodologies like MLP for detection and principal component analysis for feature extraction. [10].

7. Yoon et al. (2019) presented CTGAN as a solution to the problem of generating realistic synthetic tabular data, which is challenging due to the structural constraints and dependencies present in tabular datasets. CTGAN leverages the power of GANs to learn and designed to grasp the fundamental data distribution, enabling it to produce samples that closely mirror the original data. It incorporates a conditional framework, enabling the generation of samples based on specific attribute values. This makes it particularly useful for tasks that require generating synthetic data with specific properties or characteristics. The algorithm combines a discriminator network and a generator network in a competitive setting, where the generator aims to

produce realistic samples, while the discriminator tries to recognize between real and generated samples. The authors evaluated CTGAN on various real-world tabular datasets, including credit card transactions and census income data. The results showed that CTGAN was able to generate synthetic samples that closely matched the statistical properties of the real data. The authors also compared CTGAN with other baseline methods and demonstrated its superiority in terms of generating high-quality synthetic tabular data [17]. Table 1.1 displays the methodologies employed by researchers along with the outcomes they've attained in the effort to thwart DDoS attacks.

Table 1.1 DDoS attack on Blockchain Ecosystem and detection methods

No.	References	The target environment	Detection methods	Results
1	[18]	Ethereum Node	Automated tool support and manual methods for evaluate the system's solidity codebase's security attributes.	On the consortium blockchain installed in Microsoft Azure, the effects of the flooding and bandwidth exhaustion attacks were also identified.
2	[19]	Lisk Blockchain	Implementing two tools as countermeasure for detection	Only the delegated proof of stake consensus besed currency is affected by the flaw.
3	[20]	Bitcoin Ecosystem	K-means and Support Vector Machine	The dataset contains few attack instances, and the false positive rate is exceptionally high.
4	[21]	Bitcoin Platform	presents a parallel anti-DDoS chain design philosophy and a distributed anti-d chain detection system for	It is not a realistic assumption to use the Hadoop platform nodes as blocks in a blockchain for tests.

No.	References	The target environment	Detection methods	Results
			virtual reality based on hybrid ensemble learning.	
5	[7]	Bitfinex exchange	a DDoS attack event analysis to see how it affected the volume of trades on the Bitfinex exchange as well as a DDoS attack hourly traded data evaluation	The primary outcome is that the exchange recovers from the adverse effects of the DDoS attack the same day.
6	[10]	Mining pool and exchange	a DDoS attack detection MLP model based on PCA and deep learning	DDoS attacks were detected with a 50% accuracy, whereas regular block data were identified with a 70% accuracy.
7	[14]	Mt.Gox exchange	An econometric analysis of Bitcoin transaction data between 2011 and 2013 to evaluate the effect of the DDoS attack and associated shocks on the Mt.Gox exchange	A DDoS attack on the exchanges resulted in fewer major transactions.
8	[13]	Exchange and mining pool	An empirical study of DDoS attack effects on Bitcoin mining pools and exchange	DDoS attacks on other cryptocurrencies may present a worthwhile area for research. The accuracy of DDoS attack detection was approximately 75%.

Cryptocurrency attacks can be detected in various approaches according to the type of attack, its severity, and its impact on the labor market. Table 1.2 highlights most types of attacks against some digital currencies, the detection methods used, the methods applied to evaluate

each model's effectiveness, and the results obtained. This table shows the most common and influential attacks on the cryptocurrency ecosystem.

These techniques can be employed to create a more secure level for cryptocurrency networks with the possibility to detect and prove the specific types of malicious transactions. Importantly, the most common methods used for detecting attacks are machine learning and deep learning. Arguably, machine learning is used in artificial intelligence to identify the most effective solutions to complex issues in information science.

Table 1.2: Methods for detection of cryptocurrency attacks

	Ref.	Published year	Attack type	Cryptocurrency	Detection or Prevention Methods	Results and performance measurements
1	[22]	2020	51% attack and DAO attack	Etherum Classic Network (ETC)	RNNs as a neural encoder-decoder model	Recurrent autoencoder (RAE) model that effective detect the publicly reported attack
2	[23]	2015	51% attack	Bitcoin	Continuous-time Markov chains (CTMCs)	Results obtained are applicable for each state of the Bitcoin network
3	[10]	2019	DDoS attack	Bitcoin	Multi-layer Perceptron (MLP)	DDoS attacks were detected with a 50% accuracy, whereas regular block data were identified with a 70% accuracy.
4	[13]	2014	DDoS attack	Bitcoin	Word-based classifier	Using a confusion matrix, The DDoS attack detection accuracy was roughly 75%.
5	[24]	2019	Ransom ware	Bitcoin	Bayesian Belief Betwork(BBN)	The accuracy of Ransomware attack detection was approximately 97.5%.

	Ref.	Published year	Attack type	Cryptocurrency	Detection or Prevention Methods	Results and performance measurements
6	[25]	2019	P&D	Bitcoin	Random Forest	Using LASSO regularized GML and balanced random forests, the probability of a currency being pumped with an area under the curve of over 90% was predicted.
7	[26]	2019	P&D	The model was applied to the full-time series of 172 coins	Extreme Gradient Boosting	The result was as follows: 99.5% AUC, 99.7% specificity, and 85.5% sensitivity, using the area under the curve .
8	[27]	2019	Eclipse attack	Ethereum	Random Forest	The precision rate is approximately 72%, and the recall rate is approximately 93%
9	[28]	2020	Eclipse attack	Bitcoin	Python-Flask web framework and Flask's default webserver	The Gossip-based protocol provides multiple benefits while introducing a significantly improved detection time and low overheads, using Amazon AWS
10	[29]	2019	Cryptojacking attack	JSECoin and Monero	SVM classification model	97%TPR and 1.1%FPR
11	[30]	2022	Ransomware	Bitcoin	Rule-Based algorithms	Accuracy of approximately 96.01%, recall of approximately 96%, precision of approximately 95.9%, and an F-Measure of 95.6%,when metrics, accuracy, precision, sensitivity, and F-Measure are employed.

	Ref.	Published year	Attack type	Cryptocurrency	Detection or Prevention Methods	Results and performance measurements
12	[31]	2019	Cryptojacking attack	Ethereum, Monero, and Zcash	Shared Nearest Neighbour (SNN) clustering algorithm	Using KNN classifier,99.7% TPR, 46.1% FPR,99.9% Precision, and 99.7% Recall
13	[32]	2019	Cryptojacking attack	Monero	Capsule Network (CapsNet) technology	87% of the instances were detected immediately, and 99% of the instances were detected during a window of 11 seconds.
14	[33]	2021	Cryptojacking attack	Bitcoin, Monero, and Bytecoin	Random Forest	Using the Mean Square Error (MSE) 94.1% TPR, 59% FPR, 99% of AUC for the ROC and 96% of F1-score
15	[34]	2019	HYIP threat	Bitcoin	Random Forest	Accuracy of approximately 95% TPR and 4.9 FPR
16	[35]	2018	HYIP threat	Bitcoin	Random Forest	Accuracy of approximately 97.9%, 96.8% TPR, 96.9% recall, and 97.9% specificity
17	[34]	2017	HYIP threat	Bitcoin	Random Forest	Accuracy of approximately 83% TPR and 4.4% FPR
18	[36]	2018	HYIP threat	Ethereum	Extreme Gradient Boosting (XGBoost)	94% precision, 81% recall, and 86% F-score.
19	[37]	2019	HYIP threat	Ethereum	Random Forest	Accuracy of approximately 95% precision and 69% recall

1.3 The Problem Statement

DDoS attacks pose a significant threat to cryptocurrency network services. These attacks can disrupt the availability and stability of cryptocurrency platforms, causing financial losses and undermining user trust. The following are some of the significant shortcomings in the current literature that gave rise to the motivation for the proposed dissertation:

- Lack of high accuracy combined with reasonable time consumption and scalability.
- Unfortunately, no system can be completely immune to DDoS attacks,
- Lack of use of extensive datasets.
- Lack of novel methods for detecting DDoS attacks.

1.4 The Dissertation Aims

This work aims to detect and prevent DDoS attacks that cause large-scale impacts on the cryptocurrency ecosystem. The main aim is to enhance the Cryptocurrency Network Services security of blockchain by accurately and rapidly identifying and mitigating DDoS attacks in real-time.

1.5 The Dissertation Objectives

The objectives of this dissertation can be outlined as follows:

- Improving detection accuracy, handling complex attack vectors, and minimizing false positives.
- Proposing a Hybrid Deep Learning model for detecting DDoS attacks at the service level of Cryptocurrency Blockchain Network Services.

- Introducing a best Feature Selection Model to improve detection performance and reduce overfitting in the proposed deep learning model.
- Introducing a GUI designed for monitoring network traffic and detecting suspicious behaviors or activities on cryptocurrency network services .

1.6 The Dissertation Contributions

The contribution of this dissertation can be described as follows:

1- Introducing the Feature Selection Model using a machine learning algorithm, Random Forest, to get best feature selection.
2- Proposal of a Hybrid Deep Learning Model using recurrent neural network (RNN) and long short-term memory (LSTM) for DDoS attack detection in cryptocurrency networks.
3- Proposed Extracting Attack Dataset Model using the trained neural network model.
4- Introducing Clustering Model using Agglomerative Clustering Model or Gaussian mixture Clustering Model.
5- Creating two GUI models, the first one pertaining to the AHC model and the second to the GMM model for the detection and prevention of DDoS attacks.
6- Achieve acceptable accuracy for the detection of DDoS attacks.

1.7 The Outlines

This dissertation contains five chapters organized as follows:

Chapter Two: The Theoretical Background

This chapter reviews both the Blockchain network and DDoS attacks, the challenges faced by cryptocurrency networks. The deep learning algorithms used to detect DDoS attacks. In addition, the clustering approach will be presented in the proposed system. In addition, cyber

Attacks in cryptocurrencies In the last part of the chapter, the evaluation of the ML algorithms will be explained.

Chapter Three: The Proposed Model Design

The proposed system details have been discussed in this chapter, including the proposed system and the approaches used to identify DDoS attacks utilizing hybrid deep learning techniques.

Chapter Four: The Results Discussion and Analysis

The results of the proposed system for DDoS attack detection are discussed in this chapter. The evaluation results are presented; these results are depended on accuracy, precision, recall, F-score, and test time.

Chapter Five: The Conclusions and Suggested Future Works

The conclusion discussed DDoS attack detection in the cryptocurrency network services and suggested additional work to develop the proposed system.

Chapter Two

The Theoretical Background

2.1 Introduction

This chapter presents a momentary description of Blockchain networks, types, and generations. This chapter also identifies the biggest challenge facing this type of network which represents the security of these networks. The types of attacks that suffer from these networks and the methods and techniques that have been proposed to detect these attacks. It also discusses the types of DDoS attacks and how be launched. Then it focuses on the machine learning and deep learning approaches such as RNN and LSTM and some algorithms of clustering that are employed in the level of attack categorizing. This chapter also displays the main algorithms that are proposed for the detection of DDoS attacks and for building the GUI models. It also discusses the datasets that used in the proposal model and their features. This chapter also displays the performance metrics that are employed for this dissertation and some of the important aspects.

2.2 Blockchain Networks

The phrase "blockchain networks" can be comprehended on two levels: the "blockchains," which refer to a framework for immutable data structure, and the "blockchain networks," which specify the methods for data deployment and maintenance. The two terms are regarded as the two biggest innovations in blockchain technology[38].Since being first proposed in 2008 by Satoshi Nakamoto as the foundation of the cryptocurrency Bitcoin[39], blockchain technology has advanced rapidly by allowing trusted transactions between a number of untrusted network members. Blockchain technology is currently finding applications in a diverse range of computing and commercial fields, including supply chains, cloud computing, finance, the IoT, and many others[40]Blockchain technologies provide a secure distributed ledger that is collectively administered by a peer-to-peer network, enabling

distrusting parties to collaborate without requiring the necessity for a reliable third party. However, The core structure of blockchain technology encompasses six layers: the application layer, the contract layer, the incentive layer, the consensus layer, the network layer, and the data layer. These layers are represented in Figure 2.1 and are arranged from top to bottom[1].

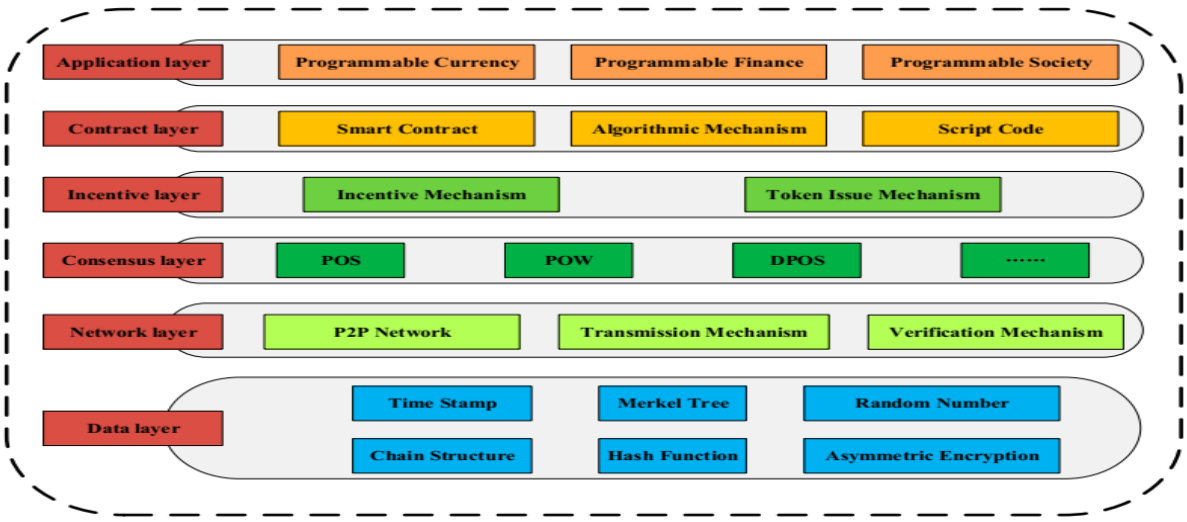


Figure 2.1: Six-layer structure of blockchain[35]

- **Data layer:** This layer amalgamates data into blocks and is responsible for functions such as hash functions, timestamps, cryptographic encryption, and the structure of the blockchain. [41].
- **Network layer:** Data verification, data transmission, and P2P networking mechanisms are all contained inside this layer. As a result of every node within the blockchain network possesses equal rights and responsibilities. Additionally, The network's communication and interaction are based on a flat architecture design. Direct communication between each node of the network allows for the transfer of data and the confirmation of new blocks. New blocks can only be added to the main chain after being verified by more than 51% of the network's users[42].

- **Consensus layer:** The consensus mechanism and algorithm are included in the consensus layer. Nowadays, there are four commonly employed consensus mechanisms: PoW , PoS, PoA, and DPoS.
- **Incentive layer:** Blockchain technology has a unique economic incentive and token distribution scheme designed to aid the nodes in upholding the network's integrity [43].
- **Contract layer:** The blockchain features a programmable attribute that enables the incorporation of scripts, algorithms, and smart contracts into each block. The blockchain system can automatically execute contract content within restrictions smart contracts, without further manual involvement. This layer significantly broadens the application possibilities for blockchain, making it one of the approaches to decrease the expense associated with obtaining credit [44].
- **Application layer:** Blockchain technology has gradually been used in aspects other than currency and finance, such as energy, network security, the Internet of Things, medical treatments, legal notarization, and copyright authentication, we're on the brink of entering the era of Blockchain 3.0, also known as the programmable society.

Originally, the blockchain is a form of DLT also it considers a secured list of records that are accessible on computing devices. Records are immutable, publicly verifiable, and sequentially generated and recognized as blocks. It is a distributed ledger that may be accessed by multiple nodes for record keeping. It consists of a network of connected nodes. So each next block commences with the genesis block, and it stores data about the previous node[2]. Figure 2.2 depicts the flow of transaction execution in a blockchain network.

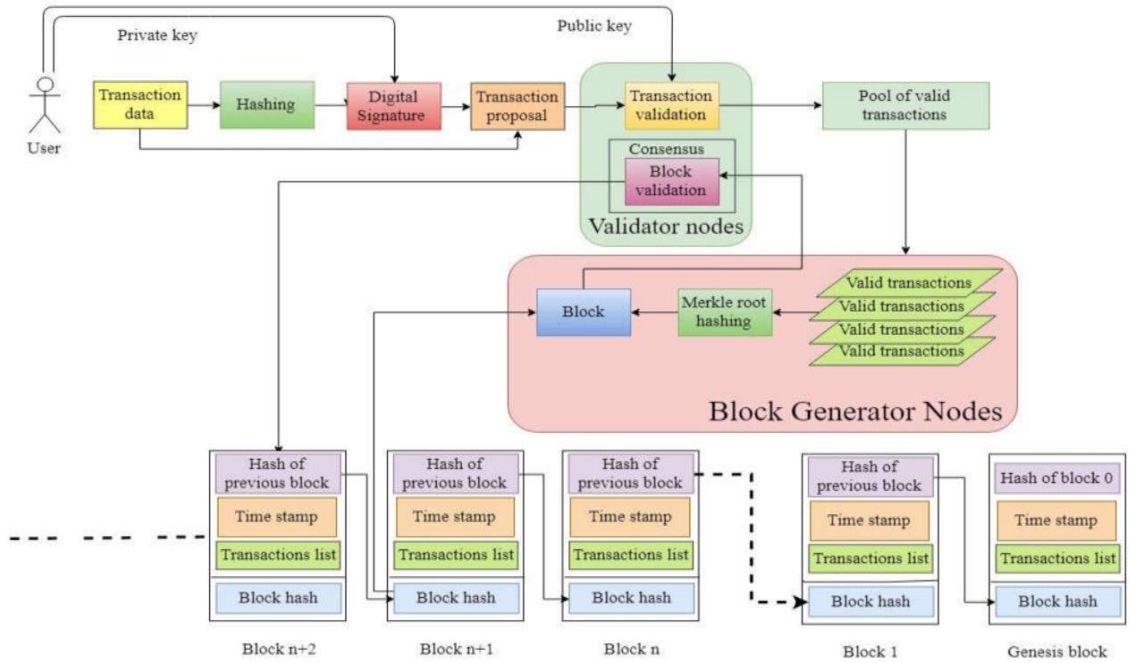


Figure 2.2 Outline of Transaction Execution Flow in Blockchain [2]

2.2.1 Blockchain Types

There are various blockchain deployment techniques that can be used depending on the application domains that are wholly dependent on the consensus mechanisms they utilize. Based on these techniques, There are generally four types of blockchains (see figure 2.3), as explained below. Based on writing access, existing blockchains can be categorized into two main groups: permissionless and permissioned blockchains. The first approach is appropriate for public networks because consensus nodes don't need to be authenticated. These open/permissionless blockchains, like those in Bitcoin and Ethereum, allow miners to join and leave the network at any moment with no constraints. Permissioned blockchains, in contrast to open blockchains, demand authorization in order to participate in consensus.

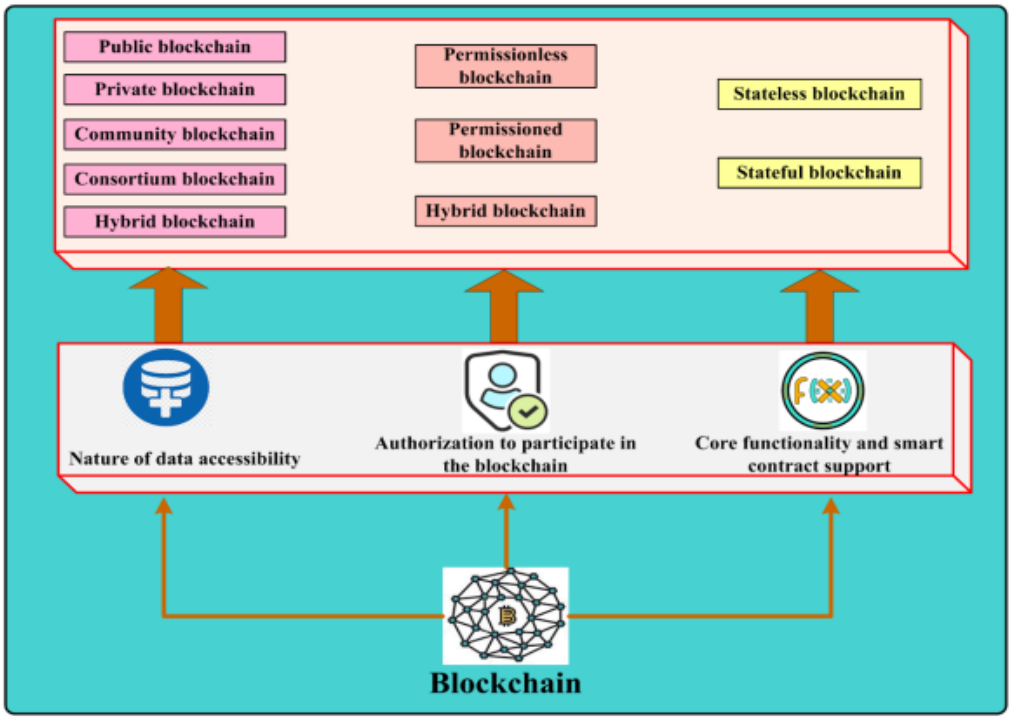


Figure 2.3: Types of blockchains[3]

1. Public Blockchain: A public blockchain, sometimes also known as the Unpermitted or Permissionless Blockchain, enables anybody to participate in the blockchain by adding and validating new blocks as well as modifying the chain's current state by storing and updating data through transactions between involved parties. This indicates that the blockchain's state, its transactions, and the data it stores are all open and transparent to everyone. This gives rise to privacy problems in specific situations where the confidentiality of such data must be maintained [45].

2. Private Blockchain: A private blockchain, also called a Permissioned Blockchain, is more constrained than its public counterpart in that only authorized and reliable companies are permitted to take part in blockchain activities. This allows a private blockchain to limit access to the chain data to trusted parties rather than the general public, which may be preferable in particular use cases. Additionally, the private chain is typically appropriate for internal business applications as well as applications that involve financial scenarios, including internal data

administration and auditing in certain businesses. In particular, in some circumstances, the organization may be able to change some private chain rules, such as restoring transaction procedures. Private chains also contain fewer nodes than public chains, which enables them to process transactions more quickly and relatively inexpensively[46].

3. Community/Consortium Blockchain: In this form of blockchain, the blockchain network is managed by a variety of companies or organizations. The peers in the chain, which are controlled by each company or enterprise, collectively record transaction data at one or more of the nodes under their control. Because the consensus process is managed by pre-selected nodes on this chain, read-write access authorization can be managed. The majority of the time, it is utilized in business-to-business situations including settlements, liquidations, and interagency transactions[47].

4. Hybrid Blockchain: - To enable inter-chain communication, a hybrid blockchain can merge permissionless and permissioned blockchains. Hybrid blockchain can combine the three types of blockchains: public, private, and community/ consortium to speed up the processing of transactions[48].

2.2.2 Generation of Blockchain

Blockchain is a novel approach that is enhancing the security and privacy of various application. Blockchain has gone through three generations, but this does not mean that each generation is better than the others. It indicates that each generation is special because of its various applications. Different "versions" of blockchain concentrate on various companies, and they are all used in various contexts. Figure 2.4 shows the four generations with their applications.

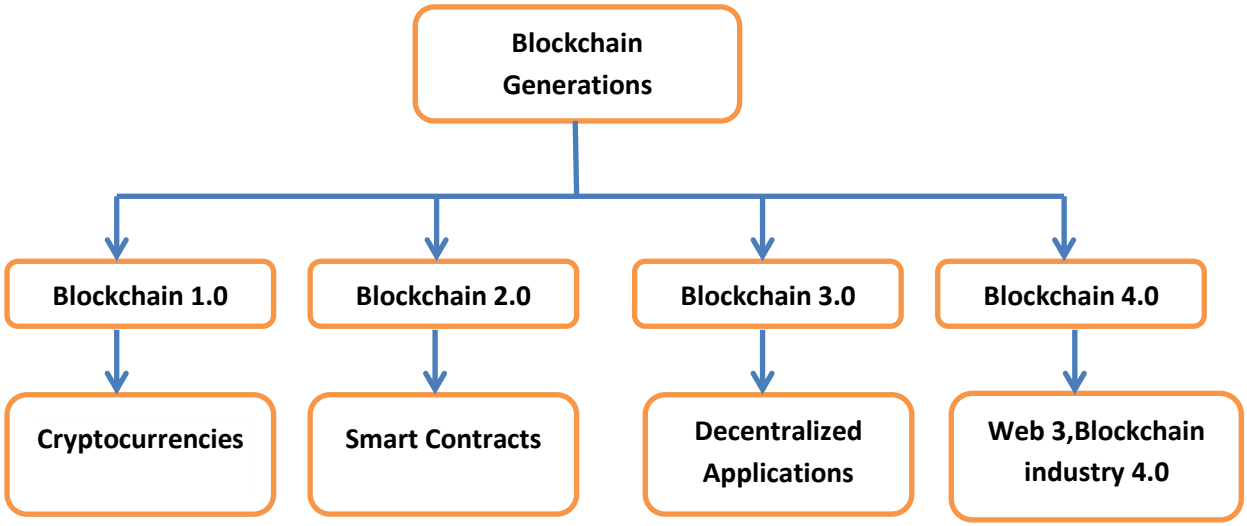


Figure 2.4: Blockchain Generations

- **Blockchain 1.0**

The fundamental utilization of blockchain in Blockchain 1.0 is for cryptocurrencies. One of the most widely used cryptocurrencies is called Bitcoin. Distributed consensus and crypto tokens were Bitcoin's first two essential blockchain components. PoW, a distributed consensus method developed by Bitcoin, determines which node will be allowed to control the chain. PoW requires nodes to expend some processing power to solve a cryptographic challenge. The first node to do so is recognized by the overall network and elevated to the position of the official validator. The actions of the new transaction validators are likewise rewarded in this system[49].

- **Blockchain 2.0**

In Blockchain 2.0, the concept of the Smart Contract was also launched, and other corporate sectors have begun to acknowledge this technology. Ethereum is one of the well-known products at this time (Ethereum Foundation, 2019). The introduction of the Smart Contract idea and a new distributed consensus is the primary distinction between version 2.0 and version 1.0. Ethereum offers completely programmable smart contracts that may be used in various enterprises, in contrast to

Bitcoin which has a very limited amount of programmability. Ethereum's distributed consensus has developed from PoW to something known as PoS. Since an algorithm's solution is no longer necessary, this has greatly decreased the amount of computational power needed[50].

- **Blockchain 3.0**

When it comes to Blockchain 3.0, applications outside of finance are the norm. A case in point is Hyperledger Fabric (Linux Foundation, 2019). It has a permissioned blockchain system with known node credentials and only permissioned nodes that can be joined to the network. For typical industrial and organizational needs, this is more appropriate. In a permissioned blockchain, the network's users are managed by a reliable identity management system. A decentralized database offers more transparency and auditability . permissioned blockchains may offer varying degrees of smart contract functionality depending on the platform's role and focus. The distributed consensus processes in permissionless blockchains can enhance system security due to their inclusive and open nature. The credentials of the nodes are known in a permissioned blockchain, and the choice of a distributed consensus is more flexible. Overall, since the involvement of nodes is expected to be tracked, a permissioned blockchain does not always require a crypto token. Nevertheless, based on the ecosystem, an application token may still be utilized for unique purposes[49].

- **Blockchain 4.0**

Real-time applications-based blockchains are processed using blockchain 4.0. This provides information about how blockchain technology may affect certain industries, such industry4.0. Industry 4.0 comprises a suite of cutting-edge manufacturing techniques that help businesses achieve their objectives more swiftly[51]. While Blockchain 3.0 primarily addressed issues with second-generation blockchains,

Blockchain 4.0 centers on leveraging blockchain for innovation. As companies across various industries increasingly adopt blockchain, it is imperative to anticipate swift advancements in this domain.

2.2.3 The Blockchain Challenges

Blockchain retrieves and commits records significantly more slowly than a conventional database. Additionally, it necessitates significantly more processing resources, whose scalability is a great concern. Furthermore, any systems interacting with the Blockchain must have the capability to work together seamlessly. The payment period should be short and adaptable, allowing for conversion into any other currency. The subsections that follow will concentrate on these difficulties. Therefore, in order to help newcomers better understand the technology, these subsections will go over the fundamentals of blockchain challenges and their alternative solutions.

- **Scalability**

Scalability refers to a system's ability to effectively respond and function even when faced with an increase in input size to meet user demands. Presently, both Ethereum and Bitcoin, which boast the highest number of users on their networks, are struggling to manage the workload. Transactions on these networks may take hours, or even days, to finalize. Ultimately, the challenges in adopting blockchain technology are leading to diminishing profitability. Therefore, this challenge must be resolved to facilitate the deployment of blockchain technologies. Scalability solutions pertain to aspects such as data storage, transaction volume, block interval duration, and data transmission. Therefore, these solutions can be categorized into four types: on-chain scalability, off-chain scalability, scalability based on distributed acyclic graphs, and scalability based on consensus mechanisms[4](see figure 2.5).

✓ **The on-chain scalability**

Necessitates altering the essential principles of the protocol, which would involve an architectural or fundamental change to Blockchain. In situations where there is a split in the community due to the emergence of factions that approve or reject the proposed update, this is referred to as a hard fork. Litecoin, DASH, and Bitcoin Cash are a few examples of the hard fork[52],[53].

✓ **The off-chain approaches:**

are referred to as second-layer scalability solutions since they use additional protocols that are constructed on top of the original Blockchain[54].This method performs transactions privately amongst the interacting partners by off-loading them from the main Blockchain. Presently, RAIDEN, Trinity Network, Plasma Cash, and Lightning Network are some of the off-chain options that are accessible[55].

✓ **Consensus Mechanism:**

The consensus algorithm's operation is enhanced to help with scalability challenges. Proof of Authority is used by VeChain, Delegated Proof of Stake is used by ARK.io, LISK, bitshares, EOS, and steemit, Federated Byzantine Agreement is used by Ripple and Stellar, and Delegated Byzantine Fault Tolerance is used by NEO. Among the best alternatives in this domain are Libra, Zilliqa, and Hyperledger[56].

✓ **Distributed Acyclic Graph-Based Scalability:**

It is yet another well-known deployment of distributed ledger technology. Transactions don't follow any particular process; they work individually and asynchronously. The system keeps track of transaction records using a topological ordering data structure. In contrast to Blockchain, DAG distributed ledger technology does not have the same scalability problems. Federated Pegged Side chains are a significant improvement in this category[57].

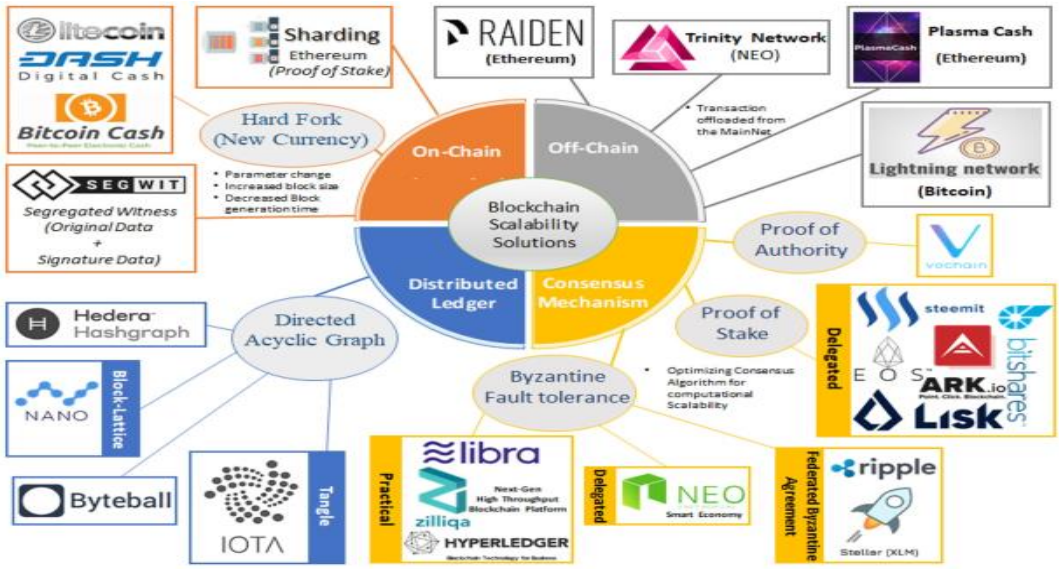


Figure 2.5: Blockchain scalability solutions[4].

