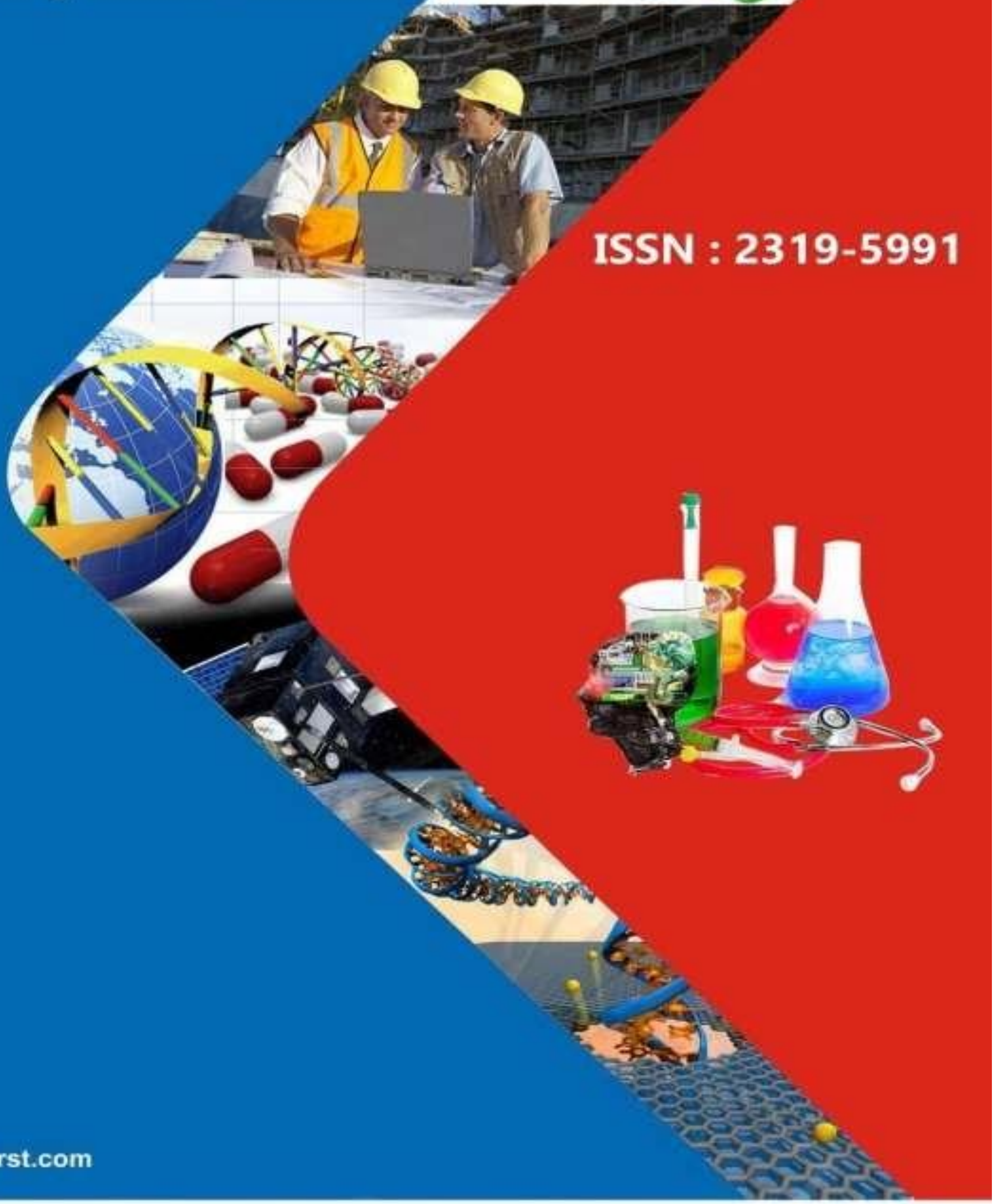
IJERST

International Journal of Engineering Research and Science & Technology

ISSN : 2319-5991



www.ijerst.com

Email: editor@ijerst.com or editor.ijerst@gmail.com

ISSN 2319-5991 www.ijerst.com
Vol. 16, Issue.1, Jan 2023

BLOCK HUNTER FEDERATED LEARNING FOR CYBER THREAT HUNTING IN BLOCKCHAIN-BASED IIOT NETWORKS

Mrs K. Manjula[1],M. Shruthi[2],B. Srujana[3],B. Anurvini[4],T. Sirisha[5],G. Snehalatha[6]
[1]Assistant professor, Department of CSE, Princeton College of engineering and technology for women
Narapally vijayapuri colony ghatkesar mandal, Pin code-500088
[2,3,4,5,6]UG Students,Department of CSE, Princeton College of engineering and technology for women
Narapally vijayapuri colony ghatkesar mandal, Pin code-500088

ABSTRACT

Nowadays, blockchain-based technologies are being developed in various industries to improve data security. In the context of the Industrial Internet of Things (IIoT), a chain-based network is one of the most notable applications of blockchain technology. IIoT devices have become increasingly prevalent in our digital world, especially in support of developing smart factories. Although blockchain is a powerful tool, it is vulnerable to cyber attacks. Detecting anomalies in blockchain-based IIoT networks in smart factories is crucial in protecting networks and systems from unexpected attacks. In this paper, we use Federated Learning (FL) to build a threat hunting framework called Block Hunter to automatically hunt for attacks in blockchain-based IIoT networks. Block Hunter utilizes a cluster-based architecture

for anomaly detection combined with several machine learning models in a federated environment. To the best of our knowledge, Block Hunter is the first federated threat hunting model in IIoT networks that identifies anomalous behavior while preserving

privacy. Our results prove the efficiency of the Block Hunter in detecting anomalous activities with high accuracy and minimum required bandwidth.

I.INTRODUCTION

THE technological trajectory of block chain makes it a valuable tool in many areas, including healthcare, military, finance and networking, via its immutable and tamperproof data security advantages. With the ever-increasing use of Industrial Internet of Things (IIOT) devices, the world is inevitably becoming a smarter interconnected environment; especially factories are becoming more intelligent and efficient