Quest Journals
Journal of Software Engineering and Simulation
Volume 9 ~ Issue 3 (2023) pp: 07-15
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

A Model for the Detection and Prevention of Anomalies in Block chain

[1]Ezekiel,P. S. and [2]Bennett, E. O.
Department of Computer Science, Rivers State University, Port Harcourt, Nigeria

Abstract
Anomaly detection is the identification of rare items, activities, or observations that do not conform towell-defined notations of normal transaction patterns in the blockchain network. Its applications in the financial sector have aided in identifying suspicious activities of hackers, and usual patterns in transactions that are being carried out for fraudulent purposes. Some of these illegal purposes are fraudulent transactions and money laundering in the blockchain networks. Due to these anomalies in the blockchain network, this paperpresents a model for the detection and prevention of anomalies in blockchain transactions. This paper uses a Graph Representation Learning (GRL) model for the detection of anomalies in blockchain. The GRL learns through the normal patterns of the blockchain network and raises an alarm when the newly generated block height must match the previously generated block height, and the values of each individual block must match the previous value of the block The GRL model achieved a detection accuracy of 98.9%. After the detection of anomalous transactions, a Long Short-Term Memory (LSTM) algorithm was used for the prevention of anomalous transactions on the blockchain. The LSTM model stops any transaction from completing after it has been identified as an anomaly. The LSTM model achieved an accuracy result of 99% in preventing anomalous transaction for completing.

Keywords*: Anomalies, Blockchain Technology, Graph Representation Learning, Long Short-Term Memory*

Received 18 Feb., 2023; Revised 28 Feb., 2023; Accepted 02 Mar., 2023 © The author(s) 2023.
Published with open access at www.questjournals.org

I. INTRODUCTION

Since its debut a decade ago, blockchain technology has attracted substantial attention from a wide range of industries, including academia, due to its potential for seamless integration with a number of common applications using Information and Commination Technology (ICT) capabilities. These applications are enhanced by the Peer-to-Peer (P2P) architecture of blockchain, which provides strong security and trust-oriented guarantees, such as immutability, verifiability, and decentralisation. While blockchain technology unquestionably outperforms the aforementioned ICT applications, current research has shown that despite these solid guarantees, blockchain networks may still be susceptible to a variety of security, privacy, and dependability-related concerns. Finding time-related anomalies is essential for resolving these issues [1].

Anomaly detection methods are crucial for the protection of blockchain networks nowadays. These technologies watch for dangers in the case of an unanticipated incursion by independently detecting and forecasting network irregularities. Frequently, virtual currencies such as Bitcoin, Ethereum, Luna, Binance Coin, and Lite coin are linked to the blockchain. It is a database storing records of transactions that is distributed, reviewed, and maintained by a worldwide computer network. Instead of being watched by a single institution, such as a bank, records are monitored by the whole community, and once recorded, they cannot be edited or erased [2]. Unlike conventional centralised databases, Blockchain's decentralised nature and peer-verified guarantees make data falsification impossible. Also, unlike the centralised database on a single server utilisedby the majority of software programs, blockchain data is distributed among all system users. Since every member in the blockchain can see the transactions of every other participant, no one entity can control the blockchain [3].

Blockchain technology depends on the concept of a decentralised database, in which several identical copies are kept on separate computers. The decentralised structure of Blockchain makes data tampering impossible, in contrast to centralised databases that are easy targets for hackers. The blockchain may be seen as a decentralised, internet-based peer-to-peer network. In the 1980s and 1990s, the foundational principles for

what would become blockchain technology were developed. The 1989 introduction of Lamport's Paxos protocol is a consensus technique for coordinating group decisions among computers in a distributed environment where certain nodes or the network as a whole, may be unreliable [4]. With the introduction of digital signatures in 1991, a signed chain of information was used as an electronic ledger to demonstrate that no documents in a collection had been altered [5]. In 2008, these concepts were combined and applied to electronic money, and in 2009, the Bitcoin blockchain network was founded. Most modern cryptocurrency schemes may trace their origins to [5] (although with variations and modifications). There will be several applications for blockchain technology outside of bitcoin. The blockchain's basic characteristics, which include immutability, security, decentralisation, tamper-resistance, and distributed consensus, improved Bitcoin's credibility. Bitcoin's quick ascent to fame as a cryptocurrency aroused the curiosity of researchers, who started investigating the blockchain technology on which Bitcoin is founded. Since then, an increasing number of studies have examined potential applications for blockchain technology [6].