- **Security and Privacy:**

A blockchain is viewed as a ledger that records data about various kinds of transactions. Everyone on the network can view the ledger because the blockchain is decentralized in a specific public network. Although every transaction cannot be changed after it has been recorded, public blockchain applications may encounter information security and privacy risks[58][59]. A significant security vulnerability for the network is the potential for a 51% attack. In this scenario, malicious actors could gain control of the network and exploit it for their own benefit. They would have the ability to alter transactions and hinder others from creating blocks. Enhancing security measures at the protocol layer is crucial to mitigate this threat. Some of the network's security weaknesses have already been identified. However, only a limited number of situations possess resilient protocols capable of addressing this issue.

As a result, the long-term security of these protocols remains uncertain. A permissioned blockchain permits secure communication between entities that cooperate for a common aim but lack mutual trust.

Permissioned blockchains have several drawbacks, including the inability to process a large number of transactions, the requirement that smart contracts be written in a domain-specific language, the incapability to assist non-deterministic transactions, and the limited performance that results from the sequential execution of transactions. Since the most of current techniques can only offer a limited level of anonymity, more research is necessary to create a fully anonymized method that satisfies the demands of many security applications[60].Blockchain size is expanding rapidly, especially in IoT situations where data is gathered from multiple sensors and is difficult for IoT devices to store and manage due to their limited resources. As a result, this may have an impact on the use of IoT devices as full nodes to verify transactions on the blockchain network. To overcome IoT privacy and security issues, several approaches can be classified such as IoT software updates , Secure communication, and access control [61].

- **Energy Consumption**

Most blockchain technologies utilize Proof of Work as their consensus mechanism and operate on a similar infrastructure to Bitcoin. However, despite its apparent robustness, Proof of Work has its limitations. This system demands significant processing power to sustain mining operations, which involve computers solving complex mathematical problems. When mining starts, each PC will need an increasing little of electricity to get by this scenario. The blockchain employs energy in diverse situations such as: P2P energy trading, Energy Cryptocurrency, Internet of Things, Internet of Vehicles , Energy control & management , Carbon trading, Green certificate, Metering & billing, and Microgrid energy market. On the other hand, there are some limitations of blockchain that affect energy systems, such as: security risks ,high latency, lack of scalability, cost of decentralization, and low

throughput . Therefore, Blockchain can validate the transitions using additional consensus techniques that require a limited of energy consumption [62].

- **Lack of Clearness**

Organizations find it challenging to adopt blockchain technology because of its perceived immaturity. The ability to develop, deploy, and use blockchain applications is further restricted by the lack of trained human resources required for their development, management, and supervision. The limited deployment of blockchain technology is due to a lack of understanding, awareness, and potential[63].Therefore, the essence of the innovation made possible by blockchain is better understood by experts and investors, who are looking for chances to create startups that take into account the design characteristics of the innovation, particularly decentralization, and openness. As opposed to this, the market leaders in the commercial and public sectors are attempting to integrate new technology without surrendering some of its unique features[64].

2.2.4 Blockchain Network Applications

Blockchain technology holds the promise of transforming numerous industries through its decentralized and secure platform for transactions and data storage. There are some applications of blockchain networks:

- **Cryptocurrencies:** The most widely recognized use of blockchain technology is in cryptocurrencies such as Bitcoin and Ethereum. Blockchain allows for secure, transparent, and peer-to-peer transactions, eliminating the necessity for intermediaries.
- **Supply Chain Management** Blockchain has the capacity to elevate transparency and traceability within supply chains, thereby diminishing fraud and enhancing operational efficiency. Firms can leverage blockchain to monitor the journey of goods from their point

of origin to the final consumer. This safeguards product authenticity and acts as a deterrent against the infiltration of counterfeit items into the supply chain.[65].

- **Healthcare Records Management**: Blockchain offers decentralized and a secure way to manage EHRs and patient data. This approach ensures data privacy, reduces data breaches, and allows patients to manage access to their information[66].
- **Voting Systems:** Utilizing blockchain for voting systems has the potential to heighten the security and transparency of elections, offering tamper-resistant and verifiable voting records. It can potentially reduce voter fraud and increase voter participation. Nevertheless, there are many shortcomings in the infrastructure nowadays in place for employing EVS to conduct elections, which unauthorized parties may use to cast invalid elections or even tamper with the EVMs after the voting session has ended. A dependable, transparent, auditable, and tamper-proof e-voting system is urgently needed to enable a more trustworthy and fair election process[67].
- **Military and Defense:** Through the incorporation of blockchain, particularly in military applications yet to be fully explored, certain cybersecurity systems' vulnerabilities may shift from a single-point-of-failure model, where an adversary only needs to compromise one node to compromise the entire system, to a majority-compromised vulnerability model. In this latter scenario, an adversary would be unable to exploit a single point of failure. Despite this, blockchain-based military applications now seem to lack some key functionality. The applications for the military blockchain that will be deployed shortly are expected to be defense logistics and data security [68].

2.3 Cryptocurrancy Ecosytem

The cryptocurrency ecosystem refers to the collection of digital currencies, blockchain networks, and various supporting technologies and services that are associated with the use and trading of cryptocurrencies. This ecosystem is constantly evolving and expanding as more cryptocurrencies are introduced and more businesses and individuals adopt blockchain technology. Overall, the cryptocurrency ecosystem is a complex and rapidly evolving space that has the potential to revolutionize many industries and change the way we interact with money and data. One of the most significant subgroups of digital currencies is cryptocurrency. Cryptocurrencies have unique traits, as opposed to other digital currencies that are bound to fiat money or institutions, centrally dispersed, or restricted to a certain area. These cryptocurrencies use an open distributed ledger that is powered by blockchain technology to keep track of transactions. Decentralization enables more capacity, better security, and more rapid solution. The majority of these characteristics address the shortcomings of conventional financial systems[69].The key components of the cryptocurrency ecosystem include:

1. Cryptocurrencies: These are forms of digital currency that employ encryption methods to control the creation of currency units and validate fund transfers. Well-known examples of cryptocurrencies encompass Bitcoin, Ethereum, and Litecoin.

2. Blockchain technology: This constitutes the fundamental technology supporting the majority of cryptocurrencies. It operates as a distributed ledger system that logs transactions and facilitates the secure and transparent exchange of data.

3. Exchanges: These platforms enable users to purchase, sell, and exchange cryptocurrencies. Noteworthy cryptocurrency exchanges comprise Coinbase, Binance, and Kraken.

4. **Wallets**: These are digital wallets utilized for the storage, transmission, and reception of cryptocurrencies. Different types of wallets are available, such as software wallets, hardware wallets, and paper wallets.
5. **Mining:** This is the process of adding new blocks to a blockchain network by solving complex mathematical equations. Miners are rewarded with new units of cryptocurrency for their efforts.
6. **ICOs and STOs:** Initial Coin Offerings and Security Token Offerings are fundraising methods used by companies to raise capital for blockchain-based projects.
7. **Decentralized Applications (DApps):** These are applications built on top of blockchain networks that enable various use cases, such as DeFi, gaming, and social media.

2.3.1 Cryptocurrency Platforms

The blockchain, a distributed and decentralized ledger where data is kept consistent throughout the network by a peer-to-peer consensus protocol and protected by cryptography. Today referred to as "cryptocurrencies" are digital assets that use cryptographic security to solve the double-spending problem, often through a decentralized ledger rather than a centralized authoritative server. Bitcoin was the first and, to date, most successful platform for these digital assets. A cryptocurrency platform is a digital venue that enables users to engage in buying, selling, and trading cryptocurrencies. It typically functions as an online marketplace where users can establish accounts, deposit funds, and execute orders for the purchase or sale of different cryptocurrencies like Bitcoin, Ethereum, or Litecoin. These platforms come in various forms, including centralized exchanges, decentralized exchanges, and peer-to-peer exchanges. Centralized exchanges are the most widely used type and function as a third-party intermediary that facilitates transactions between

buyers and sellers. On the other hand, decentralized exchanges enable users to directly trade cryptocurrencies without requiring an intermediary. Peer-to-peer exchanges empower users to purchase and sell cryptocurrencies directly from other individuals, bypassing the need for a centralized platform. Prominent cryptocurrency platforms like Coinbase, Binance, Kraken, and Gemini typically provide a variety of functionalities including advanced trading tools, price charts, and mobile applications to assist users in managing their cryptocurrency assets. Nevertheless, it's crucial to acknowledge that these platforms may be vulnerable to security threats. Therefore, users should implement precautions like employing two-factor authentication and securing their cryptocurrency in reliable wallets [70].

2.4 Cyber Attacks in Cryptocurrencies

The various types of attacks have been grouped into four main categories. The four main security concerns to cryptocurrencies are discussed in this section. These cyber-attacks were successful, resulting in substantial losses or the denial of cryptocurrency services. In all cases, the attacker must achieve sufficient utility to justify the essential cost of an attack. Figure 2.6 illustrates the taxonomy of attacks on the cryptocurrency.

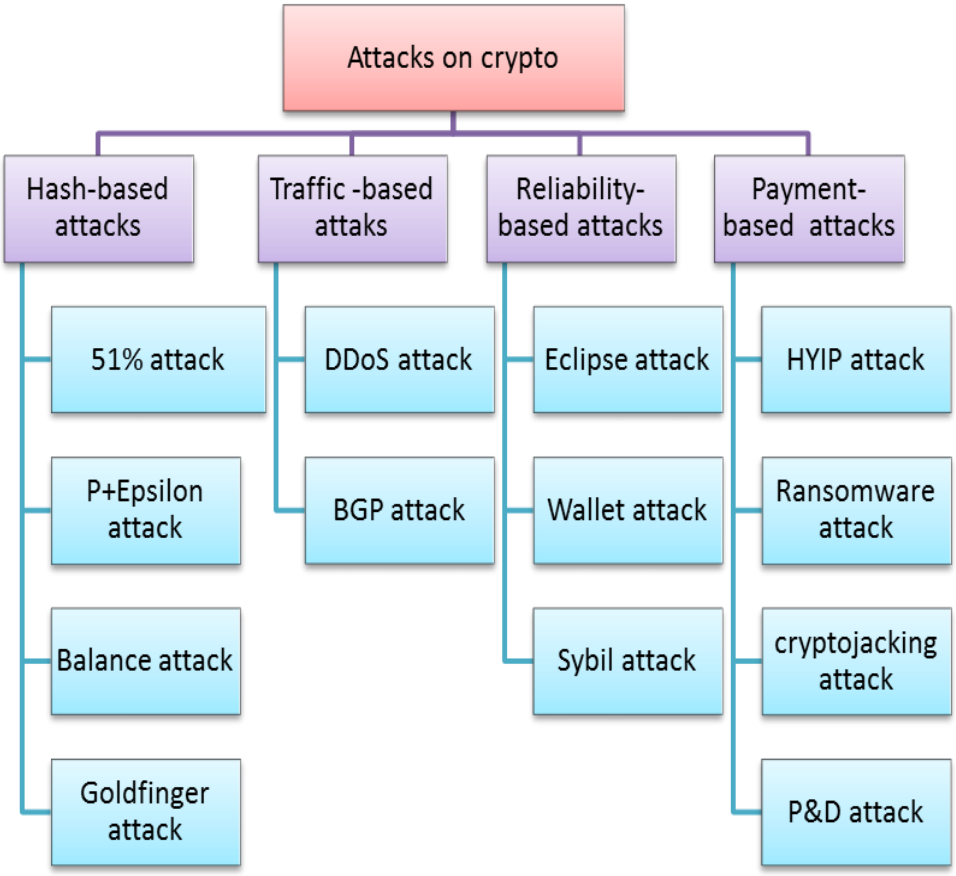


Figure 2.6: Taxonomy of attacks on cryptocurrency

a) Hash - based attack

This attack involves collecting hash values and then trying to find a matching hash value for different additional messages.

1- **51% attack:** This type of attack has an entirely negative effect on cryptocurrencies. A 51% attack takes place when a collective group of miners or an individual miner gains control over more than 50% of the network's mining power or devices. [71]. This form of threat starts with a private chain of blocks that is completely distinct from the genuine chain. Then, The network is supplied with the split chain so that it can be created as a real chain. By motivating network nodes to adhere to their chain, attackers that achieve 51% or more hashing power can drive the longest chain. When mining power is less than 40%, then 51%

attack can possibly occur but with a lesser probability, such as Bitcoin Gold [72].

2- **p + Epsilon attack:** This type of attack takes advantage of the network participants' prevailing technique. A blockchain based in facts of PoW is typically vulnerable to this type of attack. When attackers grant participants a payout, a payment matrix is used to obtain an advantage, with the dominant strategy supporting the attacker's aim fulfillment. In light of this, the participants receive no remuneration, whereas the attacker obtains the full amount [73].

3- **Balance attack :** It is a strategy that focuses on nodes with equally distributed mining power [74].This form of attack can be used to double the amount of money spent on a PoW consensus. By utilizing their limited hashing power, an attacker has the capability to slow down messages on the Ethereum network. This form of attack can be executed with just 5% of the available hashing power. [72]. Firstly, the attacker needs to pinpoint the subgroup involving the merchant before initiating transactions to buy products from them. Subsequently, the attacker must dispatch transactions to this subgroup and mine blocks for the remaining group nodes. [75].

4- **Goldfinger attack:** a majority attack, where the attacker is motivated by anything other than the cryptocurrency economy. Purchase of mining equipment, demand for rental (Nice Hash), and other indicators of dominance over the complete network hash rate can be observed. The objective of this attack is to disrupt the entire system [76].

b) Traffic -based attack

This attack can be classified into two categories, as follows:

1- **DDoS attack:** In a DDoS attack, numerous systems inundate the targeted system with an excessive amount of traffic, thereby overwhelming its resources and bandwidth. The target node refuses the transaction because the system is overloaded .Attackers utilize DDoS to prevent authentic transactions from being completed so that invalid transactions can be carried out. On the contrary, DDoS attacks can only significantly limit network activity. DDoS attacks are dangerous because they overload centralized systems with additional traffic. A DDoS attack is supposed to overwhelm centralized servers, although the bandwidth required to overwhelm them are nearly unachievable in most circumstances. According to research, DDoS becomes more prevalent, and every attack results in businesses incurring costs exceeding $2 million [71].

2- **Border Gateway Protocol Hijacking(BGP):** Border Gateway Protocol hijacking is a method in which an ISP disseminates deceptive routing system notifications to divert traffic. A routing attack is another name for it. In effect, This attack could potentially lead to the ability to carry out a double-spending attack .If the attacker wants to hijack all of the traffic for a valid prefix p, then either: (1) announce p or (2) announce a more specific prefix of p. In the first status, the attacker receives 50% of the traffic because BGP routers prefer shorter links. The longest-match entry is used by Internet routers to forward data, and the attacker engages all traffic destined to the destination in the second status [77].

c) Reliability -based attack

This attack can be classified into three categories, as follows:

1- **Eclipse Attack:** The eclipse attack enables an attacker to dominate all incoming and outgoing connections of the target, effectively isolating the victim from the rest of the network's peers[78]. Within

Bitcoin's peer-to-peer network, there are two varieties of eclipse attacks: the botnet attack and the infrastructure attack. The botnet attack is executed by bots with distinct IP address ranges. Conversely, the infrastructure attack mimics the threat posed by a company, an ISP, or even a nation-state with a substantial number of contiguous IP addresses [79].

2- **Wallet attack:** A wallet can be controlled by a software application, a hardware device, or an internet service that holds the private and public keys linked with the user's addresses. To transact with a cryptocurrency, users must have control over their cryptocurrency wallets. An attack on a wallet service provider, its users, or wallet software can have a significant impact, culminating in large coin theft and a loss of trust in the entire system. Coinbase is an online cryptocurrency exchange and wallet that, different from single-coin wallets, allows users to possess and trade multiple cryptocurrencies from the same account [76].Moreover, individual wallet user attacks can be carried out using various harmful techniques to steal user credentials and obtain access to their funds [80].

3- **Sybil attack:** Sybil attack is considered a type of reliability threat. It is a system node that manages several identities. Peer-to-peer networks rely on the concept of identity, in which each machine represents a single identity [81]. Douceur, a researcher at Microsoft, was the pioneer in drawing global attention to this attack method [82]. The assailants have the capability to set up numerous fake nodes that appear genuine to their peers. These deceptive nodes play a role in compromising the network by endorsing illicit transactions and altering legitimate ones [72].Even when the Bitcoin blockchain network has a substantial quantity of

nodes, resulting in a very expensive attach, whereas an opponent has a a considerable amount of network nodes, the possibilities of double spending increases.

d) Payment-based attack

A number of attacks that use cryptocurrencies as a payment method include the following:

1- **High Yield Investment Program(HYIP):** HYIP is considered a fraudulent activity. Thus, obtaining Bitcoin addresses linked to fraud to detect such illegal acts is crucial. Thus far, such actions have been identified by correlating Bitcoin addresses with graph mining techniques [83].According to certain studies, HYIPs account for 0.03 percent to 0.15% of smart contracts [84].Other sources believed that HYIP using Ethereum is worth approximately half a million dollars [36].

2- **Ransomware attack:** Ransomware is evolving and Enhancing malicious software designed to masquerade as Crypto or Locker, aiming to target and seize control over vital infrastructure and computer systems[85]. Some examples include CryptoWall, Cryptolocker, Manamecrypt, and CryptoDefense [86].A considerable increase is found in crypto-ransomware attacks, which encrypt individual files on a host or network-attached storage and demand a ransom in cryptocurrency [87].

3- **Cryptojacking attack:** In cryptojacking, an attacker executes crypto mining software on the devices of unknown. The two most common attacks in malware code are: web browser-based crypto mining and installable binary crypto mining. Hoya, Japan’s largest optical goods producer, shut down its production lines for three days as hackers attempted to set up an illegal cryptocurrency mining operation. A number of illegal mining operations have

already been found. “Bitcoin mining plot” has led to the arrest of Russian nuclear specialists [33].

4- **Pump &Dump attack (P&D):** Pump and dump fraud is considered a market manipulation scheme entails artificially inflating the price of a private security and subsequently selling it to other investors at a significantly elevated price [88].At present, hundreds of cryptocurrencies occur, the market is unregulated, and prices are easily influenced. Therefore, pump and dumps are extremely typical in these securities. Pump and dumps are currently led by a significant number of personality internet groups, and the movement has gone viral, despite that it is still relatively unknown [89].

2.5 The Cybersecurity of Cryptocurrency

In areas of use, such as online banking and credit card payment platforms, cybersecurity is a fairly developed industry. These systems have been working to establish a protected and dependable space for the exchange of confidential information, including records of financial transactions, passwords, and personal information[90].

Cybersecurity is a critical aspect of cryptocurrency due to the decentralized and digital nature of these assets.

The Blockchain Network is the framework employed by cryptocurrency developers in their construction process. To evaluate how well they can fix several aspects of cryptocurrencies, including the overall supply, the maximum amount that can be mined, and the different approaches to system protection. Certain researchers employed the daily ISE Cyber Security Index from Nasdaq Global Indexes, which is published on the Bloomberg platform, as a gauge of cyber-security. The index covers companies actively delivering cyber security technology and services, which began trading on December 31, 2010, with a base value

of 100. At the specified reference dates (i.e., the conclusion of January, April, July, and October annually),these factors must be an individual hardware/software developer or service provider of cyber security with a minimum open-ended value of $100 million and three-month average daily dollar trading volume of $1 million. Additionally, they must be listed on an eligible exchange with securities that have been seasoned for a minimum of three consecutive calendar months .Consequently, the index serves as a benchmark[91]. A few important considerations for cryptocurrency cybersecurity:

- **Network Security and Node Protection:** Ensuring the security of cryptocurrency nodes and network connections is vital to prevent attacks like DDoS attacks, man-in-the-middle attacks, and Sybil attacks[79].
- **Wallet Security:** Cryptocurrency wallets store private keys necessary to access and manage funds. Secure wallet management is crucial to prevent unauthorized access and theft[92].
- **Exchange Security:** Cryptocurrency exchanges are susceptible to hacking attacks due to the concentration of assets. Ensuring proper security measures and practices by exchanges is crucial[93].
- **Smart Contract Vulnerabilities:** Smart contracts on blockchain platforms can have coding vulnerabilities that hackers can exploit to drain funds or disrupt operations[94].
- **Regulatory Compliance:** Cryptocurrency businesses need to navigate regulatory frameworks To guarantee adherence to regulations regarding AML and KYC requirements. [95].
- **Privacy Concerns:** Many cryptocurrencies aim to enhance user privacy, but concerns have been raised about the effectiveness of privacy features[96].

2.6 The Distributed Denial of Service Attacks

A DDoS attack is a form of cyber assault that seeks to inundate a website or online service with traffic from numerous origins, rendering it inaccessible to users. In a DDoS attack, the attackers use a network of computers, often compromised by malware or other means, to send a massive amount of traffic to a single target, effectively flooding its servers and rendering it unable to handle legitimate requests.

DDoS attacks may be initiated for various motives, including seeking financial gains, revenge, activism, or even just for fun. Attackers often use botnets, which are networks of compromised devices under their control, to carry out these attacks. The devices can include computers, servers, IoT devices, and even smartphones.

To protect against DDoS attacks, websites and online services can use various mitigation techniques, such as Eliminating harmful traffic by employing CDNs to distribute traffic across multiple servers, and employing techniques to detect and block botnet traffic. It's also important to have a response plan in place in case of a successful attack, which may include notifying stakeholders, isolating affected systems, and collaborating with law enforcement if necessary. CDNs employ the tactic of distributing clustered traffic across numerous servers, thereby diminishing the impact of DDoS attacks. Another strategy involves the utilization of anomaly detection and behavior analysis to identify and obstruct malicious traffic. This entails scrutinizing network traffic for irregular patterns and deviations that may signify an ongoing DDoS attack. Upon detection, the system can take measures to intercept the traffic and alleviate the repercussions of the attack. Machine learning techniques can play a pivotal role in implementing this approach[97].

2.6.1 The Types of DDoS Attacks

DDoS attacks are a form of cyber assault designed to inundate a website or network with an overwhelming volume of traffic from various sources, rendering it inaccessible to legitimate users. These attacks can be categorized into three main types: application layer attacks, protocol attacks, and volumetric attacks. Application layer attacks, such as reflection/amplification based flooding attacks and HTTP flooding attacks, involve the attacker initiating multiple requests from a compromised host. These requests appear to originate from a genuine user. This type of attack exploits vulnerabilities introduced by the implementation of services like HTTPS and HTTP on TCP ports 443 and 80.The first category of attacks in this type is flooding attacks based on reflection/amplification. which can be classified into five attack types: DNS amplification attacks , Smurf, NTP amplification attacks , Fraggle, and SNMP attacks. The second category is known as HTTP flooding attacks, which can be divided into four types: asymmetric attack, slowloris attack, session flooding attack, and request flooding attack. Protocol attacks, which install and execute malicious code of the target's protocol and use of certain features. Examples of attacks that fall under this category are TCP SYN Flood attacks, Ping of Death attacks, and fragmented packets. Now each category's representative attack is described [98] [99]. Some of the DDoS attacks can be illustrated as follows:

A. ICMP Flood: This attack floods the victim's network with A significant quantity of ICMP packets, which are used for network diagnostics. These packets can overwhelm the victim's network and cause it to crash.

B. SYN Flood: This attack dispatches an extensive quantity of TCP SYN (synchronization) packets to the victim's server, overwhelming it with connection requests and preventing legitimate users from accessing the service. It happens when an attacker repeatedly sends TCP Synchronize (SYN) requests to the target to exhaust all available resources and render the server inaccessible to authorized users. Because an SYN request initiates network communication between a potential client and the target server, this situation. When an SYN request is received, the server acknowledges it and keeps the lines of communication open while it waits for the client to confirm the open connection. In contrast, a successful SYN flood prevents the client from acknowledging the request, using up server resources until the connection times out. All of the target server's resources are depleted by a huge number of incoming SYN requests, which leads to a successful DDoS assault.

C. UDP Flood: This attack floods the victim's network with a large number of UDP packets, which are used for non-connection-oriented applications like online gaming or video streaming. The victim's network can become overwhelmed with these packets, causing it to crash.

D. HTTP Flood: This attack floods the victim's web server with a large number of HTTP requests, often generated by botnets, which can overload the server and make it inaccessible to legitimate users.

E. Slowloris: This attack is designed to exploit the way that web servers handle connections. It transmits a substantial volume of incomplete HTTP requests to the server, without finalizing them, thereby occupying server resources and obstructing legitimate users from accessing the service.

F. NTP Amplification: This attack exploits vulnerabilities in the NTP to generate a flood of traffic to the victim's network. The attacker spoofs the victim's IP address and sends a small number of requests to NTP servers, which then respond with a large amount of traffic to the victim's network.

G. DNS Amplification: This attack capitalizes on weaknesses in the DNS to produce a surge of traffic directed at the victim's network. The attacker manipulates the victim's IP address and sends a small number of requests to DNS servers. In response, these servers generate a significant volume of traffic aimed at the victim's network.

H. Ping of Death attacks : POD was the previous name for ICMP ping flood assaults. The largest IPv4 packet size that can be exchanged between two devices is 65,535 bytes. Sending large or improperly formatted packets using a straightforward ping command can do significant damage to an unpatched system.

2.6.2 The Mechanism of Action of DDoS Attack

A DDoS attack can be launched in various ways. Due to the weak cyber protection standards now in place for blockchain, an adversary has focused their attention to target the applications that make up the blockchain ecosystem in order to conduct harmful activity. Applications built on the blockchain, like Bitcoin and Ethereum, have frequently fallen victim to these attacks. DDoS attacks may manifest in different ways, influenced by factors such as the application's attributes, the structure of the network, and the behavior of peers. Because the Bitcoin network can only handle a certain number of transactions per block at one time, this presents another possibility for the assault. For instance, the Bitcoin network requires 10 minutes on average to create a block with a maximum size of 1 MB. Reportedly, the maximum number of

transactions that can be added to a Bitcoin block is 2210, even though the size of Bitcoin transactions can fluctuate. On a typical basis, a transaction in Bitcoin is 500 bytes in size, allowing for around 2000 transactions each block. The adversary might send out multiple dust transactions, one for every transaction. One of the common methods used to launch DDoS attacks is botnets. A botnet is a network of infected devices (such as computers, servers, and IoT devices) that are controlled by a single attacker. The attacker has the capability to utilize the botnet to inundate a target server or network with excessive traffic, rendering it inaccessible to legitimate users. Also sending a stream of packets to a server under assault is the most typical method used by attackers[100]. An attacker runs the malware in bot systems and launches DDoS attacks in order to produce a lot of network traffic. Attack traffic from DDoS attacks comes from several source devices. In order to overwhelm the communication link bandwidth, system RAM, and CPU resources, the generated traffic is directed towards a singular victim system or infrastructure. DDoS assaults are commonly used in large-scale attacks to overwhelm their targets with resources[101].

2.7 The Machine Learning- based Approaches

In machine learning, a range of techniques is employed to tackle data-related challenges. Data scientists emphasize that there isn't a universal algorithm that excels in all scenarios. The choice of algorithm hinges on the specific problem being addressed, the number of variables involved, the optimal type of model, and various other considerations. The objective of machine learning is to glean insights from data. The pursuit of enabling robots to learn autonomously, without explicit programming, has been a focal point of extensive research. Many mathematicians and programmers employ a diverse range of techniques

to address this challenge, which revolves around processing vast amounts of data[102]. A hundred programs and numerous organizations have been replaced in recent years with Artificial Intelligence, Deep Learning, and other Machine Learning approaches. Therefore, artificial intelligence and machine learning enable tools and applications to become more accurate when anticipating results, fulfilling the objectives and requirements of smart environments[103].

In recent years, artificial intelligence and machine learning have become actively applied in the analysis of traditional financial markets. Technological advancements can now be programmed to mimic and replicate human behavior and cognitive processes, enabling them to perform tasks typically carried out by humans. This encompasses machine learning within artificial intelligence, allowing computers to autonomously learn from input-output data and generate new data through various transformations. Cryptocurrency developers employ the Blockchain Network as their framework for evaluating and addressing various aspects of cryptocurrencies, such as total supply, maximum mineable amount, and security mechanisms within the system.[104].

Therefore, at the present time, machine learning is used in many fields, the most important of which is cybersecurity, and the detection of many attacks such as DDoS attacks. A sophisticated branch of machine learning known as deep learning takes machine learning closer to artificial intelligence. It makes complex relationships and concepts easier to model .There are four main categories of machine learning algorithms: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. [105]. The first and second types have been included in this dissertation, as well as the most important type, which is deep learning. This dissertation includes the most significant type, deep learning, in addition to the first and second types. Some concepts about

the types of machine learning used in this thesis will be clarified as follows:

2.7.1 Deep Learning Algorithms

Deep learning is a particular type of machine learning. To address issues with unsupervised learning, numerous deep learning algorithms have been developed. None, however, have managed to resolve the issue in the same manner that deep learning has managed to resolve the supervised learning problem for a variety of tasks. Complex mathematical processes are carried out by the deep learning algorithms using numerous hidden layers and a large number of parameters during the training phase. [106]. Five categories have been established for the deep learning methods: semi-supervised learning such as Autoencoders, supervised instance learning such as Deep Neural Networks and Convolutional Neural Networks, supervised sequence learning such as RNN and LSTM, hybrid learning, and additional learning methods[107].This dissertation used both RNN and LSTM as a hybrid deep learning model for the detection of DDoS attacks. Certain aspects of RNN and LSTM algorithms are explained in the brief explanation that follows.

A. The Recurrent Neural Network

The RNN enables the input of sequential data, allowing the network to use the output from one step as an input along with the data input from the subsequent step. This grants RNN the flexibility to retain the output in memory and pass that knowledge along to the next set of training data, in contrast to the typical Artificial neural network. Figure 7 displays the RNN's structure, which may be mathematically defined as[5]:

$$h^{(t)} = f(h^{(t-1)}, \vec{x}^{(t)}) \ldots (2.1)$$

If x indicates the observation input, f indicates the network non-linear function, and h indicates the hidden state. The recursive relationship could be alternatively represented algebraically in the following manner:

$$h^{(t)} = \sigma 1(W_{hh} \cdot h^{(t-1)} + W_{xh} \cdot \vec{x}^{(t)} + h_0^{(t-1)}) \ldots (2.2)$$

$$y^{(t)} = \sigma 2(W_{hy} \cdot h^{(t)} + h_0^{(t)}) \ldots (2.3)$$

where W refers to the weight matrix, h_0 indicates the bias term, and σ1 and σ2 refer to activation functions.

On the other hand, capturing a connection between historical memory over a longer period of time is challenging. However, RNN was thus limited to learning only the most recent output from a few prior layers. This is a significant drawback of conventional RNN. As a result, there was a lot of research done to address this RNN deficit, and LSTM is one of the methods that accomplished so successfully. We will go through how LSTM does this by connecting various blocks utilizing the memory block design and cell states in the next section.