as technology advances [1]. IIOT is considered a subcategory of the Internet of Things (IOT). There are, however, differences between IOT and IIOT in terms of security requirements. While the IIOT makes consumers' lives easier and more convenient, the IIOT aims to increase production safety and efficiency. IIOT devices are mainly used in B2B (business-to-business) settings, while IOT devices are mostly considered in B2C (business-to consumer) environments. This would lead to a different threat profile for IIOT networks compared to their IOT counterparts where device-to-device transactions are of utmost importance.

IIOT networks provide an umbrella for supporting many applications and arm us to respond to users' needs, especially in an industry setting such as smart factories [1]. Block chain technology advantages lead to its wide adoption in IIOT based networks such as smart factories, smart homes/buildings, smart farms, smart cities, connected drones, and healthcare systems [1], [2]. While the focus of this paper is on the security of block chain-based IIOT networks in smart factories [3], [4], the suggested framework may be used in other IIOT settings as well.

In modern smart factories, many devices are connected to the public networks, and many activities are supported by smart systems such as temperature monitoring systems, Internet-enabled lights, IP cameras, and IP phones. These devices are storing private and sensitive data and may offer safety-critical services [3], [1]. As the number of IIOT devices in smart factories increases, the main issue will be storing, collecting, and sharing data securely. Industrial, critical, and personal data are therefore at risk in such a situation. Block chain technology can ensure data integrity inside and outside of smart factories through strong authentication and ensure the availability of communication backbones. Despite this, privacy and security issues are significant challenges in IIOT [3], [4]. The probability of fraudulent activity occurring in block chain-based networks [2], [4] is an important issue. Even though block chain technology is a powerful tool, it is not protected from cyber attacks either. For example, a 51% cyber attack [2] on Ethereum Classic, and three consecutive attacks in August of 2020 [5], which resulted in the theft of over $5M worth of crypto currency, have exposed the vulnerabilities of this block chain network.