II. Literature Review

In [6], the traditional Autoencoder structure is offered as the core element, and the anomaly detection process along with the main features is described in depth. Multiple separate layers compose the encoder and decoder of an autoencoder. The encoder must first compress the original data before creating a compressed version. The decoder is responsible for reversing the encoding process and transforming the compressed representation back into its original, expanded form. Thus, the difference between the original and the restored representation is referred to as the reconstruction error. The learning process is designed to minimise reconstruction error throughout profiling time. Important to the output of a neural network is its activation function; common activation units are ReLU and tanh. Based on the Autoencoder reconstruction errors derived from normal samples, anomaly detection was able to conclude that data points with exceptionally high error rates did not belong to the normal class.

Two machine learning algorithms were developed by the authors: One-Class Support Vector Machines (OCSVM) for detecting outliers and K-Means for grouping anomalies of the same sort [7]. By optimising the distance of this hyperplane from the origin, One-Class SVM is able to successfully divide all data points away from the origin. Consequently, the output is a binary function reflecting the input space's regions and the data's matching probability densities.

Anomaly detection systemthat allows users to detect anomalous transactions and prevent them from being further spread was proposed by [8]. The system guarantees the following features:

i. Spread Across (thus avoiding any central point of failure).
ii. That it cannot be manipulated (making it not possible for malicious software to remove or alter its own traces).
iii. One may depend on it (any behavioural data is collected and verified by the majority of the network).
iv. No outsider can access or keep your personal information.

They demonstrated the feasibility of their proposed strategy compared to the typical eclipse attack, and their results provide a more thorough analysis of their system and a preliminary validation of its performance over a toy network.

A fixed-size user-centric subgraph is created from the full graph made up of all the transactions, and this subgraph is then utilised as the foundation for an anomaly detection method [9].The authors reduced the execution time by utilising a subgraph structure that is suitable for Graph Processing Unit (GPU), such that all subgraph construction, feature extraction, and anomaly detection are performed on the GPU. In terms of accuracy, the true positive rate was substantially large.The difficulty of identifying anomalies across many classes is a significant shortcoming of this research.

A security system based on the examination of blockchain network traffic statistics (rather than ledger data) was provided by [10] to identify dangerous events, through the functions of data collection and anomaly detection. Periodically, the data collection engine monitors the blockchain's underlying traffic and generates multidimensional data streams. The anomaly detection engine then discovers anomalies from the created data instances based on semi-supervised learning, which is capable of identifying previously undiscovered patterns, and they describe their profiling-based detection engine built on top of AutoEncoder (AE). Their experimental results demonstrate that the offered security method is effective for the accurate, real-time detection of malicious events in blockchain network traffic data.

Using symmetric and asymmetric Blockchain technology in computer and engineering science, [11] created a novel, more effective way for detecting Bitcoin abnormalities. This investigation used a collective anomaly technique. As opposed to examining specific addresses and wallets, the characteristics of people were investigated. In addition to collective anomaly detection, the trimmed K-means clustering algorithm was used. The experimental findings of this study indicate that customers using several wallets were more likely to encounter abnormalities. In contrast to previous research, which identified a maximum of 7 addresses in 5

instances of fraud, the authors' methodology revealed 14 individuals who had committed fraud, along with 26 addresses in 9 instances.

III. SYSTEM DESIGN

System design is the process of designing the elements of a system such as the architecture, modules, and components, the different interfaces of those components, and the data that goes through the system. The architectural design shown in figure 1 below presents the high-level components that make up the system.