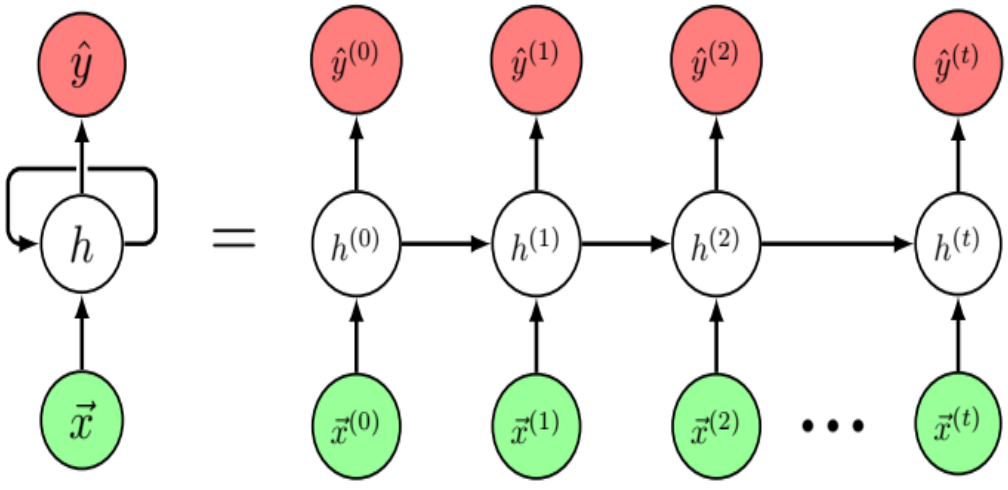


Figure 2.7: Simple Recurrent Neural Network (Left) and Unfolded Recurrent Neural Network (Right)[5]

B. Long Short-Term Memory

Schmidhuber and Hochreiter introduced LSTM in 1997[108]. They dealt with the RNN's short-term reliance issue. The LSTM network's structure enables it to get feedback as it learns from a series of data, focusing on understanding the temporal connections between data points rather than fixating solely on repetitive learning from a single data point. Every time a set of data points is entered, an input gate (i^t), forget gate (f^t), and output gate (o^t) are used by the LSTM for learning purposes. The input data (X^t), this is determined by the learning window of l days. and data points such that (X^t) is created by ($x(t\text{-}l\text{+}1)$, $\vec{x}(t\text{-}l\text{+}2)$,. . . , $\vec{x}(t)$), and the output from the previous memory block (ht1) are connected to the network between LSTM memory blocks. Finally, the network updates the current memory block output (h^t) and the current cell state (C^t). Figure 8 shows the complete structure.

The LSTM algorithm is specifically given by:

$$\mathbf{f^t = \sigma(W_f \cdot [h^{t-1}, X^t] + b_f)\ldots(2.4)}$$

W_f represents the weight matrix, b_f is the bias for the associated gate, $\sigma(s)$ is the sigmoid function, $\sigma(s) = 1/1+e^{-s}$.

$$\mathbf{i^t = \sigma(W_i \cdot [h^{t-1}, X^t] + b_i)\ldots(2.5)}$$

$$\mathbf{C^{\sim t} = \tanh(W_C \cdot [h^{t-1}, X^t] + b_C)\ldots(2.6)}$$

The hyperbolic tangent function, represented as tanh, is used in this context.

$$\mathbf{C^t = \sigma(f^t \otimes C^{t-1} + i^t \otimes C^{\sim t})\ldots(2.7)}$$

where the Hadamard product is represented by $\otimes$. To determine how much the LSTM should forget about the prior cell state and update from the present state, the equations for the forget gate f^t and input gate i^t scaled the values of C^{t-1} and $C^{\sim t}$, respectively. This sequentially connects the LSTM memory blocks.

$$o^t = \sigma(W_o \cdot [h^{t-1}, X^t] + b_o)\ldots(2.8)$$

$$h^t = o^t \otimes \tanh(C^t)\ldots(2.9)$$

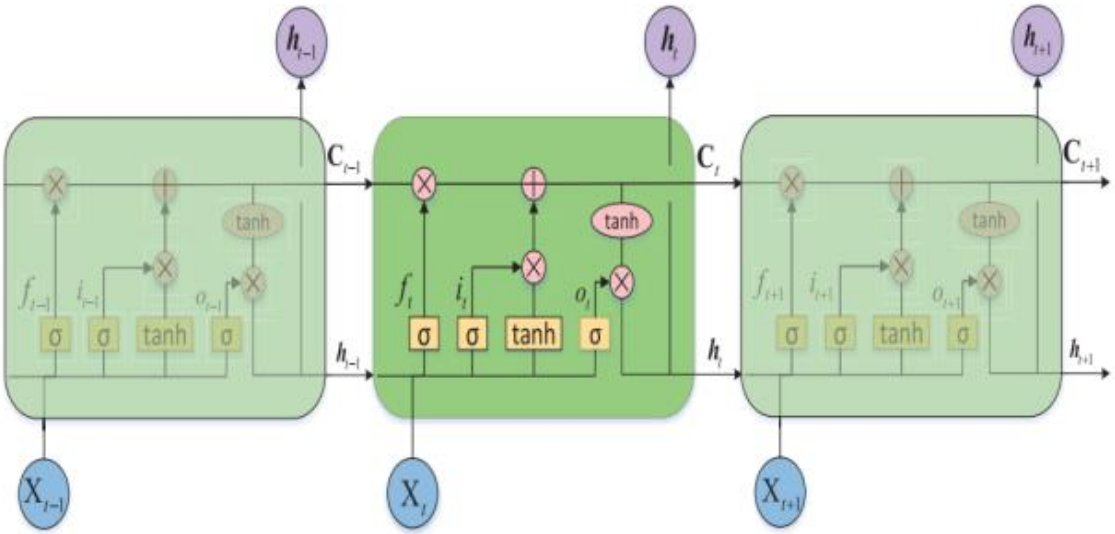


Figure 2.8 LSTM memory blocks structure [6]

2.7.2 Supervised Learning Algorithms

Supervised Machine Learning is employed to address regression and classification challenges. By leveraging labeled data, these algorithms learn from historical and current information to make predictions about future events. The process initiates with a training phase, during which the ML system creates a derived function to forecast output values. Once sufficiently trained, the system can generate results based on input data. To refine the model and rectify any shortcomings, the ML algorithm assesses the produced outcomes against the actual and anticipated results. Many supervised learning methods encompass various sub-configurations that can impact both model performance and the risk of overfitting. These are known as hyper parameters or tuning parameters, such as the depth of a classification tree or the number of trees used in constructing a random forest[109]. Our dissertation problem implements a supervised ML algorithm at the level of feature selection by using a

random forest algorithm. The idea of Random Forest is explained in the following way:

- **Random Forest Algorithm**

The RF [110] method finds extensive application in classification tasks. It employs bagging and decision tree models with subsets as a key component of its machine-learning approach. To mitigate correlation within the bootstrapped ensemble, it selects a subset of features from each tree node. RF is a forest of k trees and is computed as follows for the objective of classifying n companies X_j (j=1,k).

$$\mathbf{RF}=\{DT_i\}, \; j=1,k \ldots (2.10)$$

RF performs at making high predictions for feature selection issues. Each tree in the categorization forest determines which class the most recent instance should be assigned according to the multiple decision trees that have been used. The new class will have its classification decided by majority voting. Accuracy increases with more trees used in the decision making process. It is necessary to specify the number of trees before using the classifier on the datasets[111].

2.7.3 Unsupervised Learning Algorithms

Unsupervised machine learning techniques are used when the training set of data is unclassified and unlabeled . It is used for solving the problems of clustering and association. It explores how the system can deduce a function from the unlabeled data to reveal underlying patterns. Even though the system doesn't have knowledge of the correct output, it scrutinizes the data and makes observations from the dataset to unveil hidden patterns within the unlabeled data. The most common unsupervised learning approach to grouping data into disjoint groups is clustering, which aims to make sure that data points belonging to the same group share similar features about a reference point while those belonging to different groups have different characteristics. Clusters are

groups that fit the description. As a result, clusters are made up of various data or objects that are comparable to a referral point. Clustering is an immensely important technique in various fields of engineering and research, including data compression, statistical data analysis, pattern recognition, data mining, artificial intelligence, and more[112]. In addition, hierarchical clustering solutions offer a view of the data at various levels of abstraction making them beneficial for users to visualize and interactively investigate significant collections[113].

Our dissertation applies two cluster algorithms at the attack data level obtained from the detection model . The clustering algorithms that was used is the Agglomerative Hierarchical Algorithm and Gaussian Mixture Model Algorithm.

a) Agglomerative Hierarchical Clustering: analysis is a cluster analysis technique used in statistics and data mining that aims to build a hierarchy of clusters. The two main approaches used in HC are agglomerative (bottom-up) and divisive (top-down). Each instance is treated as a cluster when using AHC methods, and the clusters are then combined to form bigger clusters. This keeps happening until every cluster is combined into a single big cluster that contains every instance. DHC methods split a large cluster into smaller ones until each cluster contains a single instance. Initially, all instances are members of one cluster[114].

Depending on the criteria used to select which clusters to merge, there are a variety of agglomerative algorithms such as ward, single, complete, and average. The most well-known of them is the Ward technique. This approach merges the two clusters that have the lowest costs The Ward method criterion permits the merger of the two clusters with the minimal achievable increase in the total within-cluster variance.[115]. The method

'ward' uses the Ward variance minimization algorithm. The new entry d(u,v) is computed by using the equation (2.11).

$$d(u,v) = \sqrt{\frac{|v|+|s|}{T} d(v,s)^2 + \frac{|v|+|t|}{T} d(v,t)^2 - \frac{|v|}{T} d(s,t)^2} \quad \ldots(2.11)$$

where u is the newly joined cluster containing clusters s and t, v is an unused cluster in the forest, T=|v|+|s|+|t| , and |* | is the cardinality of its argument. The linkage criterion determines the distance measure between sets of observations. ACH merges pairs of clusters that minimize this criterion. Our dissertation used "ward" as a linkage criterion. If linkage is "ward" then the metric to use when calculating the distance between instances in a feature array is Euclidean Distance, which can be calculated by using the equation(2.12).

$$d(p,q) = \sqrt{\sum_{i=1}^{N} (q_i - p_i)^2} \ldots (2.12)$$

Where p,q represents two points in Euclidean n-space, q_i, p_i represents Euclidean vectors, starting from the initial point , and n represents n-space.

b) Gaussian Mixture Model Algorithm: In clustering issues, the traditional Gaussian mixture model is commonly used. The distribution density function of the sample points is represented using the weighted average sum of multiple Gaussian functions. Essentially, the GMM serves as a probability density function, as defined. The probability density function's integrals across its range must add up to 1. The probability density functions associated with different Gaussian components are linearly added to create the probability density function of the entire GMM, and the integral of each Gaussian component's probability density function must be 1. It essentially models the probability density function of sample points by combining weighted Gaussian functions. For a given dataset x in R^D, the model includes parameters like the weight π_n for the

n-th Gaussian, mean vector M_n in R^D, and covariance matrix $\sum_n$ in $(R+)^D \times (R+)^D$, where N, D $\in$ N+ .Hence, it is imperative to assign a weight to each Gaussian component that does not exceed 1, ensuring that the sum of these weights totals 1. This condition is crucial to maintain the integral of the entire GMM's probability density equal to 1, a calculation facilitated by equation (2.13)[116].

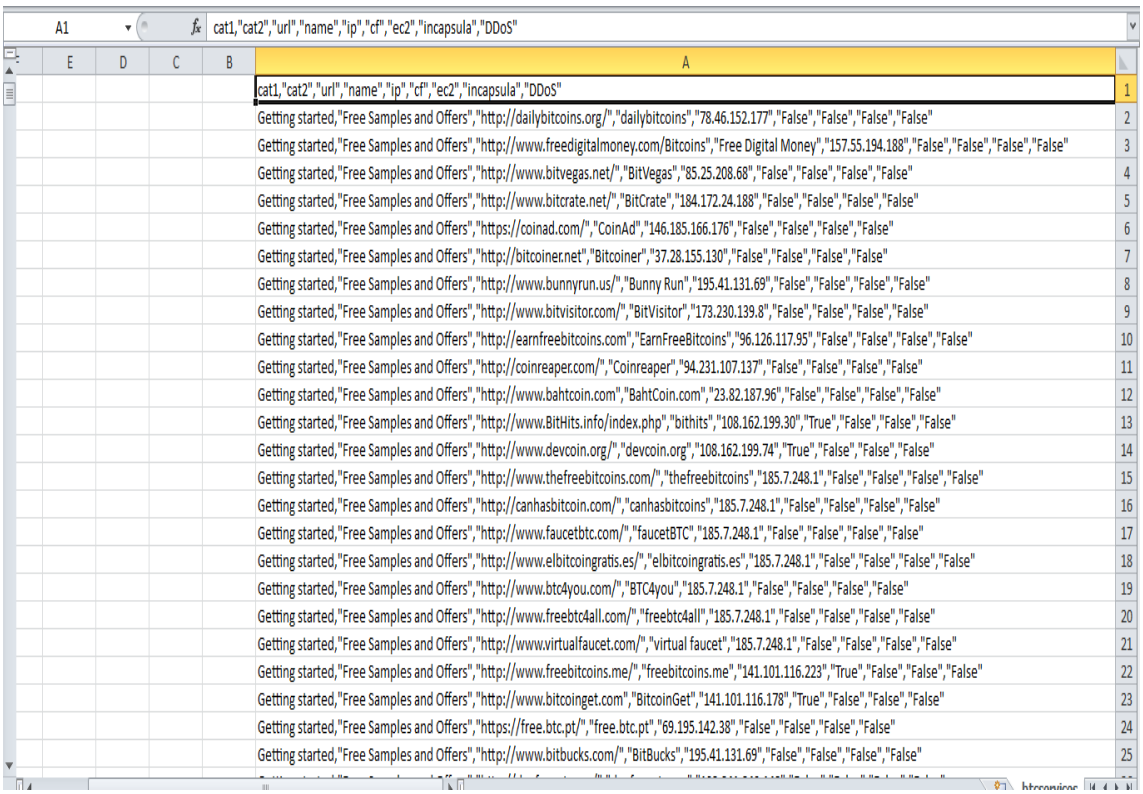
$$G(x) = \sum^{N}_{n=1} \pi_n \phi(x; M_n, \Sigma_n) \quad (2.13)$$

In this context, ϕ is denoted as the normal distribution.

2.8 Datasets Overview

For the purpose of supporting our dissertation, Three datasets used, namely:

1- Mt.Gox Dataset: The Mt. Gox dataset refers to a collection of data related to the Mt. Gox Bitcoin exchange, which was once one of the world's largest and most widely used cryptocurrency exchanges. However, in 2014, Mt. Gox filed for bankruptcy and claimed to have lost approximately 850,000 BTCs , valued at that time at hundreds of millions of dollars, due to hacking and security breaches. This dataset include various types of information, such as transaction data where information about Bitcoin transactions conducted on the Mt. Gox platform, including timestamps, transaction IDs, wallet addresses, and transaction amounts. It discovers that mining pools and currency exchanges have significantly higher odds of having DDoS defense systems like CloudFlare, Incapsula, or Amazon Cloud. It is demonstrated that operators who have not experienced an assault are over three times less likely to purchase anti-DDoS services than those who have. The present dissertation applied the hybrid model on this dataset. Sample data from the Mt.Gox dataset as can be seen in Figure (2.9) [13].



	A
1	cat1,"cat2","url","name","ip","cf","ec2","incapsula","DDoS"
2	Getting started,"Free Samples and Offers","http://dailybitcoins.org/","dailybitcoins","78.46.152.177","False","False","False","False"
3	Getting started,"Free Samples and Offers","http://www.freedigitalmoney.com/Bitcoins","Free Digital Money","157.55.194.188","False","False","False","False"
4	Getting started,"Free Samples and Offers","http://www.bitvegas.net/","BitVegas","85.25.208.68","False","False","False","False"
5	Getting started,"Free Samples and Offers","http://www.bitcrate.net/","BitCrate","184.172.24.188","False","False","False","False"
6	Getting started,"Free Samples and Offers","https://coinad.com/","CoinAd","146.185.166.176","False","False","False","False"
7	Getting started,"Free Samples and Offers","http://bitcoiner.net","Bitcoiner","37.28.155.130","False","False","False","False"
8	Getting started,"Free Samples and Offers","http://www.bunnyrun.us/","Bunny Run","195.41.131.69","False","False","False","False"
9	Getting started,"Free Samples and Offers","http://www.bitvisitor.com/","BitVisitor","173.230.139.8","False","False","False","False"
10	Getting started,"Free Samples and Offers","http://earnfreebitcoins.com","EarnFreeBitcoins","96.126.117.95","False","False","False","False"
11	Getting started,"Free Samples and Offers","http://coinreaper.com/","Coinreaper","94.231.107.137","False","False","False","False"
12	Getting started,"Free Samples and Offers","http://www.bahtcoin.com","BahtCoin.com","23.82.187.96","False","False","False","False"
13	Getting started,"Free Samples and Offers","http://www.BitHits.info/index.php","bithits","108.162.199.30","True","False","False","False"
14	Getting started,"Free Samples and Offers","http://www.devcoin.org/","devcoin.org","108.162.199.74","True","False","False","False"
15	Getting started,"Free Samples and Offers","http://www.thefreebitcoins.com/","thefreebitcoins","185.7.248.1","False","False","False","False"
16	Getting started,"Free Samples and Offers","http://canhasbitcoin.com/","canhasbitcoins","185.7.248.1","False","False","False","False"
17	Getting started,"Free Samples and Offers","http://www.faucetbtc.com/","faucetBTC","185.7.248.1","False","False","False","False"
18	Getting started,"Free Samples and Offers","http://www.elbitcoingratis.es/","elbitcoingratis.es","185.7.248.1","False","False","False","False"
19	Getting started,"Free Samples and Offers","http://www.btc4you.com/","BTC4you","185.7.248.1","False","False","False","False"
20	Getting started,"Free Samples and Offers","http://www.freebtc4all.com/","freebtc4all","185.7.248.1","False","False","False","False"
21	Getting started,"Free Samples and Offers","http://www.virtualfaucet.com/","virtual faucet","185.7.248.1","False","False","False","False"
22	Getting started,"Free Samples and Offers","http://www.freebitcoins.me/","freebitcoins.me","141.101.116.223","True","False","False","False"
23	Getting started,"Free Samples and Offers","http://www.bitcoinget.com","BitcoinGet","141.101.116.178","True","False","False","False"
24	Getting started,"Free Samples and Offers","https://free.btc.pt/","free.btc.pt","69.195.142.38","False","False","False","False"
25	Getting started,"Free Samples and Offers","http://www.bitbucks.com/","BitBucks","195.41.131.69","False","False","False","False"

Figure 2.9 : Mt.Gox Dataset Sample

The Features of the Mt.Gox Dataset represent identifying all posts including the term “DDoS” on the website bitcointalk.org appearing between February 2011 and October 2013. Table 2.1 illustrates the features of Mt.Gox dataset. It contains 9 features as follows:

cat1,cat2: represent category and subcategory information for 1240 online services supporting Bitcoin and 32 mining pools.

URL: a local copy of the page and automatically extracted the thread title.

name: represents the names of the page that performs Bitcoin services.

IP: represents the IP address of these pages

cf,ec2, and Incapsula: represents the identification of the usage of anti-DDoS providers by retrieving the addresses of all Bitcoin services that are currently in use and contrasting them with IP ranges that are known to be used by CloudFlare, Incapsula, and Amazon Web Services. Amazon

hosts material, while CloudFlare and Incapsula are content distribution networks (CDNs). The IP range identifies all three of them.

DDoS: represents the label that contains two values either true or false which means either attack or benign.

Table 2.1: The Features of the Mt.Gox Dataset

No.	All Features
1	cat1
2	cat2
3	URL
4	name
5	IP
6	cf
7	ec2
8	Incapsula
9	DDoS

2- CIC-IDS2017: The Canadian Institute for Cybersecurity Intrusion Detection Evaluation Dataset 2017 is a benchmark dataset for evaluating IDS. It was created by the Canadian Institute for Cybersecurity at the University of New Brunswick, Canada. The dataset was designed to help researchers and practitioners evaluate the performance of various intrusion detection techniques and algorithms. The feature selection model will apply to the CIC-IDS2017. Figure 2.10 displays the CIC-IDS2017 dataset sample.

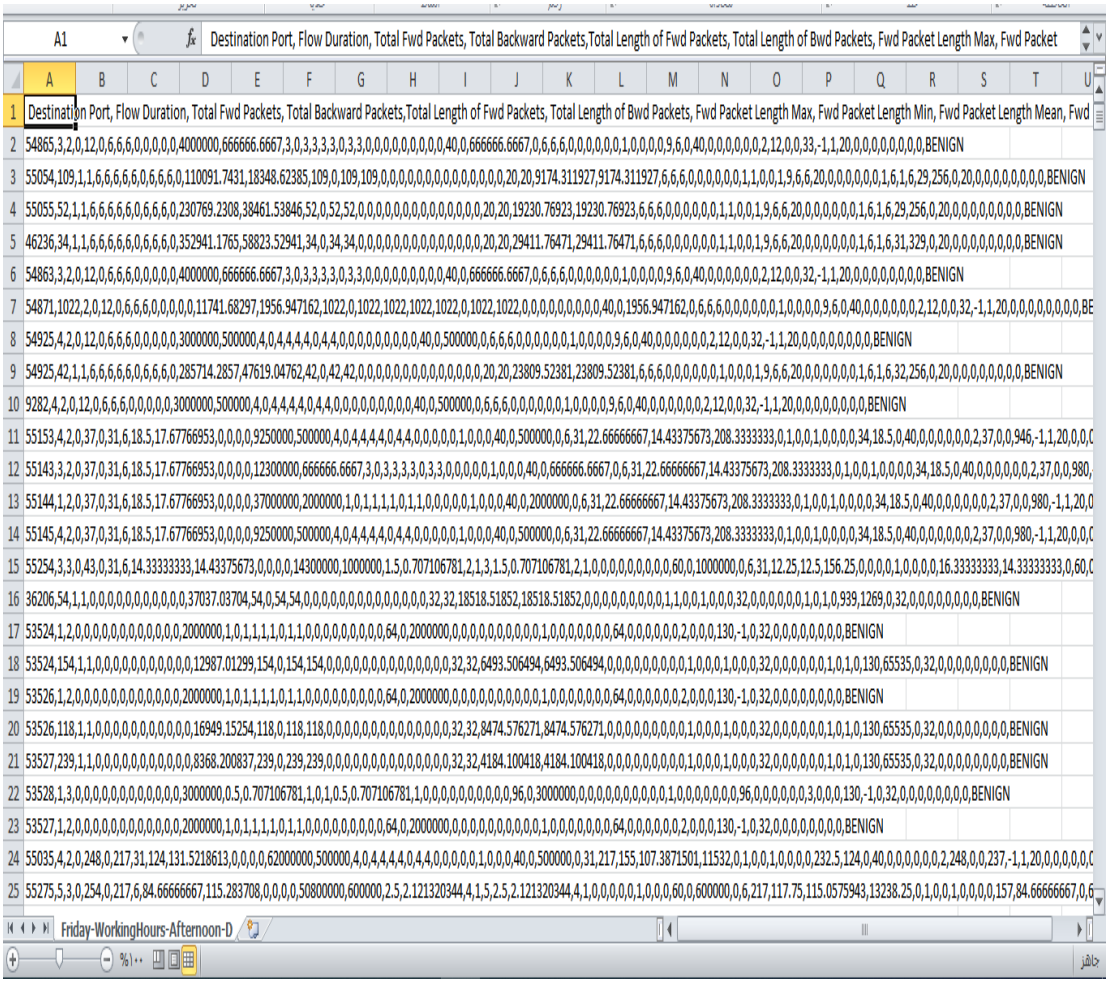


Figure 2.10: Sample of CIC-IDS2017 Dataset

The 79 features of the CIC-IDS2017 Dataset are listed in Table (2.2).

Table 2-2: The features of the CIC-IDS2017 Dataset

No.	Feature Name	No.	Feature Name	No.	Feature Name
1.	Destination Port	28.	Bwd IAT Std	55.	Avg Bwd Segment Size
2.	Flow Duration	29.	Bwd IAT Max	56.	Fwd Header Length
3.	Total Fwd Packets	30	Bwd IAT Min	57.	Fwd Avg Bytes/Bulk
4.	Total Backward Packets	31.	Fwd PSH Flags	58.	Fwd Avg Packets/Bulk
5.	Total Length of Fwd Packets	32.	Bwd PSH Flags	59.	Fwd Avg Bulk Rate
6.	Total Length of Bwd Packets	33.	Fwd URG Flags	60.	Bwd Avg Bytes/Bulk
7.	Fwd Packet Length Max	34.	Bwd URG Flags	61.	Bwd Avg Packets/Bulk
8.	Fwd Packet Length	35.	Fwd Header	62.	Bwd Avg Bulk Rate

No.	Feature Name	No.	Feature Name	No.	Feature Name
	Min		Length		
9.	Fwd Packet Length Mean	36.	Bwd Header Length	63.	Subflow Fwd Packets
10.	Fwd Packet Length Std	37.	Fwd Packets/s	64.	Subflow Fwd Bytes
11.	Bwd Packet Length Max	38.	Bwd Packets/s	65.	Subflow Bwd Packets
12.	Bwd Packet Length Min	39.	Min Packet Length	66.	Subflow Bwd Bytes
13.	Bwd Packet Length Mean	40.	Max Packet Length	67.	Init_Win_bytes_forward
14.	Bwd Packet Length Std	41.	Packet Length Mean	68.	Init_Win_bytes_backward
15.	Flow Bytes/s	42.	Packet Length Std	69.	act_data_pkt_fwd
16.	Flow Packets/s	43.	Packet Length Variance	70.	min_seg_size_forward
17.	Flow IAT Mean	44.	FIN Flag Count	71.	Active Mean
18.	Flow IAT Std	45.	SYN Flag Count	72.	Active Std
19.	Flow IAT Max	46.	RST Flag Count	73.	Active Max
20.	Flow IAT Min	47.	PSH Flag Count	74.	Active Min
21.	Fwd IAT Total	48.	ACK Flag Count	75.	Idle Mean
22.	Fwd IAT Mean	49.	URG Flag Count	76.	Idle Std
23.	Fwd IAT Std	50.	CWE Flag Count	77.	Idle Max
24.	Fwd IAT Max	51.	ECE Flag Count	78.	Idle Min
25.	Fwd IAT Min	52.	Down/Up Ratio	79.	Label
26.	Bwd IAT Total	53.	Average Packet Size		
27.	Bwd IAT Mean	54.	Avg Fwd Segment Size		

3- CSE-CIC-IDS2018 on AWS: Refers to a dataset called the "Canadian Institute for Cybersecurity Intrusion Detection Evaluation Dataset" for the year 2018. It's designed for testing and evaluating IDS and contains various network traffic data to simulate cyber-attacks and normal network activities. The dataset includes a wide range of network traffic scenarios, making it valuable for developing and testing intrusion detection algorithms. AWS refers to Amazon Web Services. Figure (2.11) displays a sample of the CSE-CIC-IDS2018 dataset.

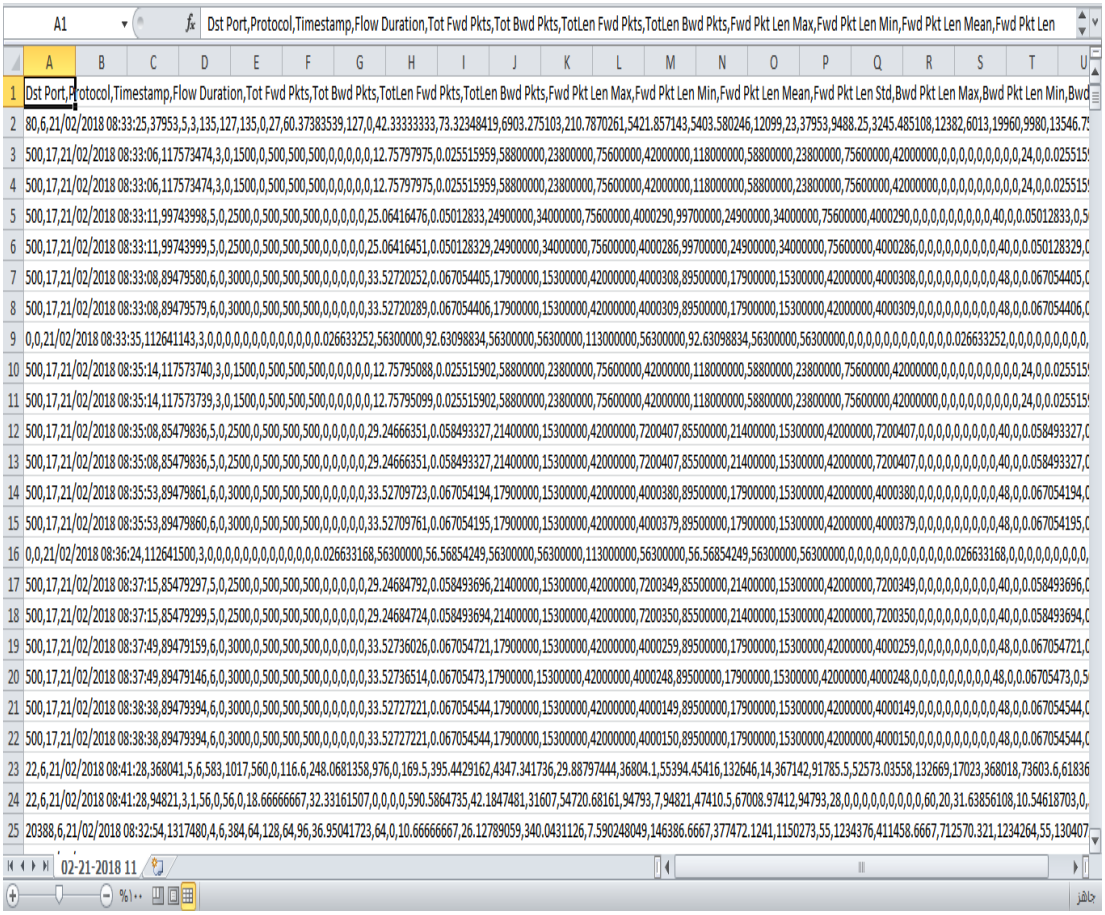


Figure 2.11: Sample of CSE-CIC-IDS2018 dataset

"CSE-CIC-IDS2018" dataset contains 80 features according to Table (2.3).