Smart factories should protect users' data privacy during transmission, usage, and storage [4]. Stored data are vulnerable to tampering by fraudsters seeking to access, alter or use the data with malicious motives. Statistically speaking, these attacks can be viewed as anomalous events, exhibiting a strong deviation from usual behavior [2], [6]. Detecting out-of-norm events are essential for threat hunting programs and protecting systems from unauthorized access by automatically identifying and filtering anomalous activities. [6], [7].

The main objective of this paper is to detect suspicious users and transactions in a block chain-based IIOT network specifically for smart factories. Here, abnormal behavior serves as a proxy for suspicious behavior as well [4]. By identifying outliers and patterns, we can leverage Machine Learning (ML) algorithms to identify out-of-norm patterns to detect attacks and anomalies on block chain. Because deep neural networks learn representations automatically from data that they are trained on, they are the candidate solution for detecting anomalies [4], [7]. However, there are challenges with any ML and deep learning-based anomaly detection techniques. These methods suffer from training data scarcity problems, and privacy issues [7].

Detecting anomalies in the block chain is a complicated issue [8]. Not only each block needs to be sent to a central server, which increases the training time, but also the model requires new block data in the testing phase [8]. In addition, when ML models are frequently updated to respond to new threats and detect anomalies, malicious adversaries can launch causative/data poisoning attacks to degrade the ML model deliberately. Attackers may intentionally send crafted payloads to evade anomaly detection.

A novel and practical approach would be to employ Federated learning (FL) models to detect anomalies while preserving data privacy, and monitoring data quality [7], [9]. FL allows edge devices to collaborate during the training stage while all data stays on the device. We can train the model on the device itself instead of sending the data to another place, and only the updates of the model are shared across the network.

FL has become a trend in ML where smart edge devices can simultaneously develop a mutual prediction between each other [7], [10]. In addition, FL ensures multiple actors construct robust machine learning models without sharing data, addressing fundamental privacy, data security, and digital rights management challenges. Considering these characteristics, this paper uses an FL-based anomaly-detection framework called Block Hunter capable of detecting attack payloads in block chain-based IIOT networks

The main contributions of the paper are summarized as follows:

ISSN 2319-5991 www.ijerst.com
Vol. 16, Issue.1, Jan 2023

1) Utilize a cluster-based architecture to formulate an anomaly detection problem in block chain-based smart factories. The cluster-based approach increase hunting efficiency in terms of bandwidth reduction and throughput in IIoT networks.

2) Apply a federated design model to detect anomalous behaviour in IIoT devices related to blockchain-based smart factories. This provides a privacy-preserving feature when using machine learning models in a federated framework.

3) Implementation of various anomaly detection algorithms such as clustering-based, statistical, subspace-based, classifier-based, and tree-based for efficient anomaly detection in smart factories.

4) The impact of block generation, block size, and miners on the Block Hunter framework are considered. Moreover, the performance measurements like Accuracy, Precision, Recall, F1-score, and True Positive Rate (TPR) anomaly detection are discussed.

Here is a breakdown of the rest of the paper. Section II discusses anomaly detection works in the block chain and FL. Section III describes the Block Hunter framework and presents the network model and topology design. In Section
IV, methodology and machine learning approaches to identify anomalies are discussed. In Section V, we present the assessment of the Block Hunter framework. Finally, In Section VI, we conclude the paper and point out future work directions.

II.EXISTING SYSTEM

The research by Sayadi et al. [15] proposes an algorithm for anomaly detection over bitcoin electronic transactions. They examined the One-Class Support Vector Machines (OCSVM) and the K-means algorithms to group outliers similar in both statistical significance and type. They analyzed their work by generating detection results and found that we could obtain high-performing results on accuracy.