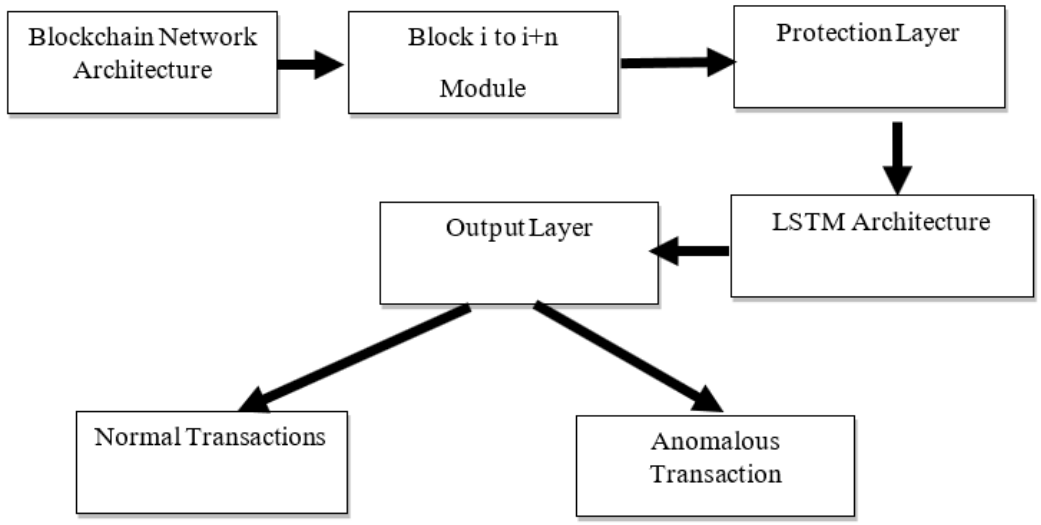

Figure 1: Architectural Design of the System

Blockchain Network: Each node in the blockchain network represents a certain transaction type. There are two primary sorts of transaction patterns on the blockchain. In this context, they are:

i. **Normal Transaction Pattern:** Regular transaction patterns on the blockchain are those that adhere to the standards for executing financial transactions.

ii. Anomalous Transaction Pattern: These are transactions that deviate from the regular blockchain transaction protocol.

Block in Blockchain: The blockchain stores cryptocurrency transactions irrevocably in "blocks," which are database structures. A block may include some or all of the most recent transactions that have not yet been validated by the network. The block is closed after the data's veracity has been confirmed. A new block is then produced to record and validate future transactions.

Protection Module: All blockchain transactions are handled via a protection module, and the processing time exceeds the approval interval. In this paper, Graph Representation Learning (GRL) is used to discover blockchain infrastructure anomalies. GRL is a subset of machine learning methods that enables a system to automatically learn the representations necessary for feature identification or classification. Learning the features (transactions), analysing network connections and connection patterns, and identifying anomalous behaviours all reduce the requirement for human feature engineering.

LSTM Architecture: Both long-term and short-term memory were used in the training of the model. Trading data will be used to train the LSTM model. The algorithm for Long Short-Term Memory (LSTM) is a kind of Recurrent Neural Network (RNN). Using the Keras tool, TensorFlow Framework will be utilised to generate the LSTM model. Our network is built using the Sequential API of Keras, which includes adding nodes and connecting them sequentially.

Using Embedding method, words (blockchain features) are encoded as 100-dimensional vectors. Pre-trained weights are supplied as an embedding parameter. To avoid updating the embeddings, trainable may be set to False.

A layer is devoted to masking the embeddings of words for which no embedding has been previously taught and is set to zero. This layer is skipped while training embeddings.

The primary processing unit of the network is an LSTM cell layer with dropout to prevent overfitting. Since we are using a single LSTM layer, no sequences are returned; this should be addressed when using two or more layers.

Output: The output represents the system's reaction to the supplied data. The output of the system might be normal or aberrant (which represents anomalous transactions or normal which represents normal transactions).

Model Training and Evaluation: Accuracy, Precision, Recall, and F1-Score were used to assess the efficiency of the LSTM model. Using a classification report and a confusion matrix, the effectiveness of the model was dissected in further depth. This was used to explain why the model sometimes predicted the test data accurately and sometimes erroneously. True positives, true negatives, false positives, and false negatives are all associated with accurate and inaccurate predictions.

3.1 Component Design

The component design is the breakdown of the component in the proposed system architecture. This is always needful because it shows further other sub-components that were not made known in the design of the system architecture. Figure 2 shows thesub-components of the LSTM architecture.