Table 2.3: The Features of the CSE-CIC-IDS2018 Dataset

No.	Feature Name	No.	Feature Name	No.	Feature Name
1.	Dst Port	28.	Bwd IAT Tot	55.	Pkt Size Avg
2.	Protocol	29.	Bwd IAT Mean	56.	Fwd Seg Size Avg
3.	Timestamp	30	Bwd IAT Std	57.	Bwd Seg Size Avg
4.	Flow Duration	31.	Bwd IAT Max	58.	Fwd Byts/b Avg
5.	Tot Fwd Pkts	32.	Bwd IAT Min	59.	Fwd Pkts/b Avg
6.	Tot Bwd Pkts	33.	Fwd PSH Flags	60.	Fwd Blk Rate Avg
7.	TotLen Fwd Pkts	34.	Bwd PSH Flags	61.	Bwd Byts/b Avg
8.	TotLen Bwd Pkts	35.	Fwd URG Flags	62.	Bwd Pkts/b Avg
9.	Fwd Pkt Len Max	36.	Bwd URG Flags	63.	Bwd Blk Rate Avg
10.	Fwd Pkt Len Min	37.	Fwd Header Len	64.	Subflow Fwd Pkts
11.	Fwd Pkt Len Mean	38.	Bwd Header Len	65.	Subflow Fwd Byts
12.	Fwd Pkt Len Std	39.	Fwd Pkts/s	66.	Subflow Bwd Pkts
13.	Bwd Pkt Len Max	40.	Bwd Pkts/s	67.	Subflow Bwd Byts
14.	Bwd Pkt Len Min	41.	Pkt Len Min	68.	Init Fwd Win Byts
15.	Bwd Pkt Len Mean	42.	Pkt Len Max	69.	Init Bwd Win Byts
16.	Bwd Pkt Len Std	43.	Pkt Len Mean	70.	Fwd Act Data Pkts
17.	Flow Byts/s	44.	Pkt Len Std	71.	Fwd Seg Size Min
18.	Flow Pkts/s	45.	Pkt Len Var	72.	Active Mean
19.	Flow IAT Mean	46.	FIN Flag Cnt	73.	Active Std
20.	Flow IAT Std	47.	SYN Flag Cnt	74.	Active Max
21.	Flow IAT Max	48.	RST Flag Cnt	75.	Active Min
22.	Flow IAT Min	49.	PSH Flag Cnt	76.	Idle Mean
23.	Fwd IAT Tot	50.	ACK Flag Cnt	77.	Idle Std
24.	Fwd IAT Mean	51.	URG Flag Cnt	78.	Idle Max
25.	Fwd IAT Std	52.	CWE Flag Count	79.	Idle Min
26.	Fwd IAT Max	53.	ECE Flag Cnt	80.	Label
27.	Fwd IAT Min	54.	Down/Up Ratio		

The descriptions of some of the features in this dataset that were selected by using the Random Forest algorithm are displayed in Table 2.4.

Table 2.4: The descriptions of some of the features of the CSE-CIC-IDS2018 Dataset

No.	Selected Features	Description
0	Dst Port	Destination Port
1	Tot Fwd Pkts	Total packets in the forward direction
2	Tot Bwd Pkts	Total packets in the backward direction
3	TotLen Fwd Pkts	Total size of packet in forward direction
4	Fwd Pkt Len Max	Maximum size of packet in forward direction
5	Fwd Pkt Len Mean	Mean length of packet in forward direction
6	Fwd Pkt Len Std	Standard deviation size of packet in forward direction
7	Bwd Pkt Len Mean	Mean time between two packets sent in the backward direction
8	Bwd Pkt Len Std	Standard deviation size of packet in backward direction
9	Fwd IAT Std	Standard deviation time two flows
10	Fwd IAT Min	Minimum time between two flows
11	Bwd IAT Min	Minimum time between two packets sent in the backward direction
12	Fwd Header Len	Total bytes used for headers in the forward direction
13	Bwd Header Len	Total bytes used for headers in the forward direction
14	Pkt Size Avg	Average size of packet
15	Fwd Seg Size Avg	Average size observed in the forward direction
16	Bwd Seg Size Avg	Average size observed in the backward direction
17	Subflow Fwd Pkts	The average number of packets in a sub flow in the forward direction
18	Subflow Fwd Byts	The average number of bytes in a sub flow in the forward direction
19	Subflow Bwd Pkts	The average number of packets in a sub flow in the backward direction
20	Init Bwd Win Byts	# of bytes sent in initial window in the backward direction
21	Fwd Act Data Pkts	# of packets with at least 1 byte of TCP data payload in the forward direction

2.9 The Basic Concepts Used in Building The Hybrid Model

1- The Rectified Linear Unit(ReLu) is a widely favored activation function employed in deep learning and RNNs for various compelling reasons:

- Non-linearity: ReLU is instrumental in introducing non-linearity to the model, a critical factor for effectively learning complex patterns and relationships within the data. Without a non-linear activation function like ReLU, a neural network, including RNNs, would essentially reduce to a linear model.
- Simplicity and Efficiency: ReLU is computationally efficient to compute compared to more complex activation functions like sigmoid or tanh. This is because ReLU only requires a simple thresholding operation, which makes it faster to compute during forward and backward passes.
- Addressing the Vanishing Gradient Problem: In traditional RNNs, especially those with many time steps, the vanishing gradient problem can occur. This occurs when gradients during back propagation diminish significantly, rendering it challenging for the network to grasp long-term dependencies. ReLU helps mitigate this problem by preventing the gradients from becoming too small for positive inputs, which can help with the flow of information over many time steps.
- Sparsity and Sparse Activation: ReLU leads to sparse activations. In a typical feedforward neural network or RNN, only a fraction of neurons will be activated at any given time. This can help the network learn more robust features since it's only paying attention to a subset of features.

- Improved Training Speed: Due to its non-saturating nature (meaning it doesn't "saturate" or get stuck for large positive inputs), ReLU can lead to faster convergence during training.
- Mitigation of the Exploding Gradient Problem: Although not as effective as some other techniques, ReLU can help with the exploding gradient problem by bounding the activations. When using ReLU, the output values don't grow too large for positive inputs.

2- **Dropout:** It's a regularization technique frequently applied in deep learning models. It offers several benefits:

- Reduces Overfitting: One of the primary benefits of dropout is that it helps reduce overfitting. Overfitting arises when a model becomes highly proficient at handling the data used for training but struggles to extend this performance to novel, unseen data. Dropout is an approach that by periodically deactivating a portion of neurons while training, prevents overfitting. This compels the network to acquire more robust features.
- Encourages Network Robustness: Dropout encourages each neuron to be less reliant on the presence of specific other neurons. This leads to a network that is more robust and less likely to rely heavily on a small subset of features.
- Improves Generalization: By reducing the dependency on specific neurons, dropout forces the network to learn more diverse and generalized features. This helps the model perform better on unseen data.
- Reduces Sensitivity to Initial Weights: Without dropout, neural networks can become highly sensitive to the initial weights. Small changes in the weights can lead to drastically different

results. Dropout makes the network more resilient to these variations.

- Allows for Larger Deeper Networks: Dropout can enable the training of much larger and deeper networks. It provides a form of regularization that allows you to build bigger models without as much risk of overfitting.
- Mitigates the Need for Fine-Tuning: Dropout can reduce the need for extensive hyperparameter tuning. It provides a degree of robustness to variations in learning rates and other hyperparameters.
- Avoids Co-Adaptation of Neurons: Neurons can sometimes become overly specialized and co-adapt to each other. Dropout helps prevent this by making neurons more independent.
- Computationally Efficient Regularization: Dropout is computationally inexpensive compared to other regularization techniques like L2 or L1 regularization. It can be applied directly during training without significantly increasing the computational load.
- Simple to Implement: Adding dropout to a neural network is straightforward. It involves specifying a dropout rate (usually between 0.2 and 0.5) which determines the probability of a neuron being dropped out during each training step.

3- **The tanh function** : LSTM system frequently use tanh function for a number of compelling reasons:

- Squashing Nonlinearity: The tanh function squashes its input values to be within the range of -1 to 1. This helps in maintaining the values within a bounded range, which can help stabilize learning in deep networks.

- Mitigating the Vanishing Gradient Problem: The tanh function has a steeper gradient compared to the sigmoid function (another commonly used activation function in LSTMs). This can help in mitigating the issue involves the vanishing gradient, which is a common issue in deep networks and particularly important in recurrent neural networks like LSTMs.
- Centered around Zero: The tanh function is zero-centered, meaning it outputs values that are centered around zero. This is in contrast to the sigmoid function which is centered around 0.5. This can make it easier for the network to learn and adapt, especially if the data is also centered around zero.
- Capturing Negative Values: The tanh function allows the LSTM cell to capture negative values. This is crucial for tasks where both positive and negative changes in the hidden state are important for learning.
- Gradient Descent: The tanh function's gradients are well-suited for gradient-based optimization techniques like backpropagation, which are used to train neural networks.
- Historical Context Handling: In LSTMs, it's important to be able to forget information from previous time steps or remember it. The tanh function, by mapping its inputs to the range (-1, 1), helps in achieving this by allowing the LSTM cell to decide which information to remember or forget.

4- **Dense:** A dense layer comprising one neuron with a sigmoid activation function is introduced as the network's output layer. The application of the sigmoid activation function in LSTM units is a common practice, driven by several key reasons:

- Squashing Nonlinearities: The sigmoid function is bounded between 0 and 1, which helps in squashing the input values to a range that is suitable for the gating mechanisms in an LSTM. This is important for controlling the flow of information through the cell state and deciding which parts to update, forget, and output.
- Gating Mechanism: LSTMs use gating units to decide what information to store, forget, and output from the cell state. The sigmoid activation function is essential for these gating units, as it allows them to output values between 0 and 1, indicating the proportion of information to pass through.
- Gradient Preservation: Sigmoid functions have well-defined gradients everywhere, which can help in mitigating the vanishing gradient problem. This is especially important in recurrent neural networks like LSTMs, where the flow of gradients over time can be a challenge.
- Historical Information Control: The sigmoid function helps in controlling how much of the past information required to be able to maintain the cell state. This is achieved through the forget gate, which decides which information to forget and which to retain.
- Non-linearity: Although the sigmoid function is not as powerful as some other activation functions like the Rectified Linear Unit (ReLU) in terms of modeling complex functions, it still introduces non-linearity to the network. This is important for enabling the LSTM to learn and model intricate relationships in the data.

- Output Gating: The sigmoid function is additionally employed in the output gate of the LSTM. This gate plays a crucial role in deciding the extent to which the cell state should be revealed as the cell's output.

5- **The Adam function :** It is renowned for its capacity to modify the rate of learning, is used to create the model. The evaluation metric employed is accuracy. Adam, short for Adaptive Moment Estimation, is a widely used optimization algorithm in deep learning model training. It amalgamates concepts from two other optimization algorithms: RMSprop and Momentum. The benefits of using the Adam optimizer in deep learning:

- Adaptively: Adam adapts the learning rates for each parameter individually, which allows it to adapt to different learning rates for different parameters. This is especially useful in scenarios where some features or parameters may have different scales or sensitivities.
- Efficient Memory Usage: Adam maintains a moving average of both the gradients (momentum) and the second moments (uncentered variance) of the gradients. This helps in efficient utilization of memory compared to methods like SGD with momentum.
- Bias Correction: Adam performs bias correction for the moving averages to ensure that they are properly initialized. This is important especially in the early stages of training when the moving averages might be biased towards zero.
- Regularization Effect: Due to its adaptive nature, Adam can act as a form of L2 regularization. It tends to penalize large weights, making it useful for preventing overfitting.

- Robustness to Noisy Gradients: Adam is less sensitive to noisy gradients compared to other optimization algorithms like vanilla SGD. This makes it more stable and less likely to be stuck in local minima.
- Convergence Speed: Adam often converges faster than traditional stochastic gradient descent algorithms, especially when dealing with large datasets or complex models.
    - Low Memory Requirements: While Adam uses more memory than simple stochastic gradient descent, it typically uses less memory.

2.10 Measures of Model Performance

As a contingency table for assessing a model's performance, Table 2.5 contains a confusion matrix. In the case of probabilistic models, given a probability threshold of z [0, 1], to describe the performance of the proposed system we extracted the associated confusion matrix and, as a result, determine several model performance metrics, including accuracy, recall, precision, Cohen's kappa, ROC AUC, and F1 score.

Table 2.5:A Binary Confusion Matrix

		Predicted Classes	
		Anomaly	**Legitimate**
Actual Classes	**Anomaly**	**True Positive(TP)**	**False Negative(FN)**
	Legitimate	**False Positive(FP)**	**True Negative (TN)**

- TP represents the count of attacks accurately identified as attacks.

- FN indicates the count of attacks erroneously classified as benign records.
- TN denotes the count of benign records accurately classified as benign records.
- FP signifies the count of benign records inaccurately classified as attacks.

To evaluate the classification model. Our dissertation used several performance metrics. These measures are computed in the following manner:

4- Accuracy: The frequency at which a classifier correctly assigns labels to events is known as prediction accuracy, and it is computed as follows:

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \dots (2.11)$$

5- Precision: Is a statistical measure of variability that characterizes random errors, and is computed as follows:

$$precision = \frac{tp}{tp + fp} \dots (2.12)$$

6- Recall: This is the proportion of accurately predicted positive cases out of all positive instances, and it is calculated as follows:

$$recall = \frac{tp}{tp + fn} \dots (2.13)$$

7- F1 score: It embodies both memory and accuracy, amalgamating the precision and recall scores of a model. This criterion is computed as:

$$f1_{score} = \frac{2 \times precision \times recall}{precision + recall} \dots (2.14)$$

8- Cohen's kappa: Is a scalar meter of accuracy where p_0 represents the observed percentage of units when both observers really classify the same objects. The sums of the rows and columns are first required in order to calculate p_e.

$$k = (p_o - p_e) / (1 - p_e) \quad \ldots(2.15)$$

9- ROC AUC: It's frequently used to compare the additional value, if any, of different classifiers. When the AUC is equal to 1, the classifier can effectively distinguish between all positive and negative class values. However, if the AUC were 0, the classifier would have incorrectly identified all positives as negatives and all negatives as positives.

Chapter Three

The Proposed Model Design

3.1 Introduction

Detecting DDoS attacks on blockchains can be challenging due to the distributed and decentralized nature of blockchain networks. DDoS attacks are profitable for the cybercriminal sector in the cryptocurrency ecosystem. It's time to enhance security measures more effectively and ensure that client data is protected. In this dissertation, the focus is on the security challenges of detecting DDoS attacks in network environments that support cryptocurrencies and other types of networks. Three fundamental procedures compose the proposed system:

The first deals with a hybrid deep learning model-based detection, which is demonstrated in section (3.2.4).

The second deals with extracting attack datasets and dividing them depending on clustering approaches, which is demonstrated in section (3.2.5.1).

The third deals with building two new GUI the first one related to the AHC model and the second related to the GMM model. These GUIs were based on real datasets in the cryptocurrency environment and then applied the first proposal that detects DDoS attacks as well as the proposal that extracts data based on cluster algorithms, which is demonstrated in section (3.2.6).

3.2 The General Proposed Model

The suggested model for DDoS attack detection is explained in this chapter using a hybrid deep learning model that combines RNN and LSTM algorithms, also building GUI for DDoS attack monitoring to prevent it. Figure (3.1) details the primary block structure for the suggested system.

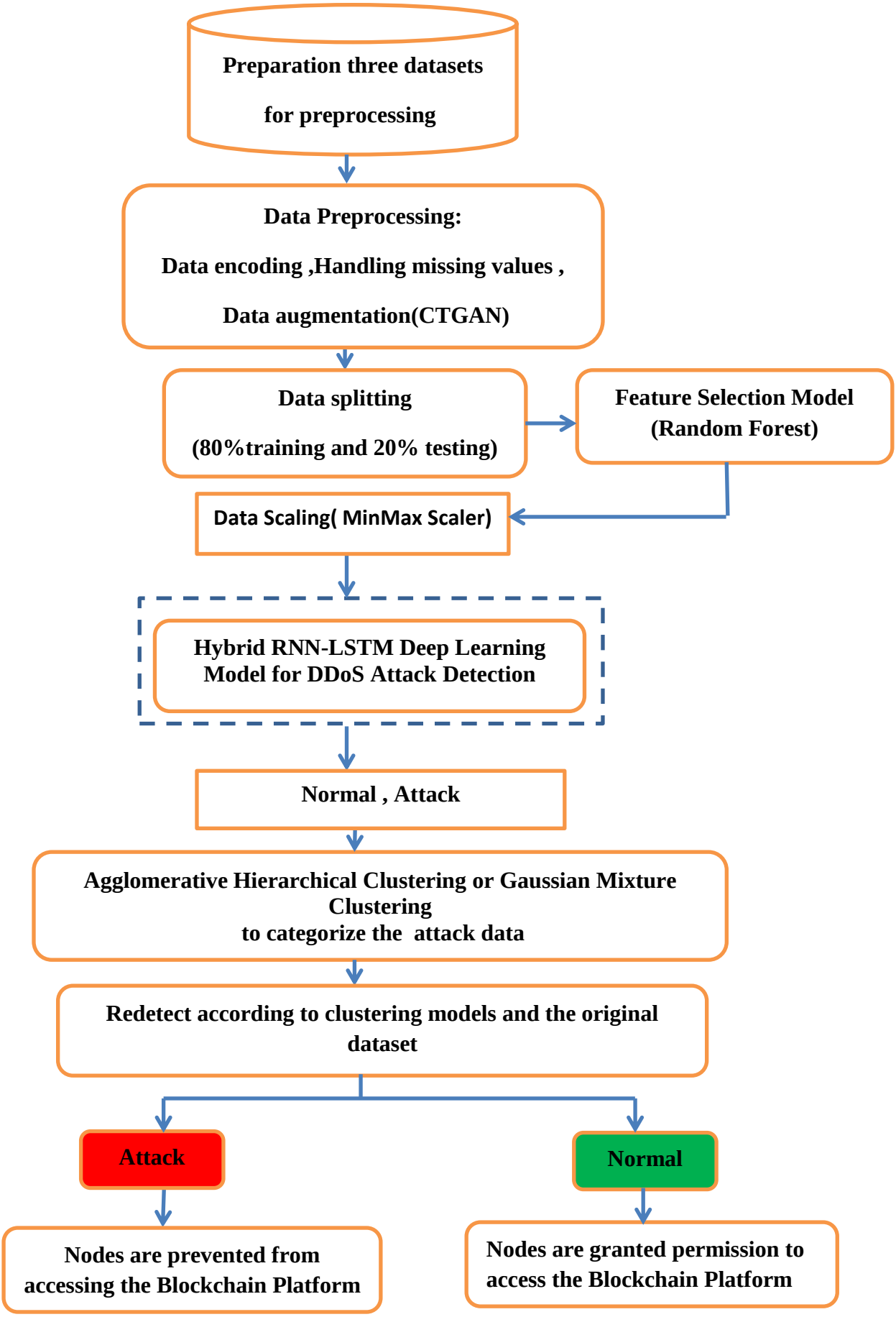


Figure 3.1: The Main Block Diagram of the Proposed System

3.2.1 The Pre-processing

The current dissertation utilizes the Mt.Gox dataset, which contains records of DDoS attacks in the Bitcoin ecosystem. This data comprises nine distinct features and is derived from actual incidents of attacks targeting Bitcoin services with DDoS. Mt.Gox, a major player in Bitcoin transactions, serves as the central focus of this case on DDoS attacks directed at currency exchanges. The dataset encompasses nine features. Furthermore, alternative datasets are employed to validate the proposed model. The dataset samples underwent analysis through the application of the subsequent preprocessing steps:

A. Data Encoding: Initially, the model addressed values that are missing within the data. For the procedure of encoding the data, The LabelEncoder function was used to create uniform labels, which can also translate non numeric labels into numerical values. Values that are missing were handled, and any null features were eliminated.

B. Data scaling: The procedure of scaling enabled the model to better understand and learn from the information. Neural networks, for instance, can encounter issues when working with un scaled independent variables. Standardization and normalization are the prevailing techniques for feature scaling. Scaling in machine learning is a preparation method that narrows the gap between data points, increasing algorithm performance and accelerating machine learning. The scale for the data range is [0, 1] in our model using a min-max scaling procedure, as shown below:

$$Xsc= \frac{X - Xmin}{Xmax - Xmin} \ldots (3.1)$$

In which the values of each feature's lowest and maximum are represented, respectively, by X_{min} and X_{max}. Then, 80% of the data for training while 20% of the data for evaluation and testing.

C. **Data augmentation :** The method of CTGAN was employed to augment the training data for the neural network. Based on the original dataset, this procedure produced additional data. The method was specifically designed for real-world datasets including DDoS attacks on Bitcoin networks. This fundamental approach to expanding the dataset was employed in smaller datasets to mitigate the risk of overfitting. To improve the size of the dataset and deal with the issue of class inequality, This dissertation introduced a new set of data points representing DDoS attacks. This step aimed to preventing the operation of overfitting and boost the predictive performance of the model. The original dataset consisted of 1290 samples, whereas 1000 samples were added to the simulated data points. Thus, the dataset included 1047 samples for DDoS attack data and 1243 samples for benign data.

3.2.2 Data splitting

Dividing data into distinct subsets, known as data splitting, holds considerable importance in data science. In this study, we adopted a two-part split approach. One segment, comprising 20% of the data, was allocated for testing and evaluation, while the remaining portion was utilized for training the model.

3.2.3 The Feature Selection Model

The feature selection procedure in the creation of a predictive model involves reducing the quantity of input data. In some circumstances, the performance of the model can be improved and the computational cost of modeling can be decreased by decreasing the count of input variables. To perform feature selection using the Select From Model class with a Random Forest Classifier. This approach involves

training a random forest classifier on the dataset that is utilized and then using the feature importance from the trained model to select important features for the proposed work. The model creates a random forest classifier with 100 decision trees (estimators). Algorithm (3.1) represents feature selection model.

Algorithm(3.1) Feature Selection Model

Input :Preproceced data, Decision Tree=N
Output: Selected Features Names

1. **Begin**
2. # Initialize feature selection using Random Forest Classifier
   Select= Random Forest (N) using equation(2.10)
3. # Get boolean mask indicating selected features
   Selected features mask
4. # Fit the feature selection model on training data
   (fitting(x_train, y_train)
5. # Get column names of the original dataset
   All features names =data from columns
6. # Create a list of selected feature names
   Selected features names=[ ]
7. For I in range all features
   If selected features mask [i]
   Append selected feature with all feature names
8. Print and store the selected feature names
9. End Algorithm

A technique for ensemble learning is random forest that combines the forecasts from numerous decision trees in order to enhance overall performance and mitigate overfitting. Due to the restricted number of attributes in our dataset related to the Bitcoin services, the model doesn't use the feature Selection model, therefore we used the proposed model on a dataset with several features due to the restricted the dataset's attributes numbers related to the Bitcoin services, the model doesn't use the feature Selection model, therefore we applied the proposed approach on a two

datasets with multiple features such as the CSE-CIC-IDS2018 dataset on AWS and CIC-IDS2017 dataset. As a result, the features selected are identified in the feature selection model as will be shown in Chapter 4 Section 4.3.1.

3.2.4 The Proposed Hybrid Model for DDoS Attack Detection

This dissertation used a hybrid deep learning approach for the detection of these attacks. Building a reliable hybrid model was first step in the hybrid phase, employing a single RNN layer and five LSTM layers. Notably, LSTM represents an advanced form of deep RNN designed specifically to address the challenges associated with vanishing gradients, which often occur when learning long phrase relationships between input variables and target outcomes in artificial neural systems. The input gate, output gate, forget gate, and cell are the main four parts of an LSTM. The gates control the information flow through the cell, which holds information for varied lengths of time. The LSTM architecture consists of interconnected memory blocks in recurrent networks. These memory blocks primarily aim to preserve their state throughout time while governing the flow of information through the application of non-linear gate units . Regrettably, due to the problem of either gradient vanishing or gradient exploding within the RNN architecture, effectively learning from data stored over extended periods can be quite challenging . The suggested solution uses the Keras API to build and train a complicated neural network model utilizing a hybrid deep neural network training methodology. The model incorporates a hybrid structure that integrates simple RNN, multiple LSTM layers, dropout layers, and dense layers, among other extras. The Adam optimizer is used to configure the model, and accuracy is used as the evaluation metric. Figure (3.2) illustrates the hybrid model's layer.

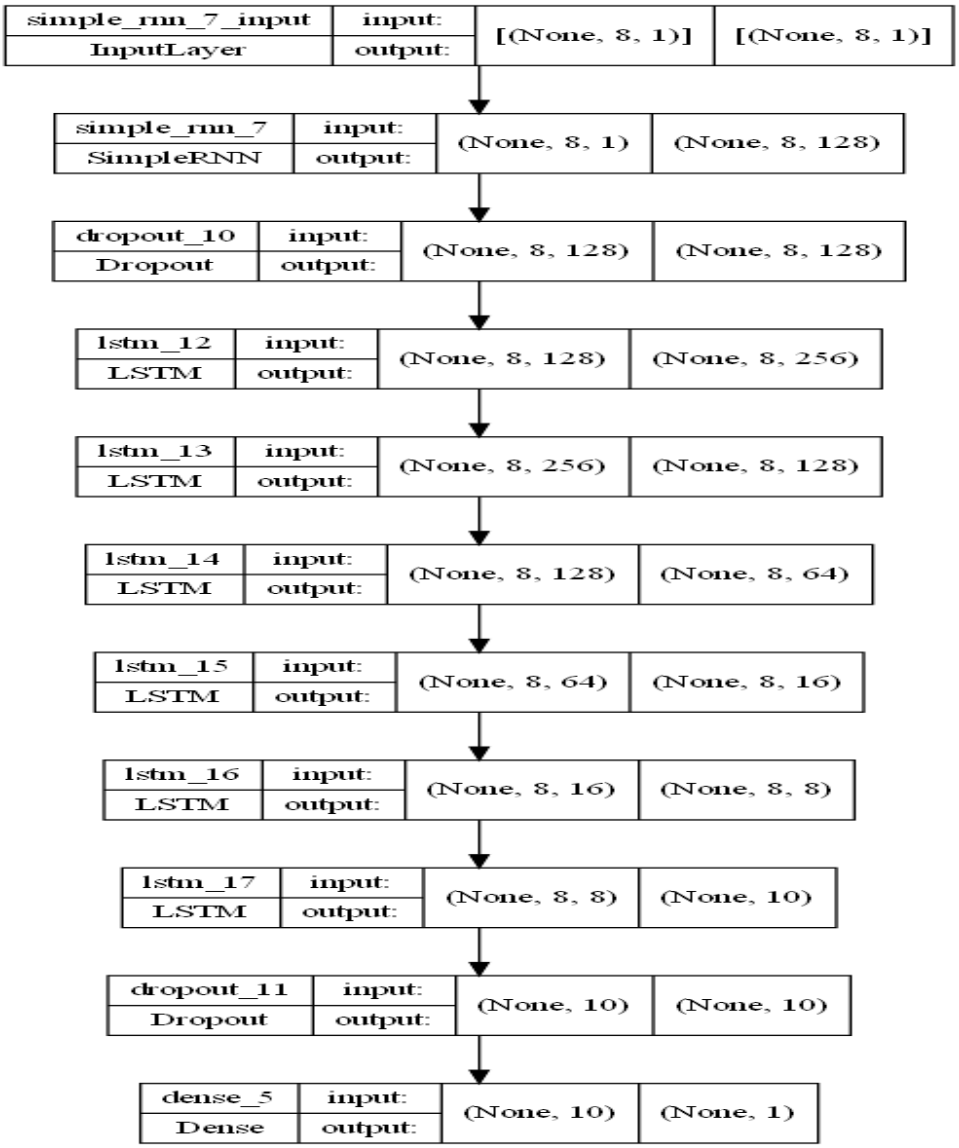


Figure 3.2: The Layers Architecture of the Hybrid Model

The RNN stands out as a powerful deep learning model, particularly adept at handling tasks involving sequential data such recognition of speech and processing of languages. By combining previous inputs into the neural network's internal representation, it gains insights from time series data. Additionally, based on prior and current data, RNN can make predictions about future outcomes. The Hybrid Model is outlined in Algorithm (3.2).

Algorithm(3.2) Hybrid Deep Learning Model(RNN+LSTM)

Input :Training data
Output: Classified Model (Benign or Attack)

1. **Begin**
2. Batch size = N
3. epochs = K
4. Sequential( )
5. **# Adding Simple RNN layer**
   Adding Simple RNN(units=128),(activation function='relu')
   Return sequences=True
   Adding(Dropout(0.2))
6. **# Adding LSTM layers**
   Adding LSTM layer (units=256), ('tanh')
   Adding LSTM layer (units=128),('tanh')
   Adding LSTM layer (units=64),('tanh')
   Adding LSTM layer (units=16), ('tanh')
   Adding LSTM layer (units=8), ('tanh')
   Return sequences=True
   Adding LSTM layer (units=10), ('tanh')
   Return sequences=False
7. Adding Dropout(0.2)
   # Adding Dense layer
8. Adding Dense(1, activation function ='sigmoid')
   # Compiling the model
9. Compiling(optimizer function='adam'),(loss='mse'), (metrics='accuracy')
   # Displaying model summary
10. summary( )

 # Training the model

11. history = fitting(train data, N, K) , validation data=(test data)

 # Saving the model
Save(model)
END

The fundamental stages of the approach's architecture can be outlined in the following manner:

1- The Sequential: This establishes a sequential stack of layers, enabling us to append layers sequentially. The Sequential API in libraries like Keras provides a simple and intuitive way to build neural

networks. It allows you to create models by adding one layer at a time. It allows to quickly put together different layers and architectures, making it great for prototyping and experimenting with different model configurations. Each layer is added in sequence, and the data automatically flows from one layer to the next.

2- Simple RNN: The initial layer is a Simple RNN layer featuring 128 units, set to return sequences, and employing the activation function called ReLU.

3- Dropout: By randomly deactivating some of the neurons during training, overfitting can be prevented by using the layer of dropout with average of 0.2.

4- The Multiple of LSTM Layers: Sequential layering of multiple LSTM layers using the activation function of the hyperbolic tangent, each with a unique number of units. These layers have been specifically aimed in capturing complicated sequential data structures.

5- Dense: A dense layer comprising one neuron with a sigmoid activation function is introduced as the network's output layer.

6- The Model of Compilation: The Adam function, it is renowned for its capacity to modify the rate of learning, is used to create the model. The evaluation metric employed is accuracy.

7- The Summary Model: On the console, the summary approach is used to illustrate the architecture of the model.

8- The Training Model: The fit function is used for training the model. Training data, batch size, epochs, and validation data are all provided as inputs. The number of batches processed per epoch is determined by the steps per epoch variable. The "fit" function enables the specification of the number of epochs, which signifies how many times the entire dataset is processed through the neural network, both forward and backward. This is important because it allows the model to

learn from the data over multiple passes. The fit function typically provides options for specifying validation data. This allows you to monitor how well the model is generalizing to data it hasn't seen during training. It's important for avoiding overfitting. It abstracts away much of the complexity of training a neural network. It provides a high-level interface for training models, which makes it accessible even to those who might not have a deep understanding of the underlying mathematics. Overall, using the fit function simplifies the process of training deep learning models and provides a standardized way to implement training loops.

3.2.5 The Clustering Model

This section will clarify the proposed model of extracting the attack data from the data set that obtained it from training the model. After that, it will discuss the use of clustering algorithms to divide this data for ease of dealing with it and its later use in the stage of building the GUI, comparing data, and checking for attack or normal nodes.

3.2.5.1 Proposed Extracting Attack Dataset Model

This section performs an attack prediction task using a machine-learning model by using a trained neural network model (loaded from a saved Keras model file) and a scalar for preprocessing the data. The goal of this step is to classify records as either attacks or Identify normal instances according to the model's predictions, and subsequently, save the instances classified as attacks by the model into a distinct CSV file. The following steps illustrate the work of the proposed model in the attack data extraction process.

- Read the dataset after augmentation. This dataset contains features and labels. .
- Read the trained hybrid model.

- Loads a scaler using the pickle module. This scaler is used to transform the features in a standardized manner. Feature scaling is a method employed to normalize the independent features in the data to a predetermined range. It is carried out as part of the data preprocessing to deal with significantly varying magnitudes, values, or units whereas Pickle is a method for converting a Python object structure into a byte stream.
- The most important step is to loop through the dataset and use a hybrid model to predict each row. If the outcome is an attack take the row information and add it to a new dataset.
- If the prediction's value is equal or greater than to 0.5, the record is considered an attack and is appended to the attack_list. Otherwise, the record is considered normal and is appended to the error_list.
- Saving the new dataset completes the process. This data will be used in the stage of clustering. Algorithm (3.3) represents the Proposed Extracting Attack Dataset Model.