In [16], the authors suggested an approach based on the semantics of anomalies in blockchain-based IoT Networks. A method was presented to detect anomalous behavior in blockchain that gathers metadata in forks to determine mutual informational recognition of anomalous activity. They developed a tool that improves blockchain security and connected devices. Also, in [17], has introduced encoder-decoder deep learning regression for detecting blockchain security. This work developed an

ISSN 2319-5991 www.ijerst.com
Vol. 16, Issue.1, Jan 2023

anomaly detection framework that relies on aggregate information derived from bitcoin blockchain monitoring. Their experiments have demonstrated that their model can detect publicly reported attacks using the historical logs of the Ethereum network.

Chai et al. [22] proposed a hierarchical blockchain framework and FL to learn and share environmental data. This framework is functional and efficient for large-scale vehicular networks. FL-based learning meets the Internet of Vehicles' distributed pattern and privacy requirements. Sharing behavior is modeled as a multi-leader, multi-player trading market process to stimulate knowledge sharing. Simulated results indicate that an algorithm based on hierarchical structures can enhance sharing, learning, and managing specific malicious attacks. Furthermore, the authors in [23] deliver a comprehensive investigation on how FL could supply better cybersecurity and prevent various cyberattacks in real-time. This work highlights some main challenges and future directions on which the researchers can focus for adopting FL in real-time scenarios.

Disadvantages

- ❖ The system is not implemented the Isolation Forest (IF) model which falls under the Tree-based anomaly detection algorithms category.
- ❖ The system is not implemented Cluster-Based Local Outlier Factor.

III.PROPOSED SYSTEM

Detecting anomalous activities is a significant contributor to automatically protecting a system from unexpected attacks. Anomalies in blockchain must be detected by sending each block of data to a central server for each block update. This is not efficient and also imposes privacy concerns. FL solutions are promising in tackling this issue. We use FL to update the model frequently and to obtain a global model for detecting an anomaly. After learning about each smart factory's data, devices, and service provider, the model's parameters will be sent to the parameter server for aggregation and to update our general model.

Cluster based architecture provides more efficient use of resources and throughput during the blockchain run in each smart factory. Clustering reduces the computational complexity in the creation of the underlying network through a hierarchical approach.

ISSN 2319-5991 www.ijerst.com
Vol. 16, Issue.1, Jan 2023

Advantages

Federation Construction: The subset of smart factory members, cluster, selected to receive the model locally.

Decentralized Training: When a cluster of smart factories is selected, it updates its model using its local data.

Model Accumulation: Responsible for accumulating and merging the data models. Data is not sent and integrated from the federation to the server individually.

Model Aggregation (FedAvg): Parameter server aggregates model weights to compute an enhanced global model.

IV.MODULES

Service Provider

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as

Login, Train & Test IIOT Network Datasets, View Trained and Tested IIOT Network Datasets Accuracy in Bar Chart, View Trained and Tested IIOT Network Datasets Accuracy in Bar Chart, View Prediction Of Cyber Threat Hunting Type, View Cyber Threat Hunting Type Ratio, Download Predicted Data Sets, View Cyber Threat Hunting Type Ratio Results, View All Remote Users.

View and Authorize Users

In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, user name, email, address and admin authorizes the users.

Remote User

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like register and login, predict cyber threat hunting type, view your profile.

V.ALGORITHMS

Decision tree classifiers

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such

ISSN 2319-5991 www.ijerst.com
Vol. 16, Issue.1, Jan 2023

generation based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as follows:

Step 1. If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labeled with this class
Step 2. Otherwise, let T be some test with possible outcomes O1, O2,…, On. Each object in S has one outcome for T so the test partitions S into subsets S1, S2,… Sn where each object in Si has outcome Oi for T. T becomes the root of the decision tree and for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

Gradient boosting

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.[1][2] When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

K-Nearest Neighbors (KNN)

- Simple, but a very powerful classification algorithm

- Classifies based on a similarity measure
- Non-parametric
- Lazy learning
- Does not “learn” until the test example is given

- Whenever we have a new data to classify, we find its K-nearest neighbors from the training data

Example

- Training dataset consists of k-closest examples in feature space
- Feature space means, space with categorization variables (non-metric variables)
- Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset

Logistic regression Classifiers

ISSN 2319-5991 www.ijerst.com
Vol. 16, Issue.1, Jan 2023

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name *logistic regression* is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name *multinomial logistic regression* is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

Naïve Bayes

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature .

Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).

ISSN 2319-5991 www.ijerst.com
Vol. 16, Issue.1, Jan 2023

While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique.

Thus, we introduce in a new presentation of the results of the learning process. The classifier is easier to understand, and its deployment is also made easier. In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra. We compare the obtained results (the parameters of the model) to those obtained with other linear approaches such as the logistic regression, the linear discriminant analysis and the linear SVM. We note that the results are highly consistent. This largely explains the good performance of the method in comparison to others. In the second part, we use various tools on the same dataset (Weka 3.6.0, R 2.9.2, Knime 2.1.1, Orange 2.0b and RapidMiner 4.6.0). We try above all to understand the obtained results.

Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho[1] using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.).The extension combines Breiman's "bagging" idea and random selection of features, introduced

ISSN 2319-5991 www.ijerst.com
Vol. 16, Issue.1, Jan 2023

first by Ho[1] and later independently by Amit and Geman[13] in order to construct a collection of decision trees with controlled variance.

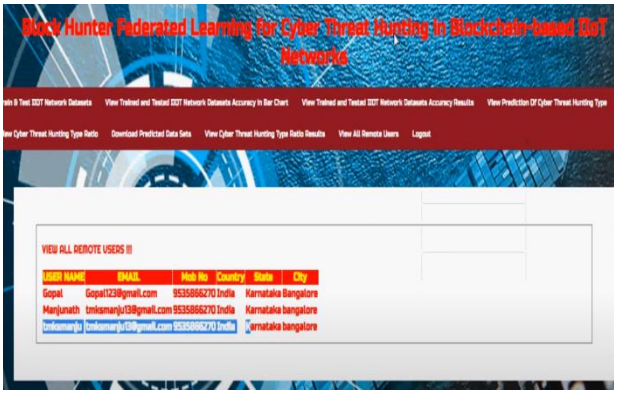
Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

SVM

In classification tasks a discriminant machine learning technique aims at finding, based on an *independent and identically distributed* (*iid*) training dataset, a discriminant function that can correctly predict labels for newly acquired instances. Unlike generative machine learning approaches, which require computations of conditional probability distributions, a discriminant classification function takes a data point *x* and assigns it to one of the different classes that are a part of the classification task. Less powerful than generative approaches, which are mostly used when prediction involves outlier detection, discriminant approaches require fewer computational resources and less training data, especially for a multidimensional feature space and when only posterior probabilities are needed. From a geometric perspective, learning a classifier is equivalent to finding the equation for a multidimensional surface that best separates the different classes in the feature space.

SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to *genetic algorithms* (*GAs*) or *perceptrons*, both of which are widely used for classification in machine learning. For perceptrons, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptrons is only to minimize error during training, which will translate into several hyperplanes' meeting this requirement.

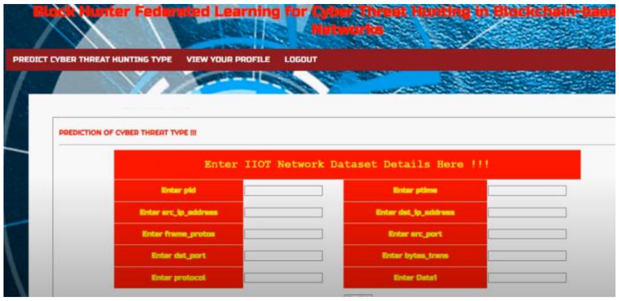
VI.RESULTS


Block Hunter Federated Learning for Cyber Threat Hunting in Blockchain-based IIoT Networks
rain & Test IIOT Network Datasets
View Trained and Tested IIOT Network Datasets Accuracy in Bar Chart
View Trained and Tested IIOT Network Datasets Accuracy Results
View Prediction Of Cyber Threat Hunting Type
iew Cyber Threat Hunting Type Ratio
Download Predicted Data Sets
View Cyber Threat Hunting Type Ratio Results
View All Remote Users
Logout

VIEW ALL REMOTE USERS !!!