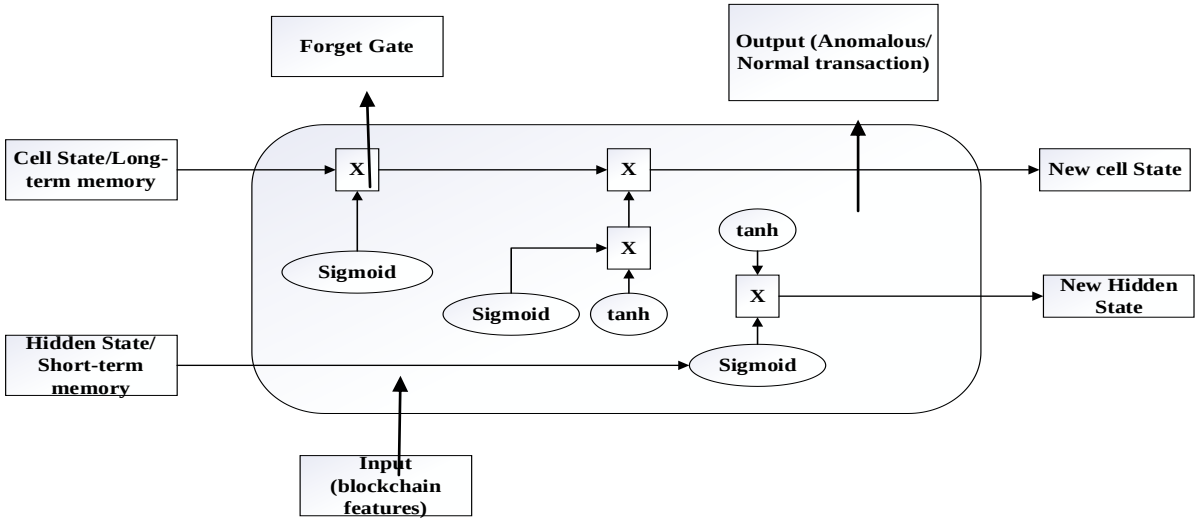

Figure 2: Component design of LSTM

Figure 2 is a pictorial representation of the LSTM model designed specifically for this study. The LSTM model consist of the Cell state, hidden layer, input layer, activation functions (sigmoid and tanh) and the output layer. Further explanation of the LSTM architecture can be seen as follows:

Forget Gate: establishes which information has to be forgotten $f^{(t)}$, by applying a sigmoid over the previously hidden state $h^{(t-1)}$ and input $x^{(t)}$

Input Gate: decides what new information should be stored in the cell state $c^{(t)}$ by applying a sigmoid over the previously hidden state $h^{(t-1)}$ and input $x^{(t)}$to decide which values to update $i^{(t)}$, and by applying a tanh over the previously hidden state $h^{(t-1)}$ and input $x^{(t)}$to generate a vector of new candidate values $c^{(t)}$.

Cell State update: this is not a proper gate but it is the long-term state update operation which is done by combining the state $c^{(t-1)}$multiplied by the output of the forget gate $f^{(t)}$, and the output of the input gates $i^{(t)}$ and $c^{(t)}$ multiplied together.

Output Gate: decides the output c_t by applying again a sigmoid over the previously hidden state $h^{(t-1)}$ and input $x^{(t)}$, and by multiplying the obtained output by the updated long-term state $c^{(t)}$.

Note: All notations and expressions used here are further defined and explained in Table 1

3.2 Concepts and Notations used

This sub-section introduces the notations and expressions used in building a model for the detection and prevention of anomalies in the blockchain. The various notations and expressions utilized by Graph Representation Learning for the detection of anomalies and that of Long Short-Term Memory (LSTM) for the presentation of Anomalies in blockchain can be seen in Table 1.

Table 1: Concept and Notations In The Model Design

S/N	Symbol	Notation	Definition
1	i_t	represents input the input gate.	The input gate decides which of the blockchain information is to be stored in the new state (new transformed inputs)
2	f_t	represents the forget gate	The forget gate is used to forget some of the inputs that are not needed in the blockchain
3	c_t	represents the output gate	The output gate is used to display results (if the blockchain transaction is anomalous or fraudulent).
4	W_t	Represents weighing matrix	The weighing matrix is used is used in measuring the weight of multiple inputs ($x_1 \ldots x_n$)
5	$h^{(t-1)}$	Represents the previous hidden state	The previous hidden state shows the values of the input featuresbefore transformation.
6	x^t	Denotes the input features of the blockchain	The input features can range from $x_{1 \ldots} x_n$. Where $n \geq 1$
7	$c^{(t-1)}$	Represents the previous state.	The precious state holds the inputs values before transforming it to the final output.
8	T	This represents the threshold	This is the value that determines if transaction in blockchain is anomalous.
9	B	Represents BlockHeaderHash	A block header is used to identify a particular block on an entire blockchain and is hashed repeatedly to create proof of work for mining rewards.
10	PB	PB represents the PreviousBlockHash	The "previous block hash" field is inside the block header and thereby affects the current block's hash.