Algorithm(3.3) Proposed Extracting Attack Dataset Model

Input : Augmented Dataset
Output: DDoS attack data

1. **Begin**
2. **# Reading label encoded dataset**
   df = Read augmented dataset
3. **# Read the scaler**
   scaler = Load (scaler.pkl)
4. **# Remove the normal data**
   df = Filter Rows(df)
5. **# Remove the label column**
   df.Drop Column(df.Last Column)
6. **# Apply scaler on dataset**
   dfscaled = Transform(df, scaler)
7. **# Load the model**
   model = Load Model(model)
8. **# Create empty lists for attack result**
   attack_list = []
   error_list = []

```
9.    # Loop on dataset and predict
        FOR record IN dfscaled:
          df1 = Create DataFrame([record])
10.   # Predict on each record
          prediction = model.Predict(df1)

11.   # Check prediction output
          IF prediction >= 0.5 THEN
            AppendToList(attack_list, record)
          ELSE
            AppendToList(error_list, record)
          END IF
        END FOR

12.   Print("attack list", Length(attack_list))
        Print("normal_list", Length(error_list))
13.   # Save the records that predict attack in new dataset
        attackdf = CreateDataFrame(attack_list)
        attackdf.SetColumns(df.Columns)
14.   END
```

3.2.5.2 Agglomerative Hierarchical Clustering Model

Agglomerative clustering is a hierarchical clustering algorithm that starts with individual data points as clusters and then progressively merges them to form larger clusters. This algorithm works to construct a dendrogram by iteratively merging clusters based on a chosen distance metric. The process starts with individual data points as clusters, and in each iteration, the two clusters with the closest proximity are combined, resulting in the creation of a larger cluster. This procedure continues until there are three clusters that contain all the data points. The main steps of this procedure are illustrated as follows:

- The first step of this model is reading the dataset obtained from section (3.3.1) that represents extracting attacks dataset.
- Then, create an Agglomerative Clustering model with number of clusters set to 3.

- Utilizing the Euclidean distance formula (2.12) to compute the distance between instances.
- After that fit the model to the data and obtain the cluster labels for each data point.
- Create the bar plot to visualize the data points colored according to their assigned clusters.
- Save the model to file for use in the section of the building platform.
- Finally, perform hierarchical clustering using 'ward' linkage according to equation (2.11). Algorithm(3.4) represents the AHC.

Algorithm(3.4) The Agglomerative Hierarchical Clustering (AHC)

Input : Extracting Attacks Dataset, Number of Clusters
Output: Selected Clusters

```
1.  Begin
2.   # Commence by designating each data point as an individual
     cluster.   clusters = [datapoint for datapoint in data]
3.      while length (clusters ) > 1:
           # Identify the pair of clusters that are nearest to each other.
           Closest clusters, min distance = find closest
     clusters(clusters)
4.   # Merge the two closest clusters
           Merged cluster = clusters merging (closest clusters)
5.      # Remove the combined clusters from the list
         Clusters remove(closest clusters 0 )
           Clusters remove(closest clusters1 )
6.      # Incorporate the combined cluster into the list
           Clusters appending (merged cluster)
7.      # Return the final cluster hierarchy
        return clusters[0]
8.   Find closest clusters(clusters):
        Min distance = infinity
        Closest clusters = No value assigned.
9.   For  k in range(length clusters):
           for j in range(k+1, len clusters ):
              distance = calculate distance(clusters[k], clusters[s])
              if distance < minimum distance:
```

```
                    minimum distance = distance
                    closest clusters = (clusters[k], clusters[s])
        return closest clusters, minimum distance
10.     Calculate distance(cluster1, cluster2): using equation (2.12)

11.     Merge clusters(cluster1, cluster2):
          # Merge two clusters into a single cluster
          Merged cluster = cluster1 + cluster2
          return merged cluster
12.     End
```

3.2.5.3 Gaussian Mixture Clustering Model

Performing clustering by using the GMM on a dataset and saving both the cluster visualization and the trained model. The main steps of this procedure are illustrated as follows:

- Reading the Dataset: The first step of this model is reading the dataset obtained from section (3.3.1) that represents extracting attacks dataset.
- Creating Clusters: using the Gaussian Mixture class to create a clustering model with 2 clusters.
- Fitting the Model: Fitting the clustering model to our data using .”fit_predict()” function and store the resulting cluster assignments in the clusters variable.
- Creating the Cluster Visualization: Create a bar plot to visualize the counts of data points in each cluster. This gives a basic idea of how the data is distributed among the two clusters.
- Saving the Visualization: saving the generated plot as an image.
- Saving the model: using the “dump()” function from the pickle module to save the clustering model to a file.

3.2.6 The Proposed GUI Model

Security is an ongoing process, and the threat landscape evolves. Therefore, the GUI model should be flexible, allowing for updates and improvements as new security challenges arise. Creating an intelligent

security platform to protect network services in a blockchain environment is a complex endeavor, but it's feasible. Such GUI would need to address various security challenges in blockchain networks, such as data integrity, confidentiality, consensus protocol vulnerabilities, and smart contract vulnerabilities. This dissertation proposed to create a GUI model to monitor network traffic and identify unusual patterns or behaviors. This proposal will implement intrusion prevention mechanisms to automatically block or mitigate suspicious activities. Two GUI models have been constructed, one using the AHC algorithm and the other the GMM algorithm for DDOS attack detection on cryptocurrency network services. The application consists of three tabs: "Configuration," "Nodes Information," and "Results.

- Tab 1 - Configuration: This tab allows to configure various aspects of the application. The user can specify the number of nodes in the blockchain platform and select files for the scaler , label encoder, classification model, clustering model, and dataset. There's also a button to load the configuration.
- Tab 2 - Nodes Information: This tab displays a list of nodes along with their information. The user can select a node from the list, and its information will be displayed below the list.
- Tab 3 - Results: This tab displays a scrollable text box where logs and results are displayed. The text box is disabled to prevent user input.

  The processing of nodes and their conditions in a blockchain-related scenario, with a focus on determining whether nodes should be blocked or not based on certain conditions and predictions, involves several steps when we apply the model that uses the GMM algorithm. A loop iterates over a list of nodes, and for each node, the following steps are performed :

a. If the node's "attack" condition is not met (if the node is not marked as an attack), then the following actions are taken:

- A random value is generated for the node from an "allUnique" source. The 'allUnique' is considered a dataset that was extracted from the original dataset by selecting the unique values from the set of data values.
- Preprocessing is applied to the node's data using a scaler and an encoder. The scaler considered the file of the scalar model for further analysis.
- A prediction is made using a classification model.

b. If the node's "attack" condition becomes true after prediction, it means the node's behavior indicates an attack. In this case, the following actions occur:

- A log message is printed with the result of the hybrid model's prediction for the node.
- A prediction is made using a clustering model.

c. After the second prediction (using clustering model), if the node's "cluster" value is equal to 1, the node is considered to be blocked:

- The appearance of the corresponding item in a list box is changed to a red background color.
- A log message is printed indicating that the node is blocked.

d. If the node's "cluster" value is not equal to one, it means the node is not fully malicious and is temporarily blocked for 5 transactions:

- The appearance of the corresponding item in a list box is changed to an orange background color.
- A log message is printed indicating that the node is blocked for 5 transactions.

e.If the node's "attack" condition is not met after prediction from model1, it means the node is not malicious:

- The appearance of the corresponding item in a list box is changed to a green background color.
- If a node's "attack" condition was initially true, a separate set of actions is taken. If the node's "cluster" value is 0, indicating that it's not part of a malicious cluster, the remaining number of transactions for which the node should remain blocked is decremented. Log messages are printed to reflect this status. If the remaining count reaches zero, the node's "attack" condition is set to False, meaning the node is no longer considered an attacker.

3.3 The Summary

- As a result, this dissertation proposes a hybrid approach to DDoS attack detection. It is a combination of the RNN and LSTM techniques. The advantage of the proposed strategy is that it is quick and highly accurate. The result of this idea is to safeguard the cryptocurrency network services from DDoS flooding fraud, which causes complete network failure.
- Building an effective DDoS detection system using a hybrid RNN-LSTM model requires careful model selection, extensive training on representative data, and thorough evaluation. Additionally, ongoing monitoring and updates are crucial to ensure the model remains effective in the face of evolving attack techniques.
- Designing an effective hybrid deep learning model for DDoS detection requires careful consideration of the individual models used, the integration strategy, and extensive testing and validation. Additionally, ongoing monitoring and updates to the model are crucial to adapt to new attack techniques and maintain high accuracy levels.

- The purpose of GUI model construction is related to detecting and preventing DDoS attacks on a blockchain platform. The application provides a user interface for configuring and managing nodes within the blockchain platform, as well as displaying the results of various model predictions and operations. The model provides a UI that allows users to configure and manage nodes within the blockchain platform. This could involve tasks such as adding new nodes, removing nodes, adjusting configurations, and monitoring the status and performance of existing nodes.
- Effective node management is crucial for ensuring the stability and reliability of a blockchain network depending on hybrid deep learning model and clustering model. This suggests that the platform may incorporate some form of predictive modeling, likely related to the blockchain's operations. This could include algorithms that predict various aspects of blockchain behavior, such as potential security vulnerabilities.

Chapter Four

The Results Discussion and Analysis

4.1 Introduction

This chapter presents the results of a proposal for DDoS attack detection using a hybrid deep learning model and attack prevention through a GUI model. Additionally, it explains the methodology introduced in chapter three.

4.2 The Execution Environment Requirements

The proposed system is implemented on an Intel Core i7 system. The system is furnished with 16 gigabytes of RAM and SSD with a capacity of 1 terabyte running the Windows operating system. It utilizes the Sci-Kit Learn libraries to implement machine learning and deep learning techniques in Python. The GUI for demonstrating the attack detection model and the clustering model, as well as preventing DDoS attacks from accessing cryptocurrency exchange platforms, was created using the Python programming language.

4.3 The Supervised ML-based DDOS Attacks Classification

Using supervised machine learning, it is possible to categorize DDoS attacks by analyzing diverse attributes and traits of network traffic. When conducting such classification, the outcomes may differ based on various factors, including the dataset used, the choice of ML algorithm, feature engineering, and the evaluation metrics applied. In our dissertation, we introduced a Hybrid Deep Learning Model that combines RNN and LSTM algorithms. The proposed system was trained using a dataset related to the Bitcoin network services.

4.3.1 The Experimental Results

The model encompasses various deep learning techniques, particularly on RNN and LSTM. Additionally, the implementation of the suggested hybrid approach, denoted as RNN-LSTM. To demonstrate the effectiveness of this hybrid approach in detecting DDoS attacks, a confusion matrix was utilized to showcase its robustness, particularly in

terms of sensitivity in the detection of DDoS attacks. In attack detection, TP and FN are pivotal metrics. TP represents the count of accurately classified attacks, while FN stands for the count of attacks erroneously categorized as benign records. Moreover, TN denotes the count of correctly classified benign records, while FP represents the count of benign records mistakenly classified as attacks. The process of feature selection assumes a crucial role in both machine learning and data analysis, involving the identification of a subset of relevant features (variables or attributes) from the original set within the dataset. The hybrid model was applied to multiple datasets, including CSE-CIC-IDS2018 Dataset and CIC-IDS2017 Dataset. Due to these data sets containing a huge number of data and many features, the feature selection model was applied to them to select a specific number of features before conducting training. Our proposal used three case study :

The 1st Dataset

The Mt.Gox dataset contains 9 features The experimental results show that the Mt.Gox dataset hybrid model in this work can identify DDoS attacks with an average accuracy of 95.84%. Figures 4.1–4.3, respectively depict the results of confusion matrix for the suggested hybrid approach, which incorporates (RNN-LSTM), as well as individual RNN and LSTM approches. As the confusion matrix for the hybrid model shows, the correct prediction for the attack data is 174, while for the normal data it is 247. The Confusion Matrix for the model, which is based on the RNN algorithm, showed that the correct number for normal cases was 232, while the number of data points for attacks was 190. Finally, the confusion matrix for the model based on the LSTM algorithm showed that the correct prediction for the normal data was 242, while the number of attack data points was 168. Table 4.1 illustrates the features of Mt.Gox dataset.

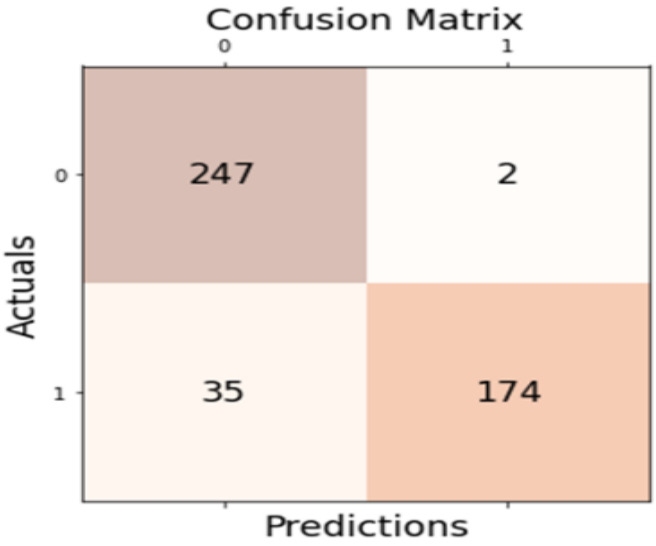


Figure 4.1: Confusion matrix for the suggested hybrid model (RNN-LSTM)

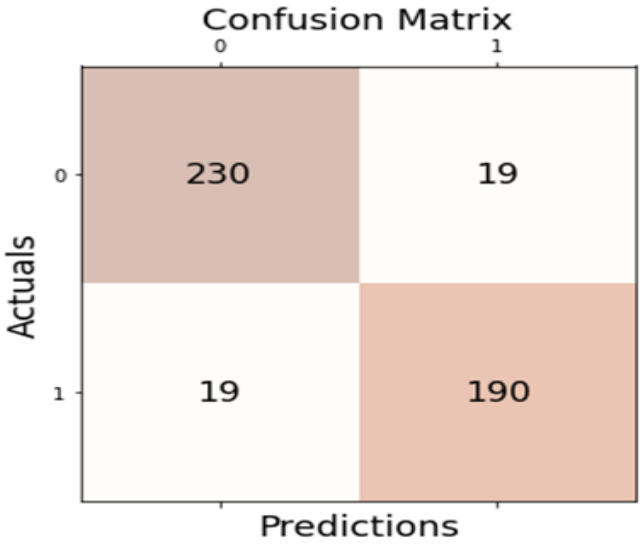


Figure 4.2: Confusion matrix of the RNN using Mt.Gox dataset

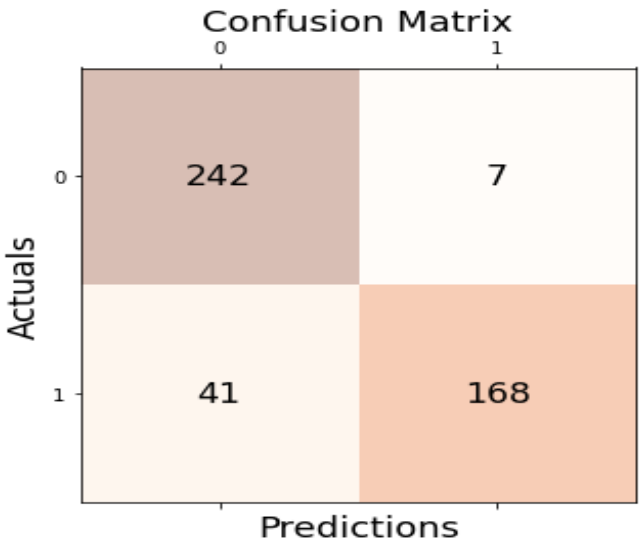


Figure 4.3: Confusion matrix of the LSTM using Mt.Gox dataset

Table 4.1 presents the outcomes achieved through the utilization of various deep learning approaches, including RNN, LSTM, and the hybrid deep learning model. Figure 4.4 displays the graphical representation of the performance evaluation findings, showcasing the proposed hybrid model's performance in comparison to LSTM and RNN algorithms.

Table 4.1: Comparison of outcomes for the detection of DDoS attack using various deep learning approaches

Performance Metrics	RNN Model	LSTM Model	Suggested Hybrid Model (RNN-LSTM)
Accuracy	0.919214	0.926100	**0.958400**
Precision	0.988636	0.960000	0.988827
Recall	0.832536	0.803828	0.846890
F1 score	0.903896	0.875000	0.912371
Cohen’s kappa	0.835095	0.785985	0.848642
ROC AUC	0.912252	0.887858	0.916393

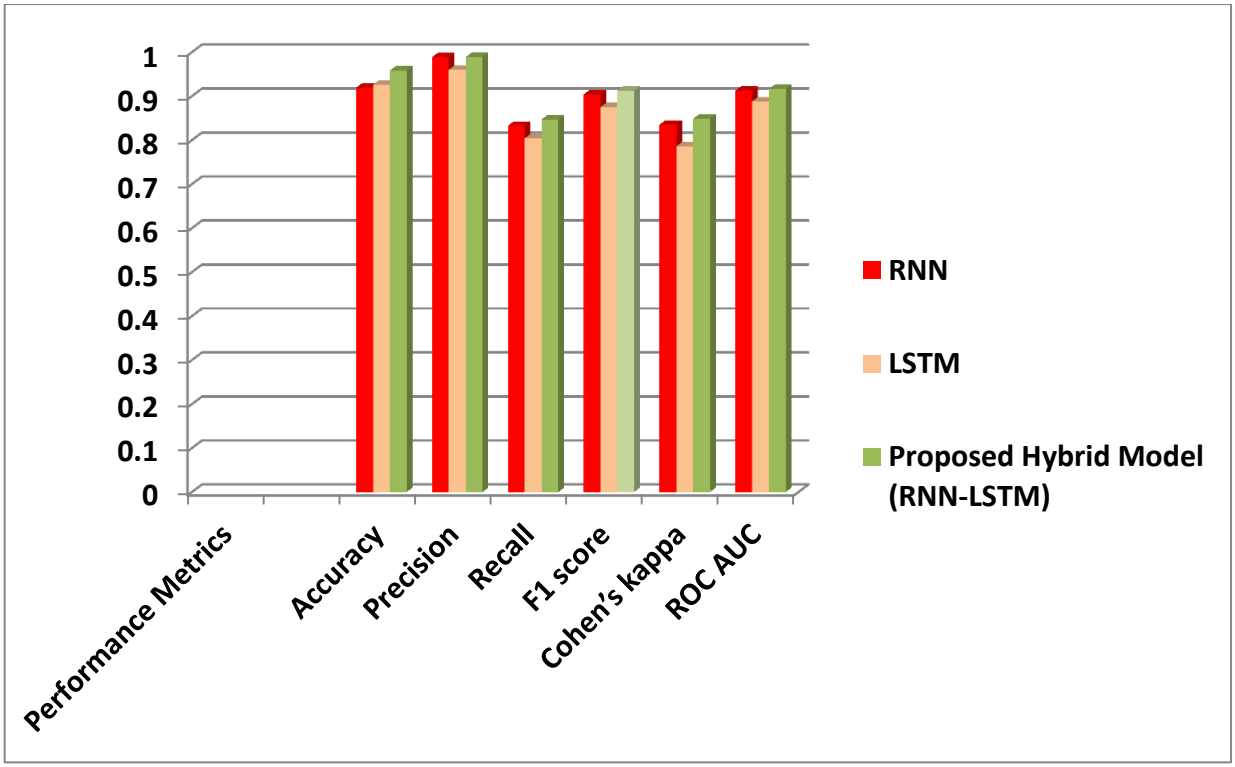


Figure 4.4:The performance evaluation graph for hybrid model in comparison to the LSTM and RNN algorithms

Ultimately, Figure 4.5 displays the results of accuracy of the suggested hybrid approach, while Figures 4.6 and 4.7 depict the results of accuracy of the RNN and LSTM approaches, respectively. Accuracy measures the overall performance on the entire dataset, while validation accuracy specifically measures performance on a separate validation dataset that the model has not seen during training. Accuracy serves as a metric for assessing the overall performance of a classification model, calculated as the ratio of correct predictions to the total number of predictions made. On the other hand, validation accuracy is a metric employed during the training of a machine learning approach. It is the considered accuracy of the model on a validation dataset. The validation dataset constitutes a distinct portion of the data that remains untouched during training but is employed to assess the model's performance throughout the training process. Vigilantly monitoring validation

accuracy is vital to prevent overfitting. A situation where training accuracy is high but validation accuracy is low suggests that the model is too tailored to the training data and may struggle to generalize to fresh, unseen data. Notably, the hybrid model demonstrated superior performance compared to the other algorithms.

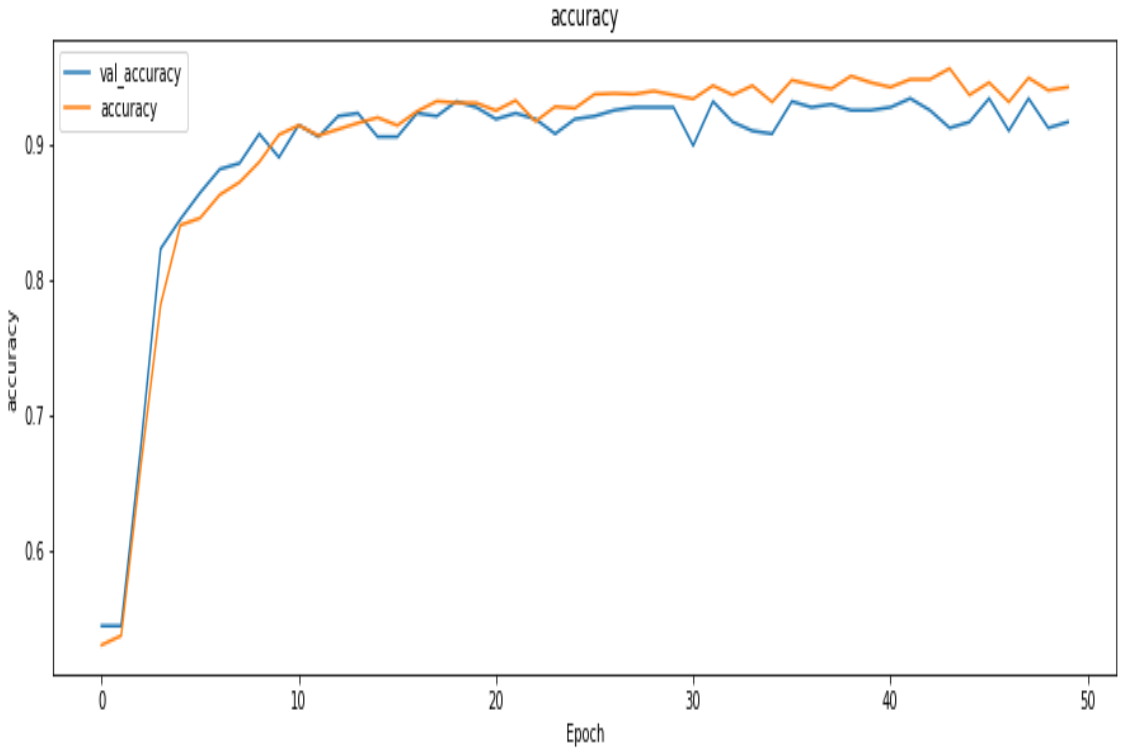


Figure 4.5: The accuracy of the suggested hybrid approach using Mt.Gox dataset

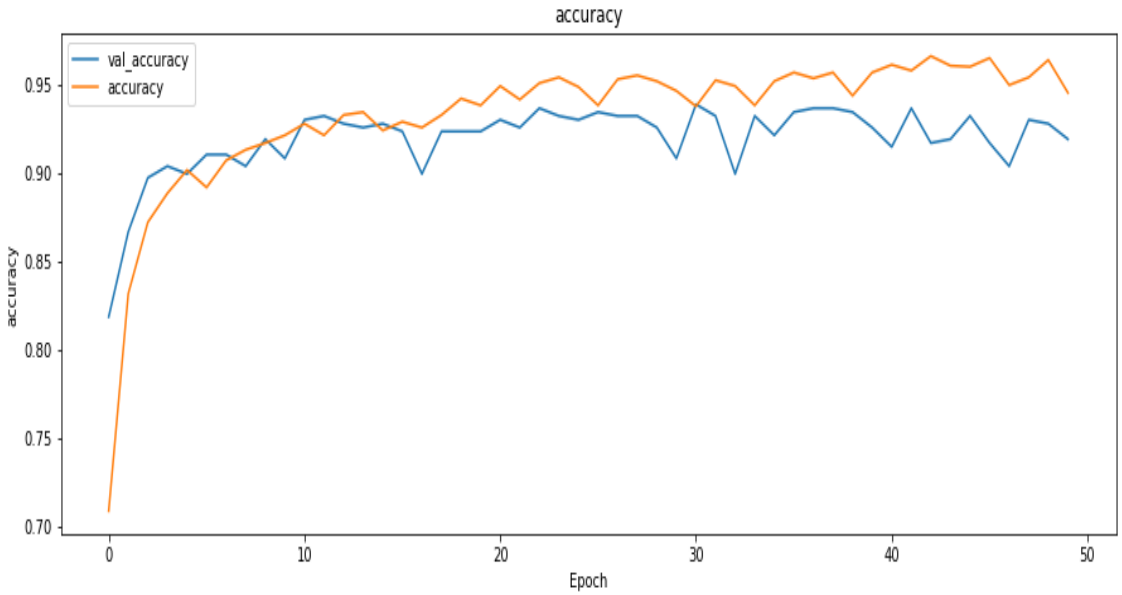


Figure 4.6: The accuracy of the RNN using Mt.Gox dataset

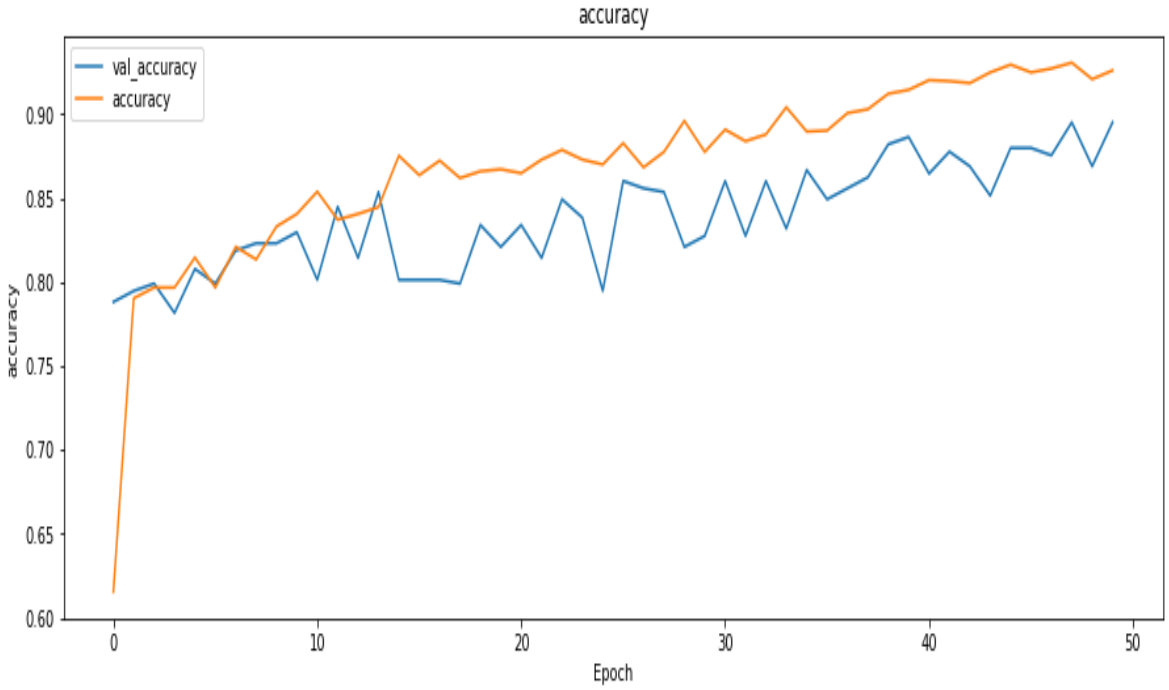


Figure 4.7: The accuracy of the LSTM using Mt.Gox dataset

The 2nd Dataset

CIC-IDS2017 Dataset contains 79 features as showed in Table (3.2). The size of the data is 225745 contains ('Benign': 97718, ' DDoS ': 128027). The system applied an RF algorithm for feature selection. However, the result of our feature selection proposal model has 18 features as indicated in Table 4.2.

Table 4.2: The selected features of the CIC-IDS2017 Dataset

No.	Selected Features
0	Destination Port
1	Total Fwd Packets
2	Total Length of Fwd Packets
3	Fwd Packet Length Max
4	Fwd Packet Length Mean
5	Fwd Packet Length Std
6	Bwd Packet Length Std
7	Fwd IAT Total
8	Fwd IAT Mean
9	Fwd IAT Std
10	Fwd IAT Max

11	Fwd Header Length
12	Avg Fwd Segment Size
13	Fwd Header Length
14	Subflow Fwd Packets
15	Subflow Fwd Bytes
16	Subflow Bwd Bytes
17	Init_Win_bytes_forward
18	act_data_pkt_fwd

The confusion matrix for the suggested hybrid model for CIC-IDS2017 Dataset is shown in Figures 4.8. Where the results showed that the number of attack data was 24595, while the prediction result for the normal data was 18472.

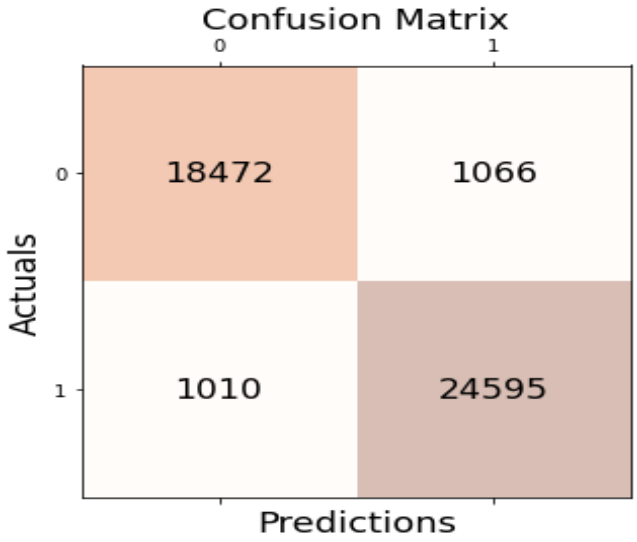


Figure 4.8: The confusion matrix for the suggested hybrid model for CIC-IDS2017 Dataset

The experimental findings indicate that the CICIDS2017 dataset hybrid model in this work can identify DDoS attacks with an average accuracy of 95.40%. Where batch size=10000, epochs = 10. The model calculates various evaluation metrics using the trained model. The table (4.3) displays metrics for evaluating performance including precision, accuracy , F1-score , recall, , cohen's Kappa, and ROC AUC.

Table 4.3: The outcomes for the detection of DDoS attack under CIC-IDS2017 Dataset using proposed hybrid model

Performance Metrics	Hybrid Approach (RNN-LSTM)
Accuracy	0.954013
Precision	0.958458
Recall	0.960555
F1 score	0.959505
Cohen's kappa	0.906302
ROC AUC	0.952997

The 3rd Dataset

CSE-CIC-IDS2018 Dataset contains 80 features as showed in Table (3.1). The size of the data is 1048575 contains ('Benign': 360833, 'Attack': 687742). After executing several steps, involving data preprocessing and feature selection, and model building hybrid model combining both SimpleRNN and LSTM layers, the model was applied to train and evaluate performance using various metrics. However, the result of our feature selection model has 18 features as illustrates in Table 4.4.