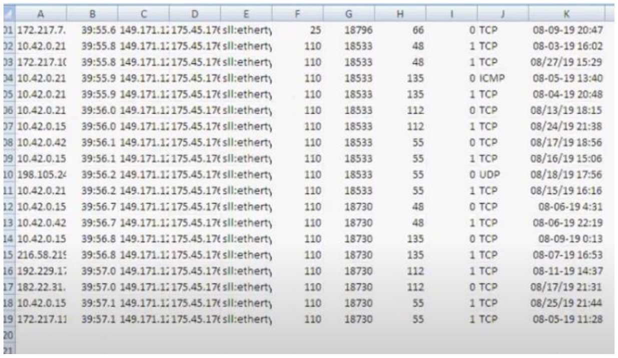
USER NAME	EMAIL	Mob No	Country	State	City
Gopal	Gopal123@gmail.com	9535866270	India	Karnataka	Bangalore
Manjunath	tmksmanju13@gmail.com	9535866270	India	Karnataka	bangalore
tmksmanju	tmksmanju13@gmail.com	9535866270	India	Karnataka	bangalore

ISSN 2319-5991 www.ijerst.com
Vol. 16, Issue.1, Jan 2023

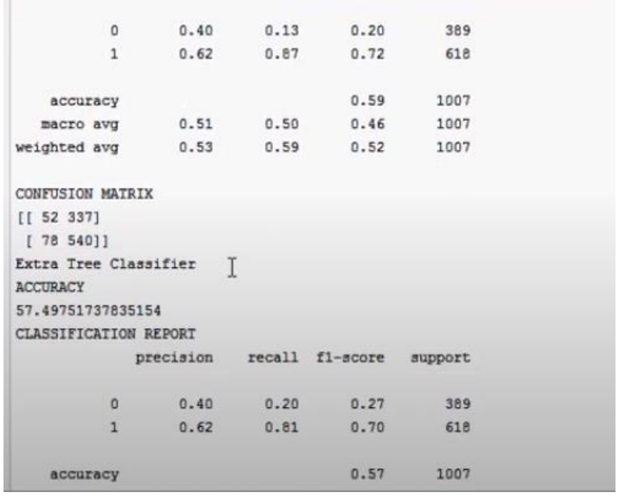

Block Hunter Federated Learning for Cyber Threat Hunting in Blockchain-base
Networks
PREDICT CYBER THREAT HUNTING TYPE
VIEW YOUR PROFILE
LOGOUT

PREDICTION OF CYBER THREAT TYPE !!!

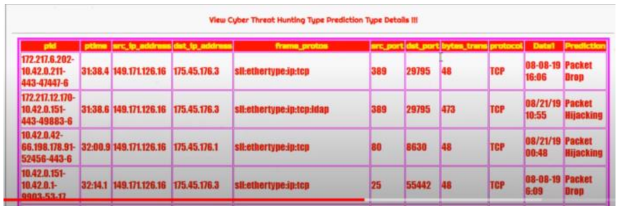
Enter IIOT Network Dataset Details Here !!!			
Enter pid		Enter ptime	
Enter src_ip_address		Enter dst_ip_address	
Enter frame_protos		Enter src_port	
Enter dst_port		Enter bytes_trans	
Enter protocol		Enter Data1	