Table 1 shows a breakdown of the notations and expressions used in for the detection and prevention of anomalies in blockchain. The symbols column shows the various symbols used.The notations show what meaning of the symbols, and the definition column shows a detailed description of the notations.

Algorithm For the Detection of Anomaly in Blockchain Transactions

This sub-section shows the steps used for the detection of anomalies in the blockchain.

Normal Operations in Blockchain Transactions.

i. For a blockchain transaction to be carried out successfully the following conditions must be met:
ii. The newly generated block height must match the previously generated block height
iii. The values of each individual block must match the previous value of the block
iv. The state in memory must be greater or equal to the amount that needs to be transacted

Pseudocode 1: Anomaly Detection Process

```
BH  //BlockHeaderHash
PB  // Previous BlockHeight
if (BH==PB):
F(N) = N <-BH <-PH
Repeat this process for all the nodes in Blockchain Network
Store the values of each of the nodes
Else(BH !=PB):
  Anomaly Block detected
  Delete PreviousBlockHashed
  End if
```

Algorithm 1: Anomaly Detection in Blockchain Transactions

```
1: Input: Training data, xTrain
2: Output: Threshold, t
3: model = fitGraphNeuralNetwork(xTrain)
4: for transaction in xTrain do
5:     reconstructTrans = modelPredict (transaction)
6:     lossTrain = meanSquaredLoss(transaction, reconstructedTrans)
7:     lossesTrain = append(lossTrain)
8:     mean = mean(lossesTrain)
```

```
9: Threshold, t = mean
10: end for
11: for transaction in xTest do
12:       reconstructedTrans = modelPredict (transaction)
13:       lossTest = meanSquaredLoss(transaction, reconstructedTrans)
14:       if lossTest> t then
15:                 y = abnormal
16:       end if
17: end for
```

Pseudocode 2: Anomaly Prevention Process

Steps to Prevent Anomalous Transactions in Blockchain
Step1. Ensure all blocks are checked and validated individually
Step2. Ensure no faulty nodes are sent across the network
Step3. Ensure the length of the reviver's address equals the length of the sender's address
The above steps can be accomplished using the algorithm below:

Algorithm For the Prevention of Anomaly in Blockchain Transactions

Algorithm 2: Anomaly Prevention in Blockchain Transactions

```
1: Input: Training data, xTrain
2: Output: Anomalous, Normal
3: model = LSTM(xTrain)
4: for transaction in xTrain do
5:       prediictTrans = modelPredict (transaction)
6:       lossTrain = errorVectors(xTrain- predictTrans)
7:       lossesTrain = append(lossTrain)
8:       For x in lossesTrain:
9           if(x≥ 1):
10.            display "Anomalous transaction found"
11.            block transaction process
12.         endif
13.    endFor
```

IV. RESULTS AND DISCUSSION

This paper presents two models for the detection and prevention of anomalies in blockchain technology. The result is made up of two phases. The first phase has to do with the identification of anomalous transactions on the blockchain using Graph representation Learning, and the second phase has to do with the use of Long-Short Term Memory for the prevention of anomaly.

4.1: Phase 1: Model Training for Anomaly detection

The model was trained using Graph Representation Learning for the detection of anomalies in the blockchain. The Graph Representation Learning (GRL) model was trained using the four layers. The first layer (input layer) contains an input neuron of 6. The second layer (hidden layer 1) contains an input neuro of 4, using an activation function of tanh. The third layer (hidden layer 2) contains an input neuron of 4 using an activation function of tanh, finally, the fourth layer the output layer used sigmoid as activation function. Other hyper parameters used in training the model are loss= mean_squared_error, optimizer=adma, drop_out=0.1, validation_size=20 epoch, 100 and batch_size=256. The model was evaluated using false positive, true positive, against accuracy. This can be seen in the Figure3. The evaluation report of the model using accuracy can be seen in. Finally, the visualized result of the detected anomaly of the test data (block chain transactions can be seen in Figure 4).