Table 4.4: The chosen attributes within the CSE-CIC-IDS2018 Dataset

No.	Selected Features
0	Dst Port
1	Tot Fwd Pkts
2	Tot Bwd Pkts
3	TotLen Fwd Pkts
4	Fwd Pkt Len Max
5	Fwd Pkt Len Mean
6	Fwd Pkt Len Std
7	Bwd Pkt Len Mean
8	Bwd Pkt Len Std
9	Fwd IAT Std

No.	Selected Features
10	Fwd IAT Min
11	Bwd IAT Min
12	Fwd Header Len
13	Bwd Header Len
14	Pkt Size Avg
15	Fwd Seg Size Avg
16	Bwd Seg Size Avg
17	Subflow Fwd Pkts
18	Subflow Fwd Byts
19	Subflow Bwd Pkts
20	Init Bwd Win Byts
21	Fwd Act Data Pkts

Figure 4.9 illustrates the results of confusion matrix for the suggested hybrid model (RNN-LSTM) applied to the CSE-CIC-IDS2018 Dataset. The results indicate that the number of attack data points was 71704, while the prediction result for normal data was 137531.

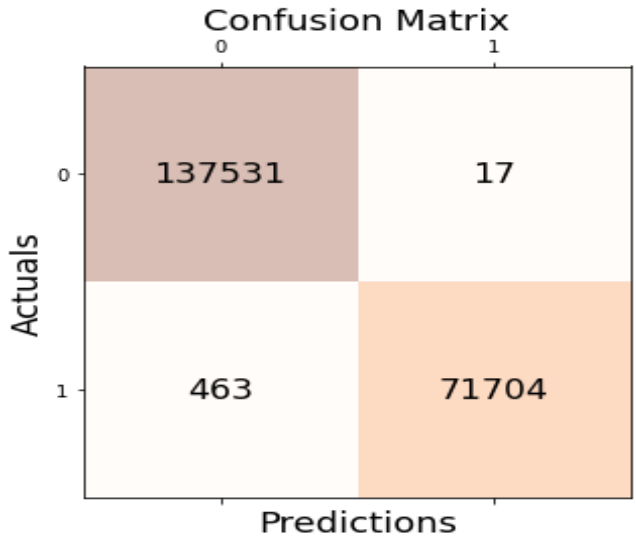


Figure 4.9: The results of confusion matrix for the suggested hybrid approach (RNN-LSTM) applied to the CSE-CIC-IDS2018 Dataset

The empirical findings demonstrate that the CSE-CIC-IDS2018 dataset hybrid approach in this work can identify DDoS attacks with an average accuracy of 99.77%.Where batch size=10000, epochs = 10 as illustrated in Table 4.5.

Table 4.5: The outcomes for detecting DDoS attacks in the (CSE-CIC-IDS2018) Dataset using the suggested hybrid model

Performance Metrics	Proposed Hybrid Model (RNN-LSTM)
Accuracy	0.997711
Precision	0.999763
Recall	0.993584
F1 score	0.996664
Cohen's kappa	0.994922
ROC AUC	0.996730

4.3.2 The Comparison of the Proposed Hybrid System Against Related Works

The current dissertation attains a notable level of accuracy, approximately 95%, when compared to other empirical investigations using the first case study dataset. Table 4.6 provides a comparison of the accuracy in DDoS attacks detection between this proposed model and previous empirical. Achieving a high level of accuracy in DDoS attacks detection is a significant accomplishment, especially when compared to previous experimental studies. It illustrates the efficacy of the proposed model in this particular case study dataset where the data in this dataset is complex and challenging due to its relation to cryptocurrency services, specifically Bitcoin.

Table 4.6: Evaluating the effectiveness of the suggested approach for detecting DDoS attacks on Bitcoin, in comparison to other empirical investigations utilizing the Mt.Gox dataset

Ref.	Published year	The Detection Approaches	The Accuracy
[13]	2014	The approach of Word based classifier	75%
[10]	2019	The approach of Multi-Layer Perceptron(MLP)	50%
The Suggested Hybrid Approach		**RNN-LSTM**	**95%**

4.4.The Unsupervised ML-based DDOS Attacks Clustering

Clustering can be used in combination with other machine learning techniques in an ensemble approach, which can enhance the overall effectiveness of DDoS detection. Clustering can provide visual representations of network traffic data, making it easier for analysts to interpret and understand the data.

There were 945 attack cases after acquiring data on the attacks from the dataset model outlined in section 3.3.1. Following this, two clustering algorithms GMM and AHC were implemented. The results obtained were from the AHC clustering model, as illustrated in Figure 4.10. The model produced three clusters: the first containing 495 attack cases, the second containing 278 attack cases, and the third containing 181 attack cases. The results of these clusters were utilized in the stage of constructing the platforms for testing and simulating the nodes that will join the blockchain network, determining whether they are attack or normal.

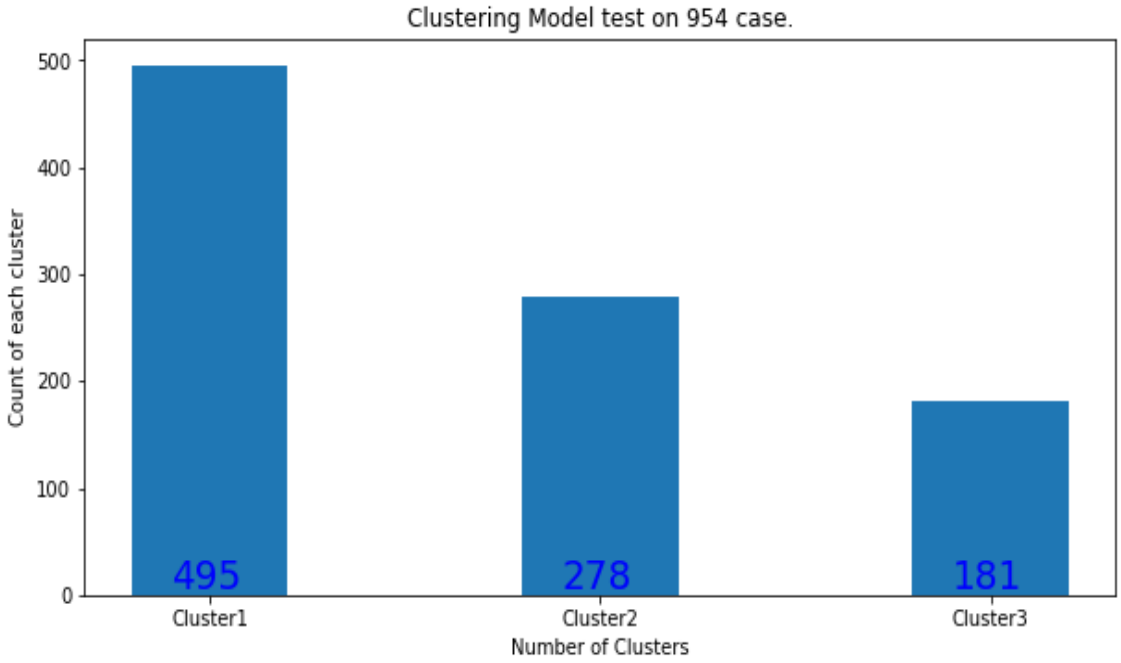


Figure 4.10: The AHC Results

The results were obtained from the clustering model of the GMM method illustrated in Figure 4.11. The first cluster comprises 484 instances of attacks, while the second cluster encompasses 470 attack cases.

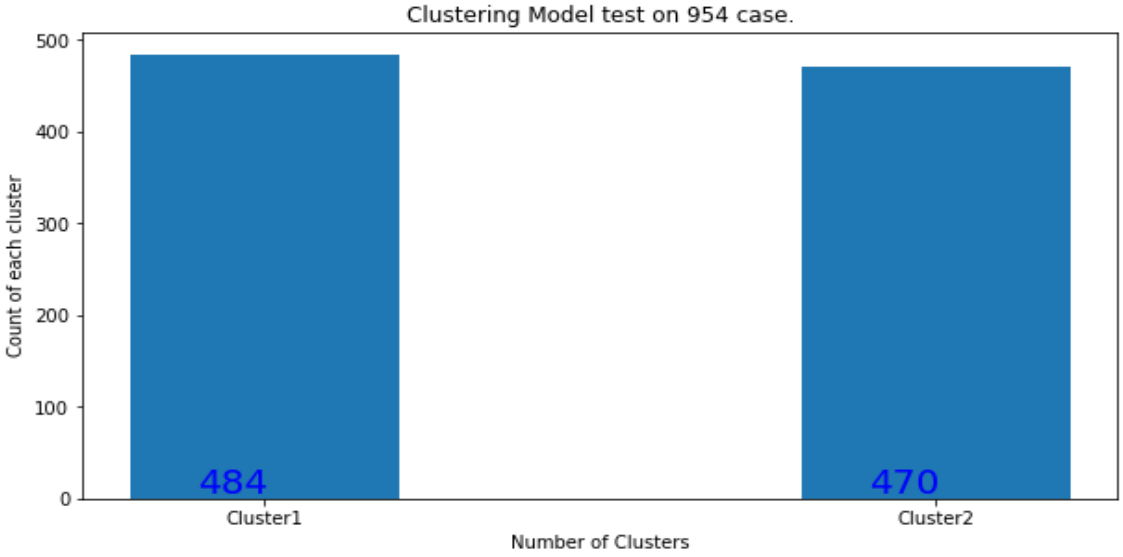


Figure 4.11: The GMM Results

4.5 The GUI Environment Results

As mentioned in Section 3.2.6, GUI model proposal was discussed. Two types of models were built, the first using the AHC algorithm, and the second using the GMM algorithm. These models were built in the Python language for the purpose of simulating the proposed system to detect DDoS attacks, as well as applying the clustering model for the purpose of determining the type of nodes, whether they are attacked or normal. If the node is an attack, it is prevented from joining the blockchain network; but, if it is normal, it will be permitted to join the network in order to conduct cryptocurrency exchanges. Simulation model illustrated in Figure 4.12.This model contains three tabs: Configuration, Nodes information, and Results. Configuration tab contains several textboxes as follows:

1- Number of nodes in Platform: represents the number of nodes that will enter this platform to be processed and is determined
2- Scalar : It contains a scalar that includes the data obtained after the scaling process
3- Label Encoder: represents the data on which an encoding operation was performed in the preprocessing stage.
4- Classification Model: It is the model for the hybrid proposal to detect DDoS attacks
5- Clustering Model: It is the model for collecting the attack data into groups according to the type of algorithm used
6- Dataset Path: represents the original data set that was used, which is the data that was referred to in section 3.4. as a first case study.

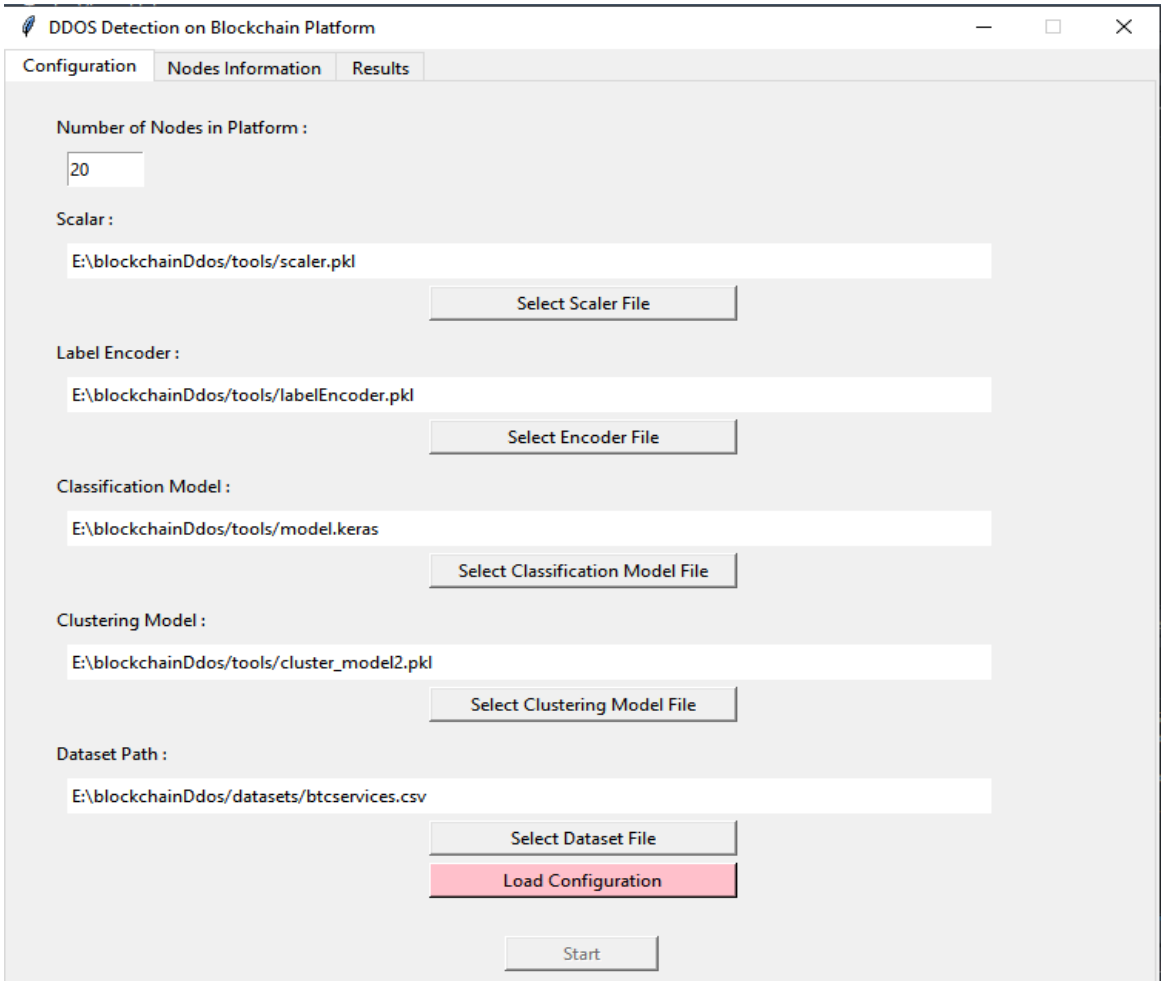


Figure 4.12: GUI Model

Our study created part of a GUI application for DDOS detection on a blockchain platform. This model uses the Tkinter library for creating the GUI and utilizes machine learning models to detect DDOS attacks on blockchain nodes. There are functions defined for various actions, such as selecting files, loading configurations, starting the operation, and handling node selection. A Node class is defined to represent a blockchain node. Each node has attributes for values, clusters, remaining blocked transactions, and an attack flag. It also includes methods for generating random data, preprocessing data, and making predictions using machine learning models. Two GUI were built as follows:

4.5.1 GUI Model 1 Using the AHC Algorithm

In this model, the clustering model, which uses the AHC algorithm, has been applied. The first step is to input 20 nodes to test them on the GUI when the model is running. The nodes were initially given values, and preprocessing was applied to each node. The values were gathered from the original dataset. finding the unique values, adding them to the new data, and then randomly assigning these values to the nodes. All nodes in the initial round are assumed to be normal, therefore they are tested after entering the first model, which represents the hybrid model. The nodes will be granted the green signal if they are normal, but if they attack, they are moved towards the second model, which is essentially the cluster represents. The cluster model of this GUI will be based on the AHC algorithm. The node will then be analyzed after that. It will give a yellow color and one transition if it is in the first cluster, represented by cluster 0, an orange color and five transitions if it is in the second cluster, represented by cluster 1, and a red color and a permanent block if it is in the third cluster, represented by cluster 2.

❖ **The First Round for GUI Model 1**

The first, second, and third nodes have all changed to a green color, indicating that all is normal. The fourth node changed to orange, indicating that the model identified it as an attack. It will remain this way for five transactions, after which it is scheduled for another test. About the fifth node, it appears that it has adopted the yellow color, indicating that it is an attack and will wait for one transition before reentering it for another test. The algorithm assigned the sixth node a red color considering it an attack, and permanently placed a block on it. These nodes are displayed in Figure 4.13.

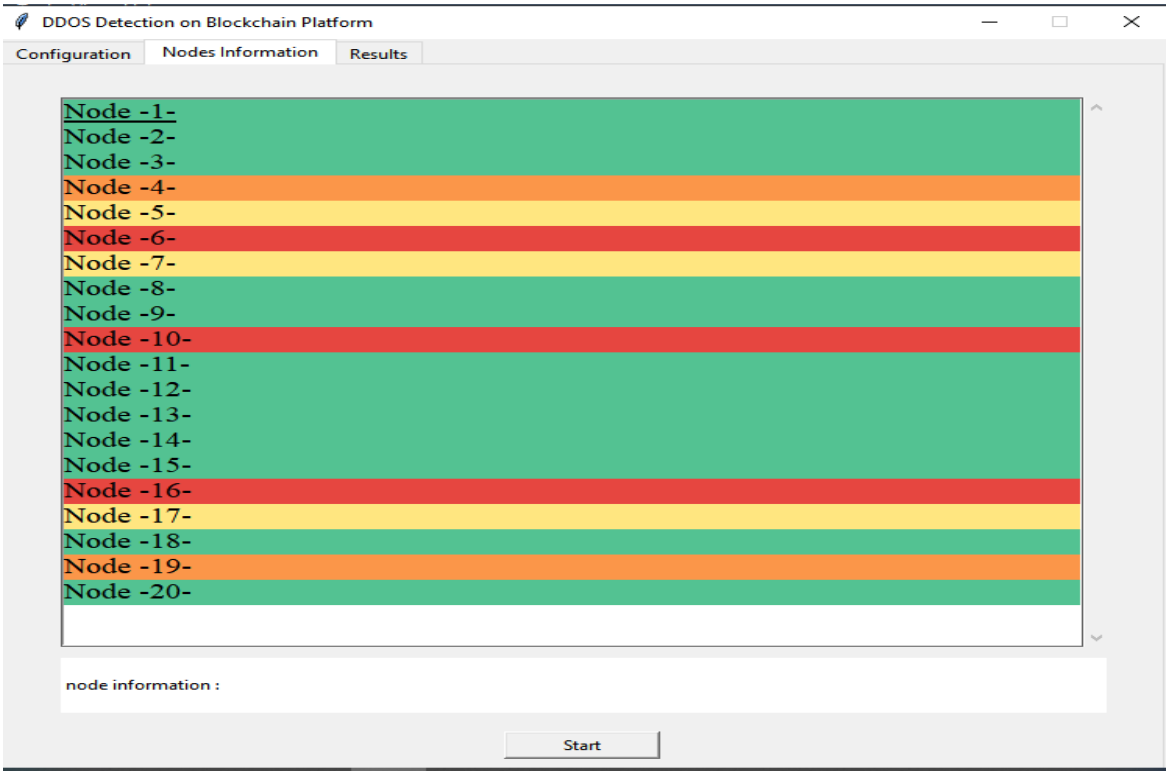


Figure 4.13. List of nodes for the first round in GUI model1

The nodes' state is shown on the Result tab. They are not shown in this window if their status is Normal. It is recorded that these nodes have been given a Block if they are attacked. Furthermore, the status of nodes that take more than one transduction will be shown. The outcomes of the hybrid model for such nodes will be displayed that represent the result of the prediction for the nodes. Additionally, The total time of these procedures will also be computed. The total time of execution for the first test was 4.869399070739746 in seconds as illustrated in Figure 4.14.

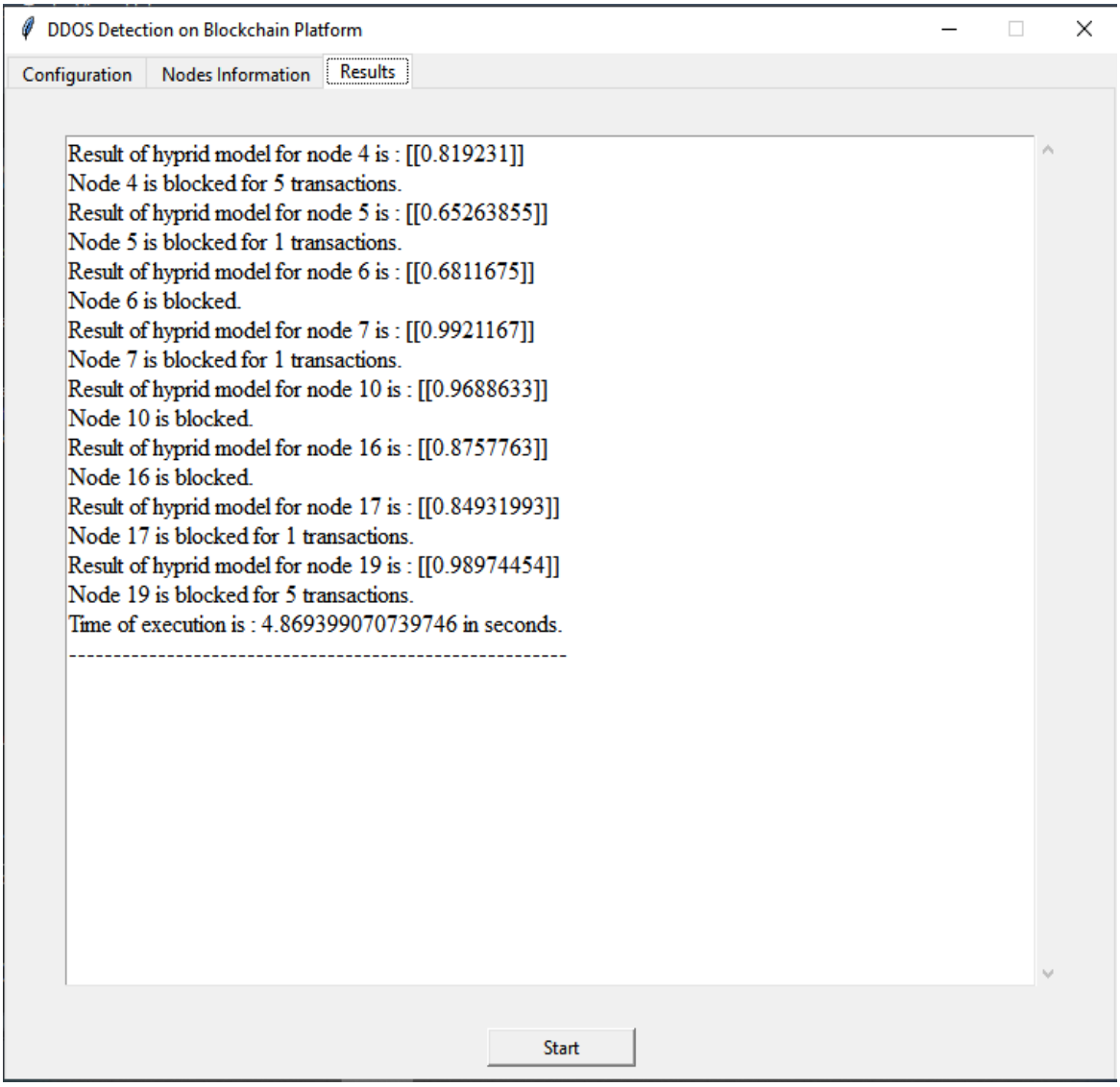


Figure 4.14. The results for the first round in GUI model 1

❖ The Second Round for GUI Model 1

The red nodes, which are nodes 6, 10, and 16, stayed constant in the second test because they had previously given a block for all time. Eventually, it found out that Node 9 had changed its status from Normal to Attack in yellow color, indicating that it would wait for one transaction before reentering the system to retest it. Whereas Nodes 12, 14, and 16 all became attacks and were permanently assigned the color red and blocks, respectively. Figure 4.15 illustrates this procedure.

Figure 4.15. List of nodes for the second round in GUI model 1

According to Figure 4.16, which displays the results of the second test, Node 4 is still blocking four transactions, while Nodes 5 and 7 are no longer blocking any transactions. Additionally, the outcome of the hybrid model for Node 9 is 0.99167717, which has led to the assignment of a block for one transition to Node 9. Since many nodes have received a permanent block, preventing them from re-entering the testing system, the overall execution time is now shorter than in the previous stage. As a result, it took 1.1465935707092285 seconds to complete all actions on all nodes.

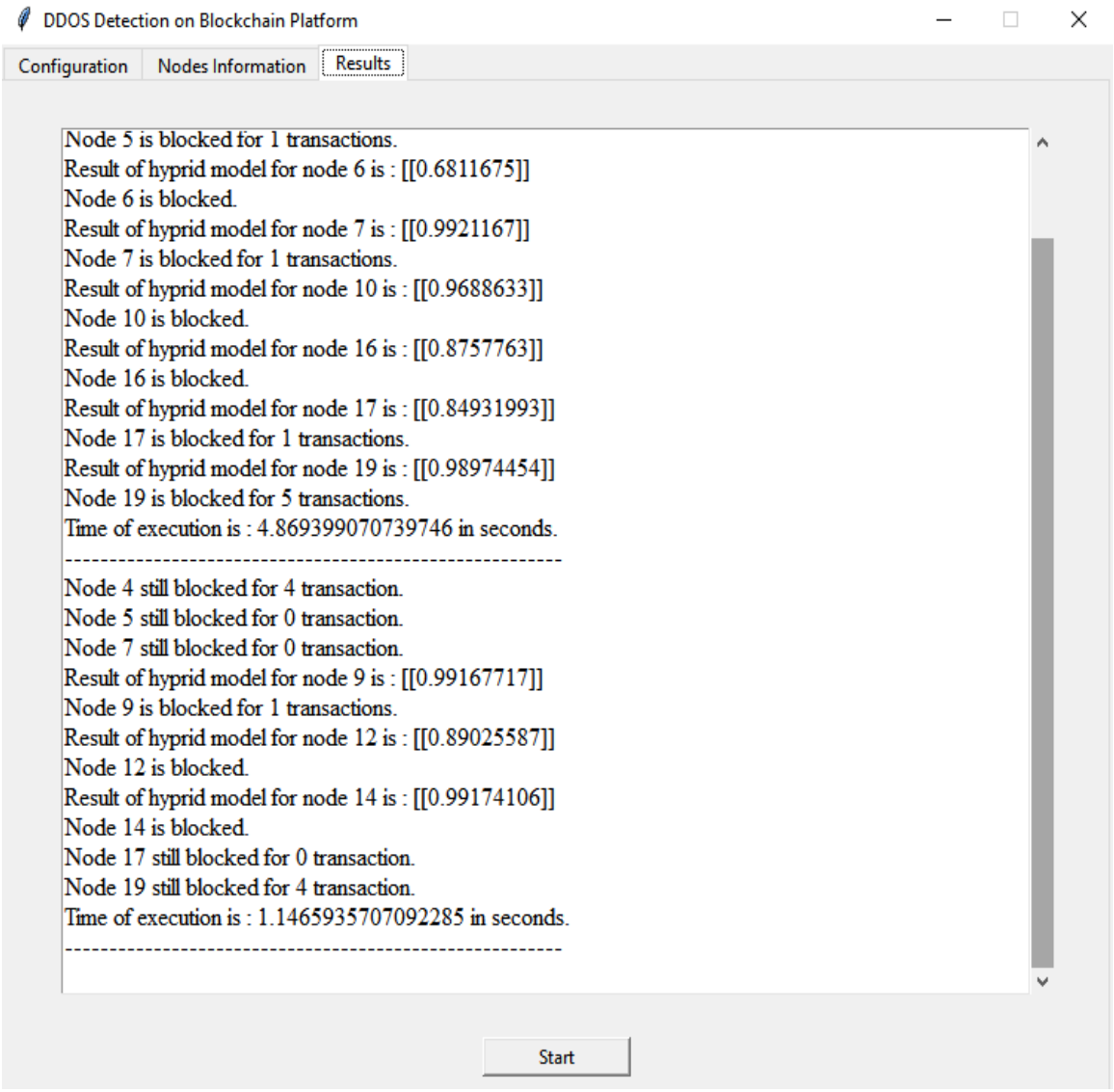


Figure 4.16. The results for the second round in GUI model 1

❖ The Third Round for GUI Model 1

In the third test, it was revealed that nodes 6, 10, 12, 14, and 16 were blocked indefinitely and did not participate in the test. On the other hand, nodes 1, 2, and 3 were marked as normal and assigned a green status, granting them access to the blockchain network.

Node 5 remains in an orange state due to its classification as an attack node, but with limitations on its transmissions. Node 7, previously yellow, has been reclassified as red, indicating it was detected as a threat and subsequently blocked from future access to the system. The status of

the remaining nodes varies according to their respective conditions as depicted in Figure 4.17.

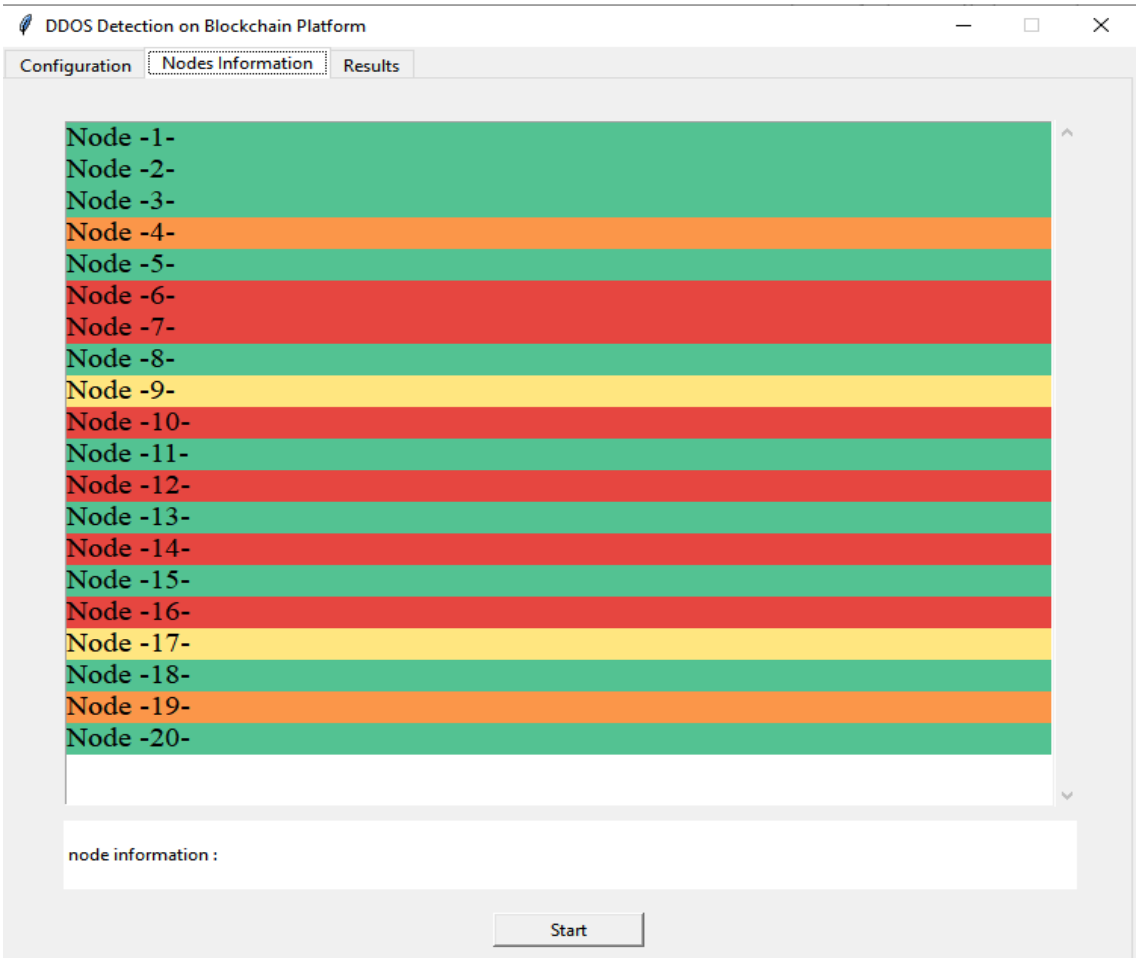


Figure 4.17. List of nodes for the third round in GUI model 1

The list of results for the third test shows that Node 4 remains blocked for three transitions, and the result of the hybrid model for Node 7 is 0.97920156. Therefore, Node 7 is considered an attack and is given a block. Node 9 remains blocked for zero transmissions. Meanwhile, the result of the hybrid model for Node 17 is 0.97154593. Consequently, Node 17 is used to block one transition. Node 19 continues to block for three transitions. Finally, the total execution time was calculated as follows: 1.1178386211395264. Figure 4.18 illustrates these results.