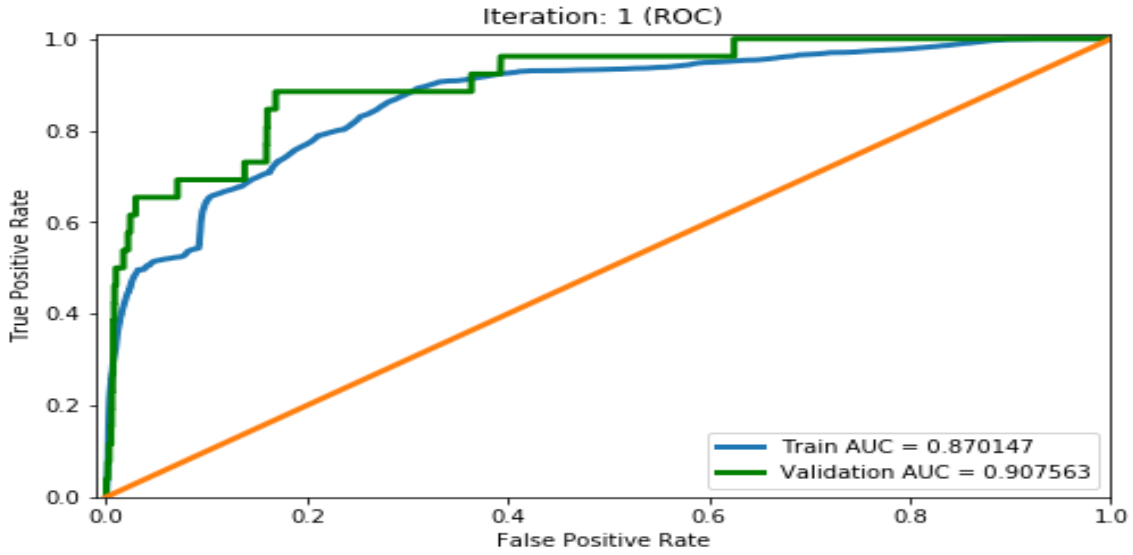

Figure 3: Accuracy result For training and validation test for the first Iterator

Figure 3 shows the number of true positive result of the model and the number of false positive result, with a training and validation accuracy.

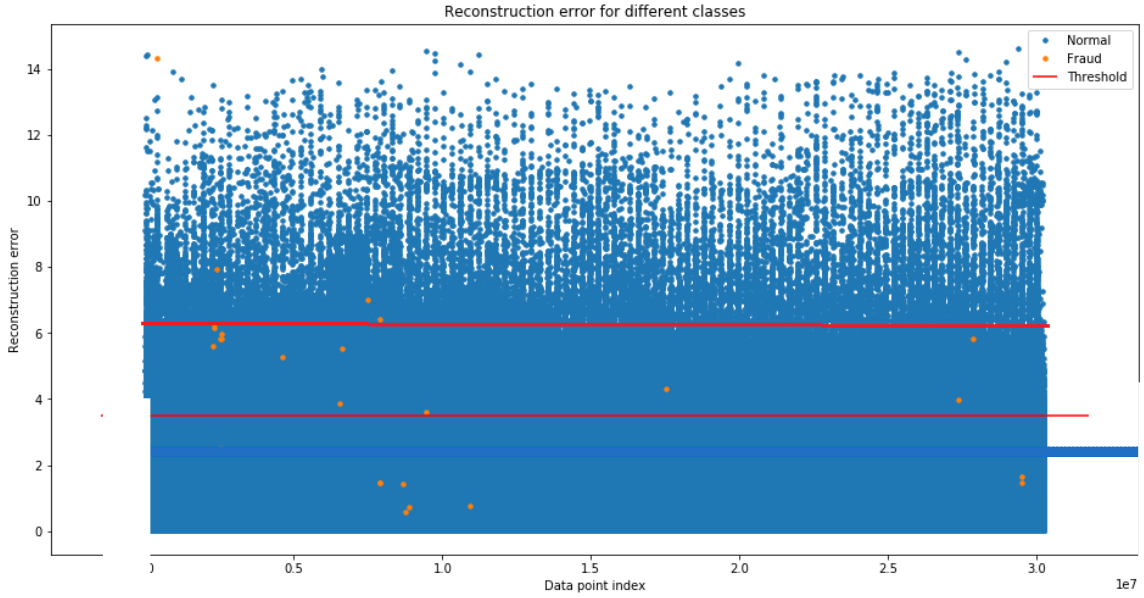

Figure 4: Visualized Result of For Anomaly Detection

The figure shows the result model in detecting anomalies in blockchain network. From the figure, a threshold function was defined using the mean square error of the outcome of the actual result minus the predicted result. From the figure the minimum threshold function is 6. Therefore, anything more than 6, shows an anomaly.