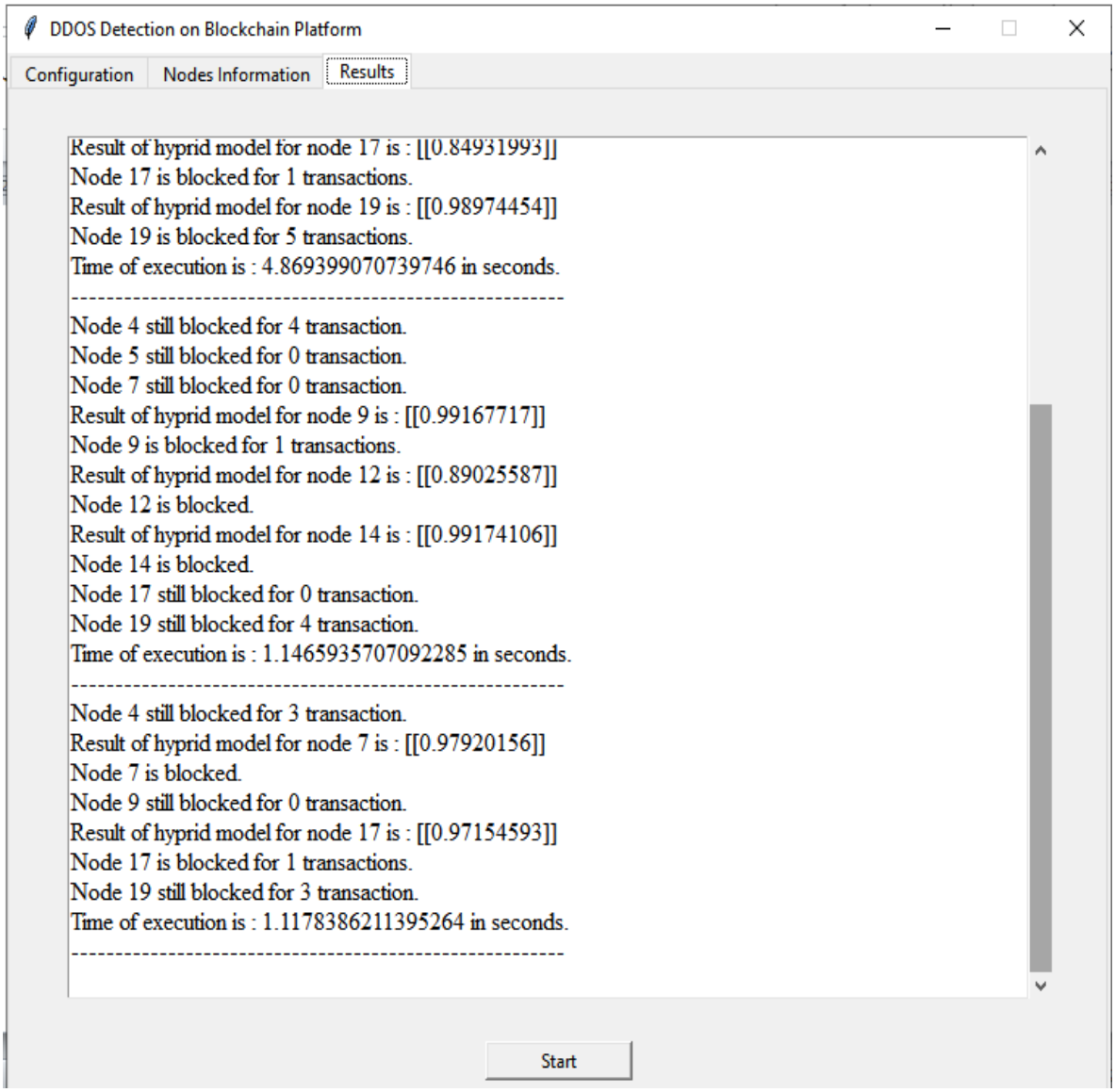


Figure 4.18: The results for the third round in GUI model 1

4.5.2 GUI Model 2 Using the GMM Algorithm

In this model, the clustering model, which uses the GMM algorithm, has been applied. The first step is to input 20 nodes to test them on the platform when the simulation is running. This platform differs from the first platform because it is based on the GMM algorithm. The result of this algorithm yields two clusters. The node can either be attacked and remain blocked indefinitely while being marked in red, or it can be attacked but marked in orange and allowed to re-enter the system for testing after five transitions.

❖ The First Round for GUI Model 2

At the beginning of the initial testing of the second simulation model, we discovered that nodes 6, 8, and 15 were assigned a red color and blocked indefinitely because they were deemed the most susceptible to attacks.

On the other hand, nodes 1, 2, 3, 11, 12, 16, and 20 were marked with orange and considered an attack, but they will remain in a transaction state for five cycles before being allowed to re-enter the system for further testing. As for the remaining nodes, they were assigned a green color, signifying that they are operating normally as depicted in figure 4.19.

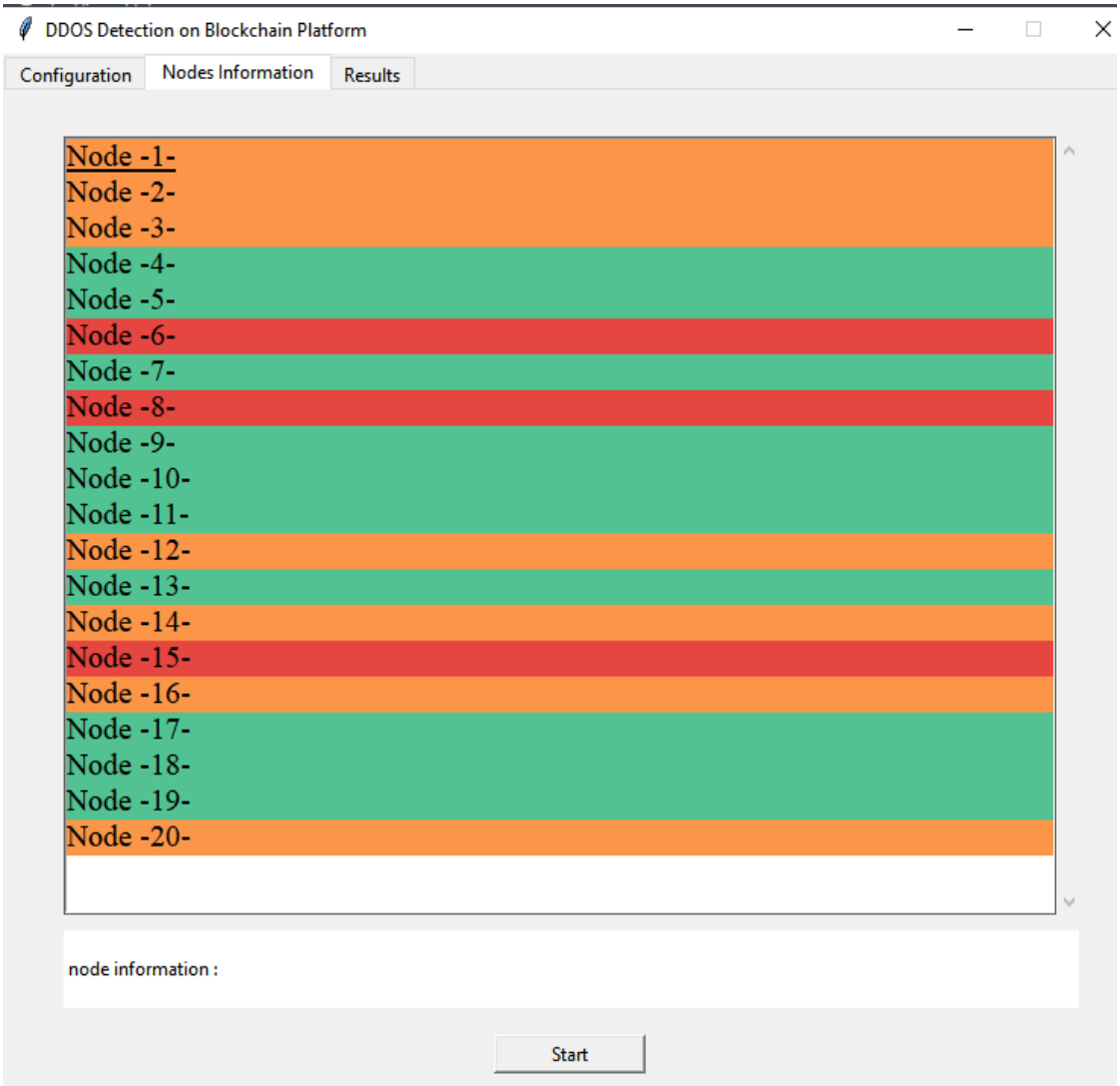


Figure 4.19. List of nodes for the first round in GUI model 2

The results window shows the result of the hybrid model for Node 1, which is 0.99223083. It was considered an attack and given a block for five transactions. The result of the hybrid model for Node 2 was 0.99133915. It was also considered an attack and was given a block for five transactions, while the result for Node 6 was given a permanent block and was considered an attack. The result of the hybrid model for it was 0.7823493 as depicted in figure 4.20.

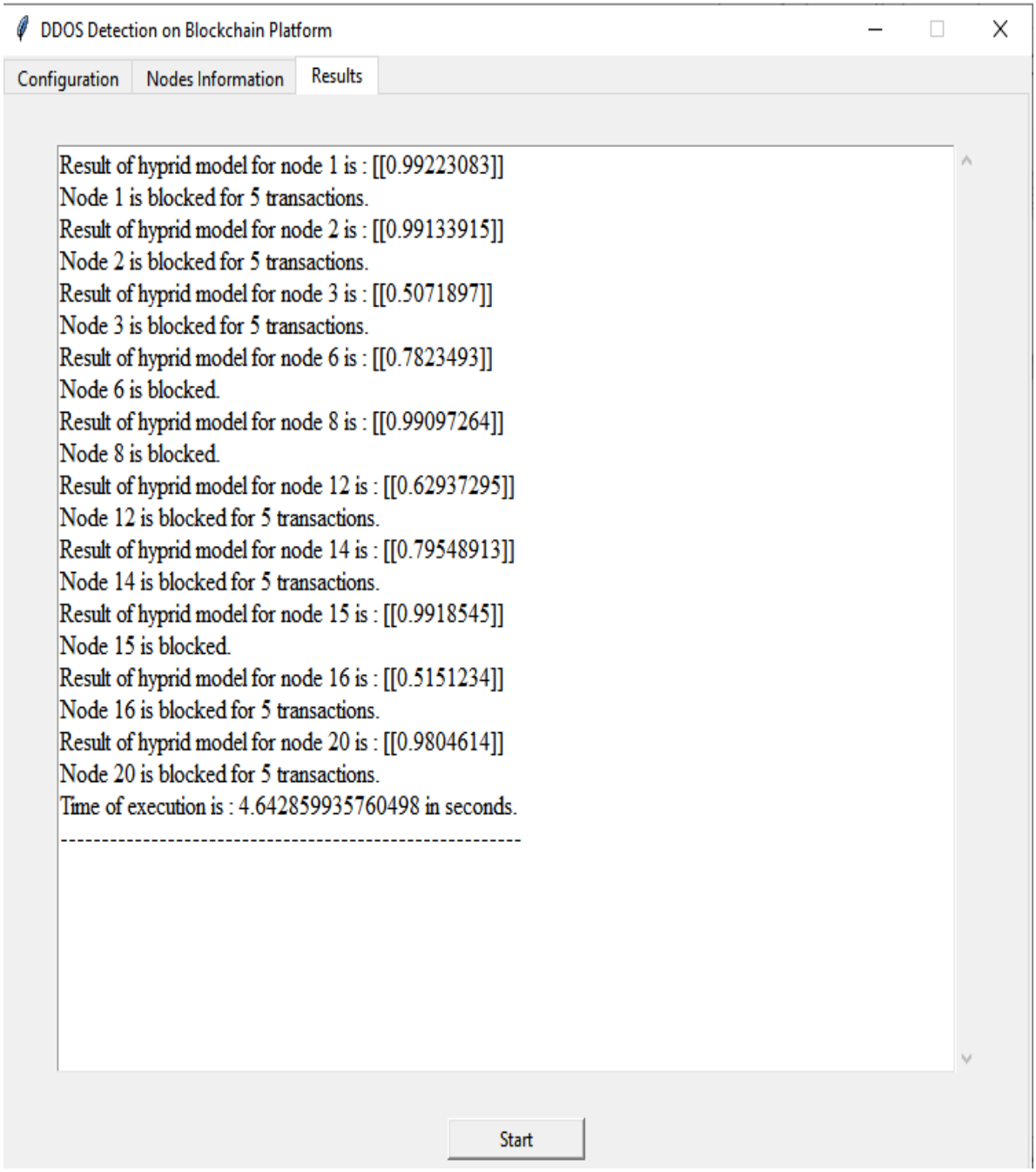


Figure 4.20. The results for the first round in GUI model 2

❖ The Second Round for GUI Model 2

In this attempt, it was found that nodes 5, 9, 10, and 13 had changed from normal to attack. However, they would be given a block for five transactions. After that, they would be allowed to enter the testing mode again. As for nodes 6, 8, and 15, they are not allowed to enter the blockchain network at all. Instead, they are given a block from the previous state of the test as depicted in figure 4.21.

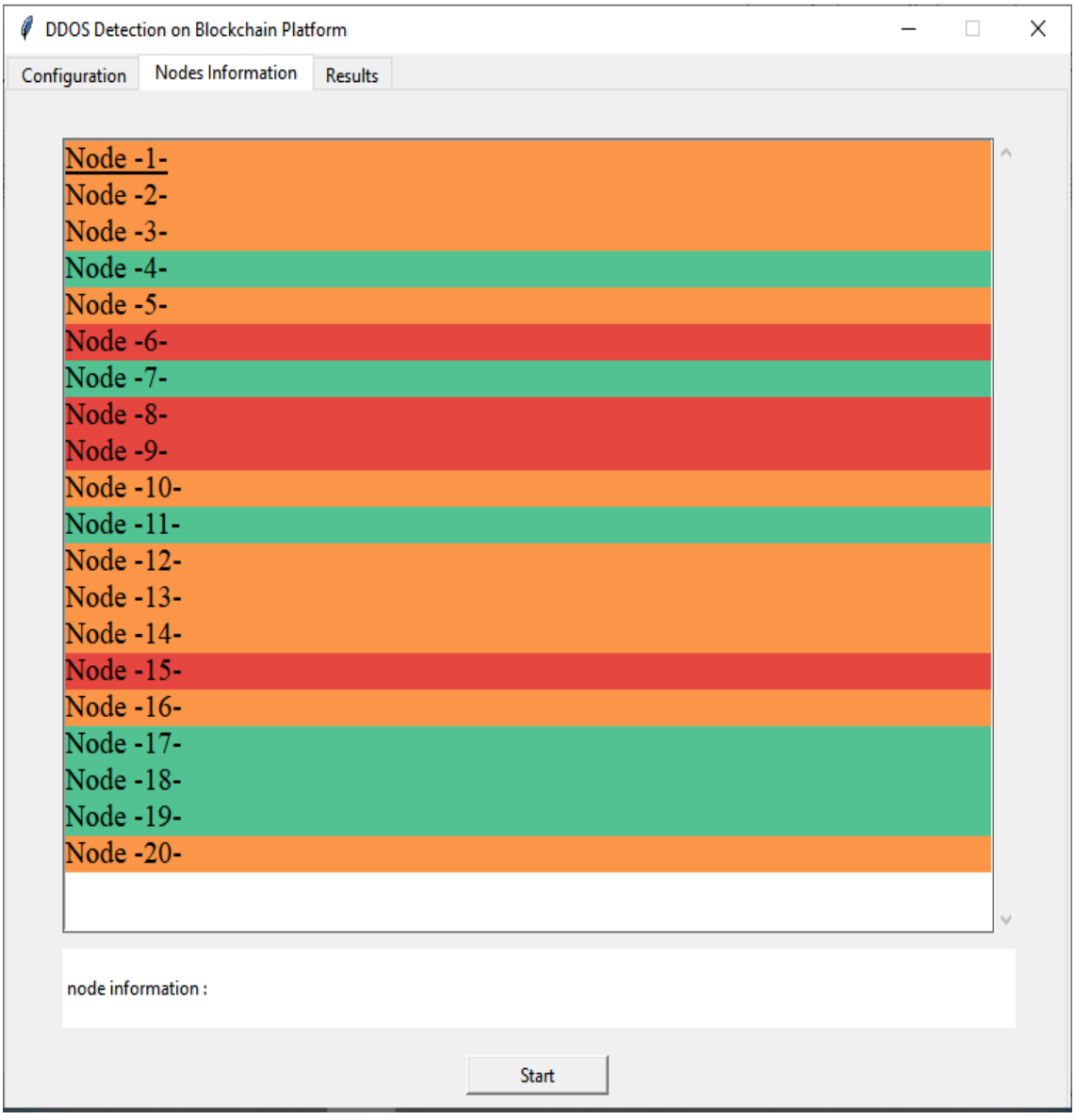


Figure 4.21. List of nodes for the second round in GUI model 2

The results shows that nodes 1, 2, 3, 5, and 13 remained in a block for five transactions, while nodes 14, 16, and 20 remained in a block for

four transactions. Node 9 was given a block indefinitely, as the hybrid model result for Node 9 was 0.96820396. In contrast, the results of the hybrid model for nodes 10 and 13 were 0.5569448 and 0.8662779, respectively. Ultimately, the total execution time for the second test was calculated to be 0.880664587020874 seconds as depicted in figure 4.22.

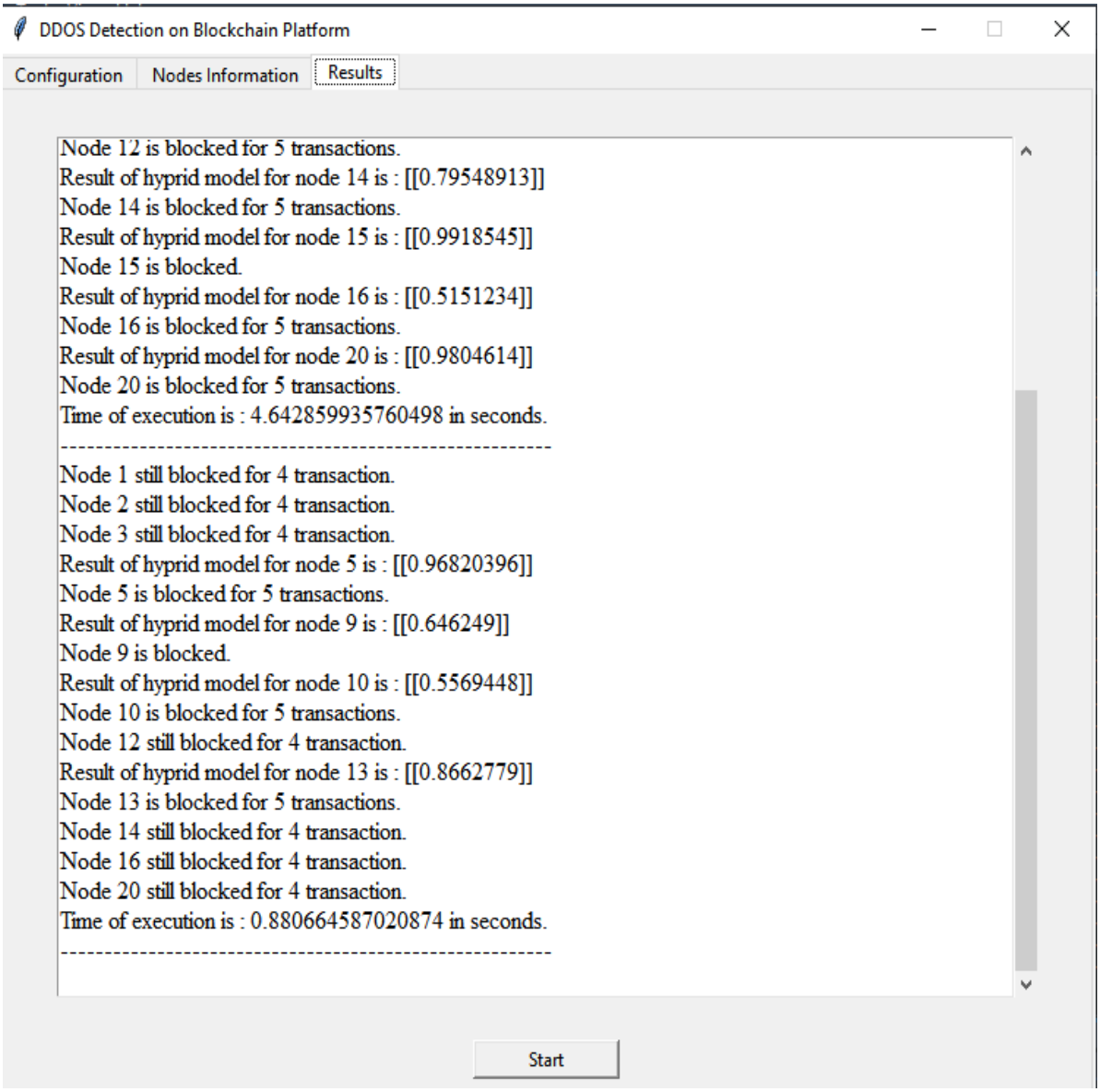


Figure 4.22. The results for the second round in GUI model 2

❖ The Third Round for GUI Model 2

Nodes 6, 8, 9, and 15 are still blocked indefinitely. Meanwhile, nodes 4, 7, 11, 17, and 19 have been marked in green, as the system deems them normal. The remaining nodes are considered attacks and will

remain blocked for a certain number of transfers, until re-entry into the system is permitted for further testing. When referring to any node, we find that all of its information is displayed at the bottom of the screen, represented by the images associated with these nodes. Figure 4.23 provides a visual representation of this process.

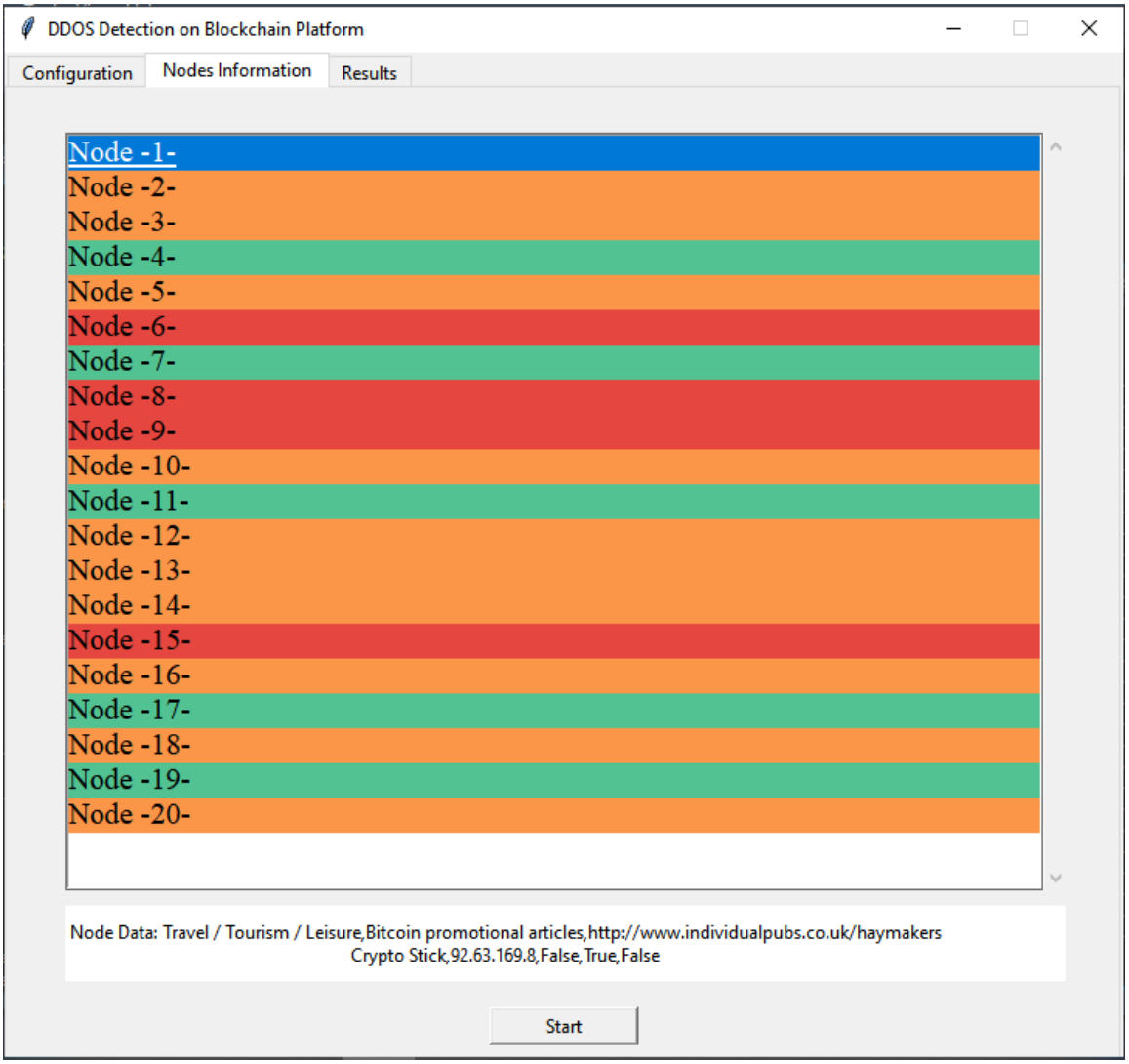


Figure 4.23. List of nodes for the third round in GUI model 2

The results screen shows that nodes 1, 2, 3, 12, 14, 16, and 20 still form a three-transaction block. Nodes 5 and 13 continue to constitute a four-transaction block. Node 18, which was attacked and given five transactions, has been re-entered into the system for testing purposes. This decision was made because the result of the hybrid model for these nodes is 0.99203783. The total time taken for this third test was

0.5468471050262451. Figure 4.24 provides a detailed overview of this stage.

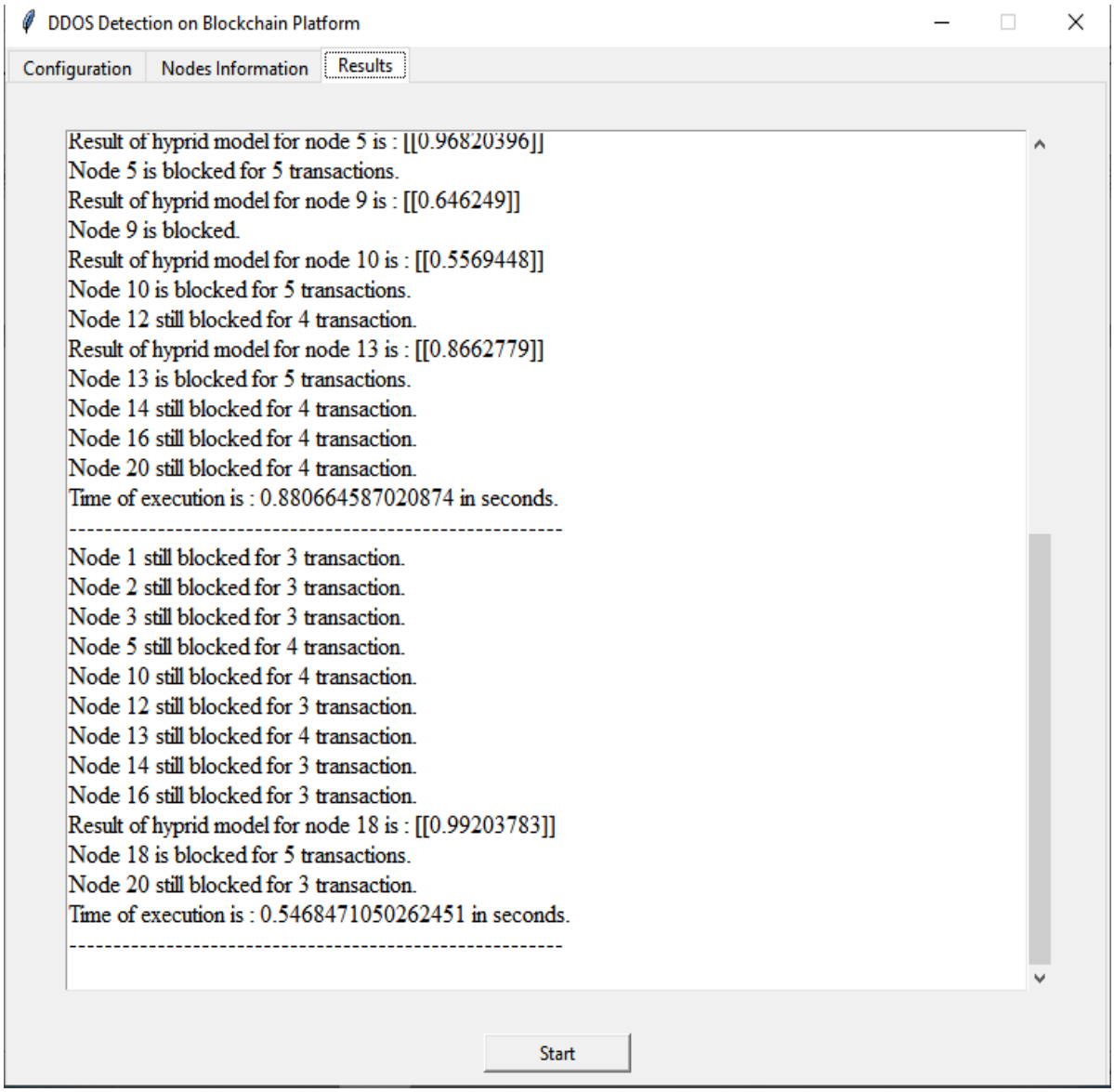


Figure 4.24. The results for the third round in GUI model 2

Table 4.7 displays the total time taken to perform the three tests on the first and second GUI models. It turns out that the second GUI model, which utilizes the GMM algorithm in the assembly stage, requires less execution time than the first simulation model, which relies on the AHC algorithm.

Table 4.7. Total execution time for all GUI models

Round n.	Total execution time for GUI model 1	Total execution time for GUI model 2
First Round	4.869399070739746 second	4.642859935760448 second
Second Round	1.465935707092285 second	0.880664587020874 second
Third Round	1.1178386211395264 second	0.5468471050262451 second

Figure 4.25 showing the total time taken to perform tests on the first and second GUI models, and highlights the difference in execution times between the two models due to their reliance on different algorithms (GMM for the second GUI model and AHC for the first GUI model).

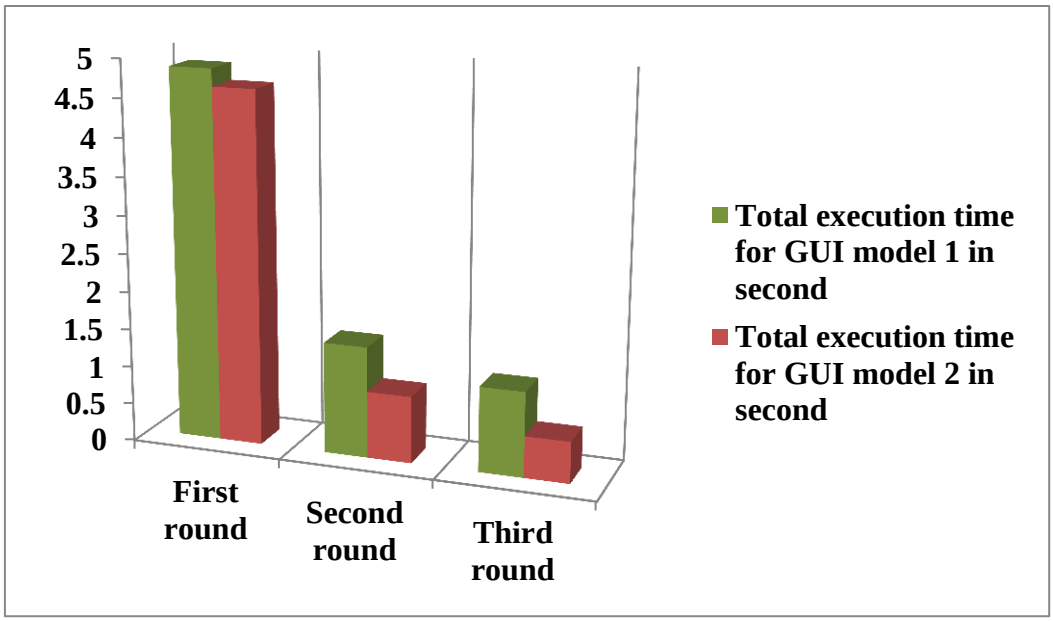


Figure 4.25:Total execution time for GUI model 1 and GUI model 2

Chapter Five

The Conclusions and Future Works

5.1 The Conclusions

Protecting a cryptocurrency platform from DDoS attacks is crucial to ensure the platform's availability and reliability. The employing a combination of some strategies can significantly reduce the risk and impact of such attacks on cryptocurrency platform. The rest of the conclusion can be listed as follows:

1- The proposed hybrid deep learning model for DDoS attack detection is considered an acceptable solution for detection and a highly accurate model as a classification model .

2- The success of proposed hybrid model greatly depends on specific data and the data preprocessing steps. It's recommended to experiment with different architectures and hyper parameters and to ensure that the data is properly preprocessed before feeding it into the model. Also, it's good practice to monitor the training process using validation data to prevent overfitting and achieve the best results.

3- The proposed clustering model can adapt to changing network conditions and evolving attack patterns. They can learn and adjust over time, improving their effectiveness by grouping similar data points together.

4- Data augmentation is a frequently employed technique in deep learning and machine learning geared towards improving the performance of models, particularly in tasks like object detection and natural language processing..

5- The specific results we obtain will vary depending on the quality and size of our dataset, the choice of ML approach, and the level of expertise in feature engineering and model tuning. Continuous monitoring and updating of the model are also crucial, as DDoS attack patterns can change over time.

6- The proposal simulation model identifies and mitigates DDoS attacks, helping to keep the platform of cryptocurrency exchange accessible.

5.2 The Suggested Future Works

1- Remaining current with the latest research is imperative and adapt the clustering-based detection system accordingly to maintain its effectiveness. Additionally, combining unsupervised clustering with other anomaly detection techniques or supervised machine learning approaches can enhance the accuracy of DDoS attack detection.

2- Implement advanced artificial intelligence and machine learning techniques for real-time DDoS detection on blockchain networks. This could involve the development of models that adapt to evolving attack patterns.

3- Investigate methods for coordinating DDoS detection and mitigation across multiple blockchain networks, potentially through inter-blockchain communication protocols.

4- The blockchain landscape evolves, and so do the potential threats and attack vectors. It's important to stay updated with the latest research and technologies to effectively defend against DDoS attacks on blockchain networks.

5- Develop algorithms and protocols for dynamically allocating resources (such as bandwidth and computing power) based on real-time network conditions and potential DDoS threats.

References

References

[1] Z. Che, Y. Wang, J. Zhao, Y. Qiang, Y. Ma, and J. Liu, “A distributed energy trading authentication mechanism based on a consortium blockchain,” *Energies*, vol. 12, no. 15, p. 2878, 2019.

[2] M. J. Mirchev and S. T. Mirtchev, “System for DDoS attack mitigation by discovering the attack vectors through statistical traffic analysis,” *Int. J. Inf. Comput. Secur.*, vol. 13, no. 3–4, pp. 309–321, 2020.

[3] P. Singh, Z. Elmi, Y. Lau, M. Borowska-Stefańska, S. Wiśniewski, and M. A. Dulebenets, “Blockchain and AI technology convergence: Applications in transportation systems,” *Veh. Commun.*, p. 100521, 2022.

[4] S. Jabbar, H. Lloyd, M. Hammoudeh, B. Adebisi, and U. Raza, “Blockchain-enabled supply chain: analysis, challenges, and future directions,” *Multimed. Syst.*, vol. 27, pp. 787–806, 2021.

[5] N. S. Lo, “An overview of the LSTM-based model in forecasting the directional movement of Bitcoin prices,” 2022.

[6] X. Xu and M. Yoneda, “Multitask air-quality prediction based on LSTM-autoencoder model,” *IEEE Trans. Cybern.*, vol. 51, no. 5, pp. 2577–2586, 2019.

[7] A. Abhishta, R. Joosten, S. Dragomiretskiy, and L. J. M. Nieuwenhuis, “Impact of successful ddos attacks on a major crypto-currency exchange,” in *2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, 2019, pp. 379–384.

[8] F. Sabry, W. Labda, A. Erbad, and Q. Malluhi, “Cryptocurrencies and artificial intelligence: Challenges and opportunities,” *IEEE Access*, vol. 8, pp. 175840–175858, 2020.

[9] S. Wani, M. Imthiyas, H. Almohamedh, K. M. Alhamed, S. Almotairi, and Y. Gulzar, “Distributed denial of service (DDoS) mitigation using blockchain—A comprehensive insight,” *Symmetry (Basel).*, vol. 13, no. 2, p. 227, 2021.

[10] U.-J. Baek, S.-H. Ji, J. T. Park, M.-S. Lee, J.-S. Park, and M.-S. Kim, “DDoS attack detection on bitcoin ecosystem using deep-learning,” in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2019, pp. 1–4.

[11] H. S. Yin and R. Vatrapu, “A first estimation of the proportion of cybercriminal entities in the bitcoin ecosystem using supervised machine learning,” in *2017 IEEE international conference on big data (Big Data)*, 2017, pp. 3690–3699.

[12] J. M. Aguiar-Pérez, M. A. Pérez-Juárez, M. Alonso-Felipe, J. Del-Pozo-Velázquez, S. Rozada-Raneros, and M. Barrio-Conde, “Understanding Machine Learning Concepts,” in *Encyclopedia of Data Science and Machine Learning*, IGI Global, 2023, pp. 1007–1022.

[13] M. Vasek, M. Thornton, and T. Moore, “Empirical analysis of denial-of-service attacks in the Bitcoin ecosystem,” in *International conference on financial cryptography and data security*, 2014, pp. 57–71.

[14] A. Feder, N. Gandal, J. T. Hamrick, and T. Moore, “The impact of DDoS and other security shocks on Bitcoin currency exchanges:

Evidence from Mt. Gox," *J. Cybersecurity*, vol. 3, no. 2, pp. 137–144, 2017.

[15] K. Toyoda, T. Ohtsuki, and P. T. Mathiopoulos, "Multi-class bitcoin-enabled service identification based on transaction history summarization," in *2018 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData)*, 2018, pp. 1153–1160.

[16] S. Dragomiretskiy, "The influence of DDoS attacks on cryptocurrency exchanges." University of Twente, 2018.

[17] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[18] R. Greene and M. N. Johnstone, "An investigation into a denial of service attack on an ethereum network," 2018.

[19] D. Alves, "A strategy for mitigating denial of service attacks on nodes with delegate account of Lisk blockchain," in *Proceedings of the 2020 the 2nd International Conference on Blockchain Technology*, 2020, pp. 7–12.

[20] S. Sayadi, S. Ben Rejeb, and Z. Choukair, "Anomaly detection model over blockchain electronic transactions," in *2019 15th international wireless communications & mobile computing conference (IWCMC)*, 2019, pp. 895–900.

[21] B. Jia and Y. Liang, "Anti-D chain: A lightweight DDoS attack detection scheme based on heterogeneous ensemble learning in

blockchain," *China Commun.*, vol. 17, no. 9, pp. 11–24, 2020.

[22] F. Scicchitano, A. Liguori, M. Guarascio, E. Ritacco, and G. Manco, "A Deep Learning Approach for Detecting Security Attacks on Blockchain.," in *ITASEC*, 2020, pp. 212–222.

[23] M. Bastiaan, "Preventing the 51%-attack: a stochastic analysis of two phase proof of work in bitcoin," 2015.

[24] P. S. Goyal, A. Kakkar, G. Vinod, and G. Joseph, "Crypto-ransomware detection using behavioural analysis," in *Reliability, Safety and Hazard Assessment for Risk-Based Technologies*, Springer, 2020, pp. 239–251.

[25] J. Xu and B. Livshits, "The Anatomy of a Cryptocurrency {Pump-and-Dump} Scheme," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 1609–1625.

[26] F. Victor and T. Hagemann, "Cryptocurrency pump and dump schemes: Quantification and detection," in *2019 International Conference on Data Mining Workshops (ICDMW)*, 2019, pp. 244–251.

[27] G. Xu *et al.*, "Am I eclipsed? A smart detector of eclipse attacks for Ethereum," *Comput. Secur.*, vol. 88, p. 101604, 2020.

[28] B. Alangot, D. Reijsbergen, S. Venugopalan, and P. Szalachowski, "Decentralized lightweight detection of eclipse attacks on bitcoin clients," in *2020 IEEE International Conference on Blockchain (Blockchain)*, 2020, pp. 337–342.

[29] A. Kharraz *et al.*, "Outguard: Detecting in-browser covert cryptocurrency mining in the wild," in *The World Wide Web*

Conference, 2019, pp. 840–852.

[30] H. S. Talabani and H. M. T. Abdulhadi, “Bitcoin Ransomware Detection Employing Rule-Based Algorithms,” *Sci. J. Univ. Zakho*, vol. 10, no. 1, pp. 5–10, 2022.

[31] A. Zimba, M. Chishimba, C. Ngongola-Reinke, and T. F. Mbale, “Demystifying cryptocurrency mining attacks: A semi-supervised learning approach based on digital forensics and dynamic network characteristics,” *arXiv Prepr. arXiv2102.10634*, 2021.

[32] R. Ning, C. Wang, C. Xin, J. Li, L. Zhu, and H. Wu, “Capjack: Capture in-browser crypto-jacking by deep capsule network through behavioral analysis,” in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, 2019, pp. 1873–1881.

[33] M. Caprolu, S. Raponi, G. Oligeri, and R. Di Pietro, “Cryptomining makes noise: Detecting cryptojacking via Machine Learning,” *Comput. Commun.*, vol. 171, pp. 126–139, 2021.

[34] K. Toyoda, P. T. Mathiopoulos, and T. Ohtsuki, “A novel methodology for hyip operators’ bitcoin addresses identification,” *IEEE Access*, vol. 7, pp. 74835–74848, 2019.

[35] M. Bartoletti, B. Pes, and S. Serusi, “Data mining for detecting bitcoin ponzi schemes,” in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, 2018, pp. 75–84.

[36] M. Bartoletti, S. Carta, T. Cimoli, and R. Saia, “Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact,” *Futur. Gener. Comput. Syst.*, vol. 102, pp. 259–277, 2020.

[37] W. Chen, Z. Zheng, E. C.-H. Ngai, P. Zheng, and Y. Zhou,

"Exploiting blockchain data to detect smart ponzi schemes on ethereum," *IEEE Access*, vol. 7, pp. 37575–37586, 2019.

[38] W. Wang *et al.*, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *Ieee Access*, vol. 7, pp. 22328–22370, 2019.

[39] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Bus. Rev.*, 2008.

[40] A. Singh, K. Click, R. M. Parizi, Q. Zhang, A. Dehghantanha, and K.-K. R. Choo, "Sidechain technologies in blockchain networks: An examination and state-of-the-art review," *J. Netw. Comput. Appl.*, vol. 149, p. 102471, 2020.

[41] J. Golosova, A. Romanovs, and N. Kunicina, "Review of the blockchain technology in the energy sector," in *2019 IEEE 7th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, 2019, pp. 1–7.

[42] D. Wang, J. Zhou, A. Wang, and M. Finestone, "Loopring: A decentralized token exchange protocol," *URL https//github. com/Loopring/whitepaper/blob/master/en_whitepaper. pdf*, 2018.

[43] Y. He, H. Li, X. Cheng, Y. Liu, C. Yang, and L. Sun, "A blockchain based truthful incentive mechanism for distributed P2P applications," *IEEE access*, vol. 6, pp. 27324–27335, 2018.

[44] Y. Yuan and F.-Y. Wang, "Blockchain: the state of the art and future trends," *Acta Autom. Sin.*, vol. 42, no. 4, pp. 481–494, 2016.

[45] M. S. Ferdous, M. J. M. Chowdhury, and M. A. Hoque, "A survey of consensus algorithms in public blockchain systems for crypto-

currencies," *J. Netw. Comput. Appl.*, vol. 182, p. 103035, 2021.

[46] B. Cao, X. Wang, W. Zhang, H. Song, and Z. Lv, "A many-objective optimization model of industrial internet of things based on private blockchain," *IEEE Netw.*, vol. 34, no. 5, pp. 78–83, 2020.

[47] B. Zhong, H. Wu, L. Ding, H. Luo, Y. Luo, and X. Pan, "Hyperledger fabric-based consortium blockchain for construction quality information management," *Front. Eng. Manag.*, vol. 7, no. 4, pp. 512–527, 2020.

[48] É. R. Keresztes, I. Kovács, A. Horváth, and K. Zimányi, "Exploratory analysis of blockchain platforms in supply chain management," *Economies*, vol. 10, no. 9, p. 206, 2022.

[49] C. Shen and F. Pena-Mora, "Blockchain for cities—a systematic literature review," *Ieee Access*, vol. 6, pp. 76787–76819, 2018.

[50] E. Erturk, D. Lopez, and W. Y. Yu, "Benefits and risks of using blockchain in smart energy: A literature review," *Contemp. Manag. Res.*, vol. 15, no. 3, pp. 205–225, 2019.

[51] S. Joshi *et al.*, "Adoption of blockchain technology for privacy and security in the context of industry 4.0," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022.

[52] S. S. Hazari and Q. H. Mahmoud, "A parallel proof of work to improve transaction speed and scalability in blockchain systems," in *2019 IEEE 9th annual computing and communication workshop and conference (CCWC)*, 2019, pp. 916–921.

[53] H. Chen and Y. Wang, "Sschain: A full sharding protocol for public blockchain without data migration overhead," *Pervasive Mob.*

Comput., vol. 59, p. 101055, 2019.

[54] S. Lee and H. Kim, "On the robustness of lightning network in bitcoin," *Pervasive Mob. Comput.*, vol. 61, p. 101108, 2020.

[55] M. Bez, G. Fornari, and T. Vardanega, "The scalability challenge of ethereum: An initial quantitative analysis," in *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, 2019, pp. 167–176.

[56] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "A survey on the scalability of blockchain systems," *IEEE Netw.*, vol. 33, no. 5, pp. 166–173, 2019.

[57] B. Cao *et al.*, "Performance analysis and comparison of PoW, PoS and DAG based blockchains," *Digit. Commun. Networks*, vol. 6, no. 4, pp. 480–485, 2020.

[58] P. T. Duy, D. T. T. Hien, D. H. Hien, and V.-H. Pham, "A survey on opportunities and challenges of Blockchain technology adoption for revolutionary innovation," in *Proceedings of the 9th International Symposium on Information and Communication Technology*, 2018, pp. 200–207.

[59] T. McGhin, K.-K. R. Choo, C. Z. Liu, and D. He, "Blockchain in healthcare applications: Research challenges and opportunities," *J. Netw. Comput. Appl.*, vol. 135, pp. 62–75, 2019.

[60] B. Bhushan, P. Sinha, K. M. Sagayam, and J. Andrew, "Untangling blockchain technology: A survey on state of the art, security threats, privacy services, applications and future research directions," *Comput. Electr. Eng.*, vol. 90, p. 106897, 2021.

[61] O. Alfandi, S. Khanji, L. Ahmad, and A. Khattak, “A survey on boosting IoT security and privacy through blockchain: Exploration, requirements, and open issues,” *Cluster Comput.*, vol. 24, pp. 37–55, 2021.

[62] T. Wang, H. Hua, Z. Wei, and J. Cao, “Challenges of blockchain in new generation energy systems and future outlooks,” *Int. J. Electr. Power Energy Syst.*, vol. 135, p. 107499, 2022.

[63] M. Avital, “Peer review: Toward a blockchain-enabled market-based ecosystem,” *Commun. Assoc. Inf. Syst.*, vol. 42, pp. 646–653, 2018.

[64] E. Toufaily, T. Zalan, and S. Ben Dhaou, “A framework of blockchain technology adoption: An investigation of challenges and expected value,” *Inf. Manag.*, vol. 58, no. 3, p. 103444, 2021.

[65] G. M. Hastig and M. S. Sodhi, “Blockchain for supply chain traceability: Business requirements and critical success factors,” *Prod. Oper. Manag.*, vol. 29, no. 4, pp. 935–954, 2020.

[66] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, “Using blockchain for medical data access and permission management’,” in *2nd International Conference on Open and Big Data (OBD).(Vienna, 2016)*, 2016, pp. 1–2.

[67] A. Mukherjee, S. Majumdar, A. K. Kolya, and S. Nandi, “A Privacy-Preserving Blockchain-based E-voting System,” *arXiv Prepr. arXiv2307.08412*, 2023.

[68] M. Krichen, M. Ammi, A. Mihoub, and M. Almutiq, “Blockchain for modern applications: A survey,” *Sensors*, vol. 22, no. 14, p. 5274, 2022.

[69] D. K. Sharma, S. Pant, M. Sharma, and S. Brahmachari, "Cryptocurrency mechanisms for blockchains: models, characteristics, challenges, and applications," *Handb. Res. blockchain Technol.*, pp. 323–348, 2020.

[70] C. Campajola, R. Cristodaro, F. M. De Collibus, T. Yan, N. Vallarano, and C. J. Tessone, "The evolution of centralisation on cryptocurrency platforms," *arXiv Prepr. arXiv2206.05081*, 2022.

[71] N. Anita and M. Vijayalakshmi, "Blockchain security attack: a brief survey," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2019, pp. 1–6.

[72] S. Sayeed and H. Marco-Gisbert, "Assessing blockchain consensus and security mechanisms against the 51% attack," *Appl. Sci.*, vol. 9, no. 9, p. 1788, 2019.

[73] N. A. Akbar, A. Muneer, N. ElHakim, and S. M. Fati, "Distributed Hybrid Double-Spending Attack Prevention Mechanism for Proof-of-Work and Proof-of-Stake Blockchain Consensuses," *Futur. Internet*, vol. 13, no. 11, p. 285, 2021.

[74] C. Natoli and V. Gramoli, "The balance attack against proof-of-work blockchains: The R3 testbed as an example," *arXiv Prepr. arXiv1612.09426*, 2016.

[75] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Futur. Gener. Comput. Syst.*, vol. 107, pp. 841–853, 2020.

[76] S. Ramos, F. Pianese, T. Leach, and E. Oliveras, "A great disturbance in the crypto: Understanding cryptocurrency returns

under attacks," *Blockchain Res. Appl.*, vol. 2, no. 3, p. 100021, 2021.

[77] M. Apostolaki, A. Zohar, and L. Vanbever, "Hijacking bitcoin: Routing attacks on cryptocurrencies," in *2017 IEEE symposium on security and privacy (SP)*, 2017, pp. 375–392.

[78] A. Singh, T. W. Ngan, P. Druschel, and D. S. Wallach, "Eclipse attacks on overlay networks: Threats and defenses," 2006. doi: 10.1109/INFOCOM.2006.231.

[79] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on {Bitcoin's}{peer-to-peer} network," in *24th USENIX Security Symposium (USENIX Security 15)*, 2015, pp. 129–144.

[80] M. Guri, "Beatcoin: Leaking private keys from air-gapped cryptocurrency wallets," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 1308–1316.

[81] S. Shalini and H. Santhi, "A survey on various attacks in bitcoin and cryptocurrency," in *2019 International Conference on Communication and Signal Processing (ICCSP)*, 2019, pp. 220–224.

[82] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*, 2002, pp. 251–260.

[83] K. Toyoda, T. Ohtsuki, and P. T. Mathiopoulos, "Identification of high yielding investment programs in bitcoin via transactions pattern analysis," in *GLOBECOM 2017-2017 IEEE Global*

Communications Conference, 2017, pp. 1–6.

[84] E. Badawi and G.-V. Jourdan, “Cryptocurrencies emerging threats and defensive mechanisms: A systematic literature review,” *IEEE Access*, vol. 8, pp. 200021–200037, 2020.

[85] A. M. Maigida, S. M. Abdulhamid, M. Olalere, J. K. Alhassan, H. Chiroma, and E. G. Dada, “Systematic literature review and metadata analysis of ransomware attacks and detection mechanisms,” *J. Reliab. Intell. Environ.*, vol. 5, no. 2, pp. 67–89, 2019.

[86] A. Zimba, Z. Wang, and H. Chen, “Multi-stage crypto ransomware attacks: A new emerging cyber threat to critical infrastructure and industrial control systems,” *Ict Express*, vol. 4, no. 1, pp. 14–18, 2018.

[87] S. Maniath, P. Poornachandran, and V. G. Sujadevi, “Survey on prevention, mitigation and containment of ransomware attacks,” in *International Symposium on Security in Computing and Communication*, 2018, pp. 39–52.

[88] D. B. Kramer, “The Way It Is and the Way It Should Be: Liability Under § 10 (b) of the Exchange Act and Rule 10b-5 Thereunder for Making False and Misleading Statements as Part of a Scheme to" Pump and Dump" a Stock,” *Univ. Miami Bus. Law Rev.*, vol. 13, no. 2, p. 243, 2005.

[89] M. La Morgia, A. Mei, F. Sassi, and J. Stefa, “The doge of wall street: Analysis and detection of pump and dump cryptocurrency manipulations,” *ACM Trans. Internet Technol.*, 2021.

[90] T. S. Ustun and S. M. S. Hussain, “A review of cybersecurity issues

in smartgrid communication networks," in *2019 International Conference on Power Electronics, Control and Automation (ICPECA)*, 2019, pp. 1–6.

[91] G. M. Caporale, W.-Y. Kang, F. Spagnolo, and N. Spagnolo, "Cyber-attacks, cryptocurrencies, and cyber security," 2021.

[92] A. M. Antonopoulos, *Mastering Bitcoin: unlocking digital cryptocurrencies*. " O'Reilly Media, Inc.," 2014.

[93] T. Moore and N. Christin, "Beware the middleman: Empirical analysis of Bitcoin-exchange risk," in *Financial Cryptography and Data Security: 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers 17*, 2013, pp. 25–33.

[94] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts (sok)," in *Principles of Security and Trust: 6th International Conference, POST 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings 6*, 2017, pp. 164–186.

[95] V. G. Comizio, "Virtual currencies: Growing regulatory framework and challenges in the emerging Fintech ecosystem," *NC Bank. Inst.*, vol. 21, p. 131, 2017.

[96] E. Ben Sasson *et al.*, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE symposium on security and privacy*, 2014, pp. 459–474.

[97] C. Harshini, D. D. Parameswari, E. Harshitha, F. Tazeen, and G. Maneesha, "A MACHINE LEARNING BASED

CLASSIFICATION AND PREDICTION TECHNIQUE FOR DDOS ATTACKS," *Int. J. Manag. Res. Rev.*, vol. 13, no. 3, pp. 1–7, 2023.

[98] A. R. Yusof, N. I. Udzir, and A. Selamat, "Systematic literature review and taxonomy for DDoS attack detection and prediction," *Int. J. Digit. Enterp. Technol.*, vol. 1, no. 3, pp. 292–315, 2019.

[99] R. Chaganti, B. Bhushan, and V. Ravi, "The role of Blockchain in DDoS attacks mitigation: techniques, open challenges and future directions," *arXiv Prepr. arXiv2202.03617*, 2022.

[100] M. Saad, J. Spaulding, L. Njilla, C. A. Kamhoua, D. Nyang, and A. Mohaisen, "Overview of attack surfaces in blockchain," *Blockchain Distrib. Syst. Secur.*, pp. 51–66, 2019.

[101] R. Chaganti *et al.*, "A comprehensive review of denial of service attacks in blockchain ecosystem and open challenges," *IEEE Access*, 2022.

[102] B. Mahesh, "Machine learning algorithms-a review," *Int. J. Sci. Res. (IJSR).[Internet]*, vol. 9, no. 1, pp. 381–386, 2020.

[103] A. K. Tyagi and P. Chahal, "Artificial intelligence and machine learning algorithms," in *Research Anthology on Machine Learning Techniques, Methods, and Applications*, IGI Global, 2022, pp. 421–446.

[104] T. Choithani, A. Chowdhury, S. Patel, P. Patel, D. Patel, and M. Shah, "A comprehensive study of artificial intelligence and cybersecurity on Bitcoin, crypto currency and banking system," *Ann. Data Sci.*, pp. 1–33, 2022.

[105] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 2, no. 1, pp. 41–50, 2018.

[106] I. Goodfellow, “Deep Learning-Ian Goodfellow, Yoshua Bengio, Aaron Courville,” *Adapt. Comput. Mach. Learn.*, 2016.

[107] M. Mittal, K. Kumar, and S. Behal, “Deep learning approaches for detecting DDoS attacks: A systematic review,” *Soft Comput.*, vol. 27, no. 18, pp. 13039–13075, 2023.

[108] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[109] T. Jiang, J. L. Gradus, and A. J. Rosellini, “Supervised machine learning: a brief primer,” *Behav. Ther.*, vol. 51, no. 5, pp. 675–687, 2020.

[110] R. Saravanan and P. Sujatha, “A state of art techniques on machine learning algorithms: a perspective of supervised learning approaches in data classification,” in *2018 Second international conference on intelligent computing and control systems (ICICCS)*, 2018, pp. 945–949.

[111] R. S. Khairy, A. S. Hussein, and H. T. H. Salim ALRikabi, “The Detection of Counterfeit Banknotes Using Ensemble Learning Techniques of AdaBoost and Voting.,” *Int. J. Intell. Eng. Syst.*, vol. 14, no. 1, 2021.

[112] N. Rajalingam and K. Ranjini, “Hierarchical clustering algorithm-a comparative study,” *Int. J. Comput. Appl.*, vol. 19, no. 3, pp. 42–46, 2011.

[113] R. J. Gil-Garcia, J. M. Badia-Contelles, and A. Pons-Porrata, “A general framework for agglomerative hierarchical clustering algorithms,” in *18th International Conference on Pattern Recognition (ICPR’06)*, 2006, vol. 2, pp. 569–572.

[114] T. Li, A. Rezaeipanah, and E. M. T. El Din, “An ensemble agglomerative hierarchical clustering algorithm based on clusters clustering technique and the novel similarity measurement,” *J. King Saud Univ. Inf. Sci.*, vol. 34, no. 6, pp. 3828–3842, 2022.

[115] R. C. de Amorim, “Feature relevance in ward’s hierarchical clustering using the L p norm,” *J. Classif.*, vol. 32, pp. 46–62, 2015.

[116] W. Lu, X. Wu, D. Ding, and G. Yuan, “An Efficient 1 Iteration Learning Algorithm for Gaussian Mixture Model And Gaussian Mixture Embedding For Neural Network,” *arXiv Prepr. arXiv2308.09444*, 2023.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية تكنولوجيا المعلومات
قسم شبكات المعلومات

طريقة مطورة للكشف عن هجمات DDoS في خدمات شبكة العملة المشفرة

أطروحة مقدمة إلى مجلس كلية تكنولوجيا المعلومات للدراسات العليا جامعة بابل وهو جزء من متطلبات نيل درجة دكتوراه الفلسفة في تكنولوجيا المعلومات / شبكات المعلومات

بواسطة
آمنة عبد العباس عبد الأمير مهدي

بأشراف
الأستاذ الدكتور وسام سمير بهيه

١٤٤٥ هـ ٢٠٢٣ م

خلاصة

إن الاهتمام المتزايد بتكنولوجياblockchain ، التي تعمل كدفتر حسابات غير قابل للتغيير يسهل المعاملات الموزعة، أمر لافت للنظر. ومع ذلك، فإن أمان البلوكشين عرضة لمجموعة متنوعة من المخاطر، وخاصة هجمات رفض الخدمة الموزعة(DDoS) ، والتي تركز تدريجياً على خدمات شبكة العملة المشفرة. ولمعالجة هذه المشكلة، ظهرت أساليب التعلم العميق المتقدمة كحل فعال للتحديات المعقدة في مجال علم المعلومات.

تقترح الأطروحة دمج هذه الأساليب في أطر عمل هجينة لمواجهة تحديات الأمن السيبراني المعقدة. في البداية، تم تقديم نهج التعلم العميق الهجين المستخدم للتصنيف الثنائي، والذي يجمع بين خوارزميات الشبكة العصبية المتكررة (RNN) والذاكرة طويلة المدى(LSTM) ، للكشف عن هجمات DDoS في خدمات شبكة العملة المشفرة.

وفي وقت لاحق، تم تقديم اقتراح لبناء واجهة مستخدم رسومية (GUI) لمراقبة حركة مرور الشبكة وتحديد الأنماط أو السلوكيات غير الطبيعية. تتضمن هذه الخطة تنفيذ تدابير لمنع التطفلات والرد تلقائيًا على الأنشطة المشبوهة. أنشأ أنموذجين، أحدهما يستخدم طريقة التجميع الهرمي التجميعي (AHC) والآخر يستخدم نموذج الخليط الغاوسي (GMM) لتحديد هجمات DDOSعلى خدمات شبكة العملة المشفرة.

تم تقييم الأنموذج المقترح لاحقًا باستخدام مجموعة البيانات الحقيقية التي تمثل البيانات المرتبطة بخدمات Bitcoin التي تعرضت لهجوم .DDoS ومن الواضح أن النموذج المقترح تفوق على تطبيقات التعلم العميق القياسية، وحقق مستوى مذهلاً من الدقة. عند تقييمه باستخدام مجموعة بياناتMt.Gox ، أظهر النموذج دقة تبلغ ٩٥.٨٤%. علاوة على ذلك، خضعت للتحقق من الصحة باستخدام مجموعات البيانات المعترف بها على نطاق واسع، وتحديدًا CIC–IDS2017وCSE–CIC–IDS2018، محققة دقة تبلغ ٩٥.٤٠% و٩٩.٧٧% على التوالي. لذلك، تقدم هذه الأطروحة استراتيجية واعدة للتخفيف من هجمات DDoS داخل النظام البيئي للعملات المشفرة.