4.2 Phase 2: Model Training for Anomaly Prevention

The model was trained using Long-Short Term Memory. The LSTM model was trained using the four layers. The first layer contains an input neuron of 64 and used relu as activation function. The second layer contains an input neuron of 64, and activation function of tanh. The third layer contain an input neuron of 128, and an activation function of relu, and finally the fourth layer being the output layer used sigmoid as activation function. Other hyper parameters used in training the model are loss= categorical, optimizer=adam, epoch, 20 and batch_size=128. The training results display the mean squared error for both training and validation tests. Figures 5, and 6 show the graphical analysis of the model's performance using accuracy and loss for the first 20 steps.

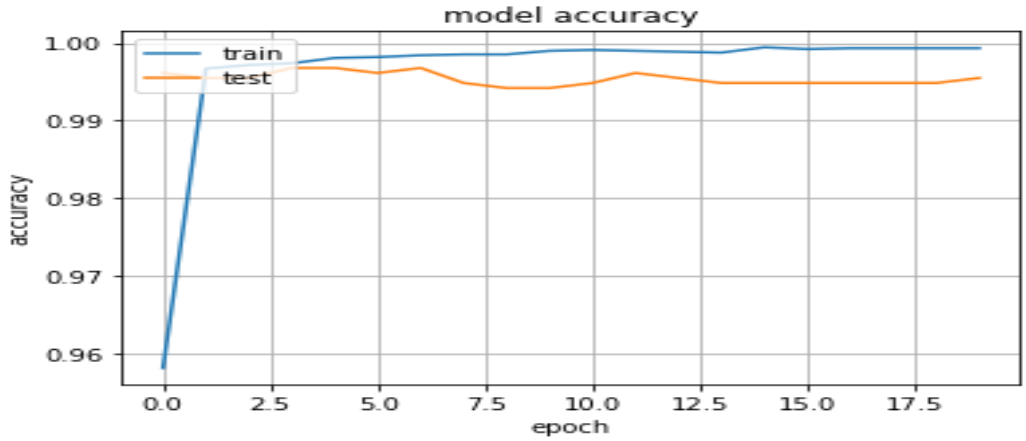

Figure 5: Accuracy for both Training and Testing Data.

This shows the accuracy of the LSTM model for the prevention of anomalies in blockchain network. The result shows that the model achieved both training and validation accuracy of about 99.9%. The accuracy represents the performance of the model, while the epoch represents the number of steps in which the model is trained.

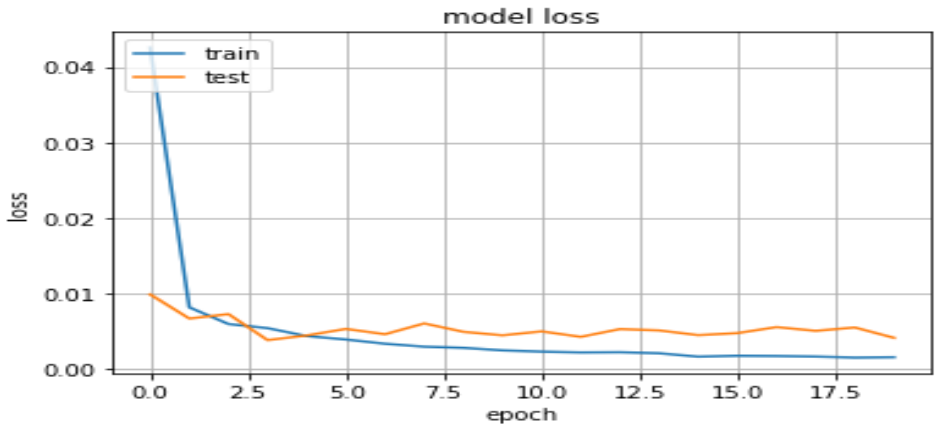

Figure 6: Loss values for both Training data and Testing Data

This shows the accuracy of the LSTM model for the prevention of anomalies in blockchain network. The result shows that the model had both training and validation lossbelow 0.01%.

Figure 7: Web-based System for transaction Testing

Here, the model was tested on some parameters of the blockchain for the detection and prevention of anomalies. Certain inputs were given such as receiver's address, sender's address, block height, and amount.

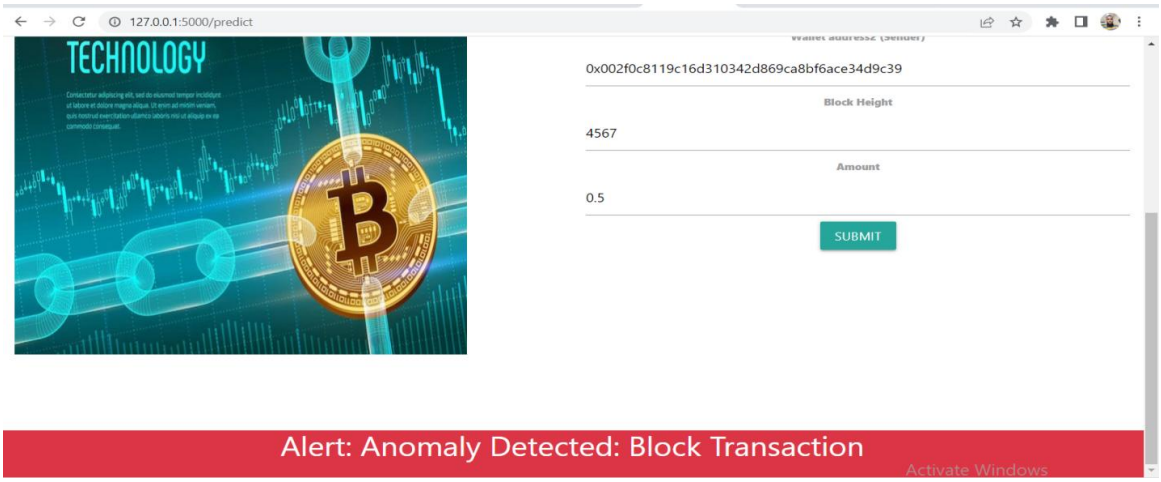

Figure 8:Anomaly Prevention

This shows the result of the model when certain parameters were given as input. The results show how the model detected the anomaly and stopped the transaction from being completed.

V. CONCLUSION

This system provides a model for the detection and prevention of anomalies in blockchain transactions. The system starts by making use of a cryptocurrency dataset. The dataset was pre-processed by checking for null or empty values. The processed features were used in building a Graph Representation Learning Model (GRL) for the detection and prevention of anomalies on blockchain. The GRL model achieved a detection accuracy of 98.9%. After the detection of anomalous transactions, A Long Short-Term Memory (LSTM) algorithm was used for the prevention of anomalous transactions on blockchain. The LSTM model achieved an accuracy result of 99% in preventing anomalous transactions for completing.

REFERENCES

[1]. Hassan, M.U., Rehmani, M.H., & Chen, J. (2021). Anomaly Detection in Blockchain Networks: A Comprehensive Survey. ArXiv, abs/2112.06089.
[2]. Simanta, S. S. (2018). Understanding Blockchain Technology. Computer Science and Engineering. 8(2), 23-29, DOI: 10.5923/j.computer
[3]. Crosby, M., Pattanayak, P., Verma, S., & Kalyanaraman, V. (2016). Blockchain Technology: Beyond Bitcoin. International Journal of Applied Innovation, 2, 6-10
[4]. Dewanta, F., Mambo, M., (2020). Bpt scheme: Establishing trusted vehicular fog computing service for rural area based on blockchain approach. IEEE Transactions on Vehicular Technology, 70(2). 1752– 1769.
[5]. Narayanan, A., Bonneau, J., Felten, E., Miller, A., and Goldfeder, S. (2016). Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton University Press.
[6]. M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani (2018). Applications of blockchains in the Internet of Things: A comprehensive survey. IEEE Communications Surveys & Tutorials, 21(2), 1676–1717
[7]. Torres, C. F., Steichen, M. (2019). The art of the scam: Demystifying honeypots in ethereum smart contracts. in 28th {USENIX} Security Symposium. USENIX Security 19, 1591–1607.
[8]. Simanta, S. S. (2018). Understanding Blockchain Technology. Computer Science and Engineering. 8(2), 23-29
[9]. Shin M. (2020). Scalable Anomaly Detection Method for Blockchain Transactions using GPU. 10th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), 163-168
[10]. Weller-Fahy D. J., Borghetti B. J.& Sodemann A. A. (2015). A survey of distance and similarity measures used within network intrusion anomaly detection. IEEE Communications Surveys Tutorials, 17(1), 70–91
[11]. Signorini, M., Pontecorvi, M., Kanoun, W., & Di Pietro, R. (2020). BAD: A blockchain anomaly detection solution. IEEE Access, 8, 173481-173490.