	A	B	C	D	E	F	G	H	I	J	K
01	172.217.7.	39:55.6	149.171.1:	175.45.17(	sll:etherty	25	18796	66	0	TCP	08-09-19 20:47
02	10.42.0.21	39:55.8	149.171.1:	175.45.17(	sll:etherty	110	18533	48	1	TCP	08-03-19 16:02
03	172.217.1(	39:55.8	149.171.1:	175.45.17(	sll:etherty	110	18533	48	1	TCP	08/27/19 15:29
04	10.42.0.21	39:55.9	149.171.1:	175.45.17(	sll:etherty	110	18533	135	0	ICMP	08-05-19 13:40
05	10.42.0.21	39:55.9	149.171.1:	175.45.17(	sll:etherty	110	18533	135	1	TCP	08-04-19 20:48
06	10.42.0.21	39:56.0	149.171.1:	175.45.17(	sll:etherty	110	18533	112	0	TCP	08/13/19 18:15
07	10.42.0.15	39:56.0	149.171.1:	175.45.17(	sll:etherty	110	18533	112	1	TCP	08/24/19 21:38
08	10.42.0.42	39:56.1	149.171.1:	175.45.17(	sll:etherty	110	18533	55	0	TCP	08/17/19 18:56
09	10.42.0.15	39:56.1	149.171.1:	175.45.17(	sll:etherty	110	18533	55	1	TCP	08/16/19 15:06
10	198.105.2	39:56.2	149.171.1:	175.45.17(	sll:etherty	110	18533	55	0	UDP	08/18/19 17:56
11	10.42.0.21	39:56.2	149.171.1:	175.45.17(	sll:etherty	110	18533	55	1	TCP	08/15/19 16:16
12	10.42.0.15	39:56.7	149.171.1:	175.45.17(	sll:etherty	110	18730	48	0	TCP	08-06-19 4:31
13	10.42.0.42	39:56.7	149.171.1:	175.45.17(	sll:etherty	110	18730	48	1	TCP	08-06-19 22:19
14	10.42.0.15	39:56.8	149.171.1:	175.45.17(	sll:etherty	110	18730	135	0	TCP	08-09-19 0:13
15	216.58.21	39:56.8	149.171.1:	175.45.17(	sll:etherty	110	18730	135	1	TCP	08-07-19 16:53
16	192.229.1	39:57.0	149.171.1:	175.45.17(	sll:etherty	110	18730	112	1	TCP	08-11-19 14:37
17	182.22.31.	39:57.0	149.171.1:	175.45.17(	sll:etherty	110	18730	112	0	TCP	08/17/19 21:31
18	10.42.0.15	39:57.1	149.171.1:	175.45.17(	sll:etherty	110	18730	55	1	TCP	08/25/19 21:44
19	172.217.1	39:57.1	149.171.1:	175.45.17(	sll:etherty	110	18730	55	1	TCP	08-05-19 11:28
20											
21											

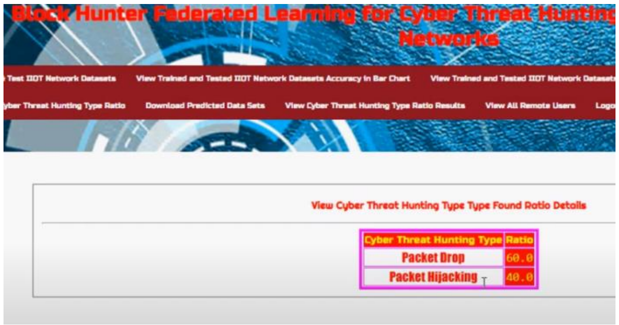


```
           0       0.40      0.13      0.20       389
           1       0.62      0.87      0.72       618

    accuracy                           0.59      1007
   macro avg       0.51      0.50      0.46      1007
weighted avg       0.53      0.59      0.52      1007

CONFUSION MATRIX
[[ 52 337]
 [ 78 540]]
Extra Tree Classifier
ACCURACY
57.49751737835154
CLASSIFICATION REPORT
              precision    recall  f1-score   support

           0       0.40      0.20      0.27       389
           1       0.62      0.81      0.70       618

    accuracy                           0.57      1007
```



View Cyber Threat Hunting Type Prediction Type Details !!!

pid	ptime	src_ip_address	dst_ip_address	frame_protos	src_port	dst_port	bytes_trans	protocol	Data1	Prediction
172.217.6.202-10.42.0.211-443-47447-6	31:38.4	149.171.126.16	175.45.176.3	sll:ethertype:ip:tcp	389	29795	48	TCP	08-08-19 16:06	Packet Drop
172.217.12.170-10.42.0.151-443-49883-6	31:38.6	149.171.126.16	175.45.176.3	sll:ethertype:ip:tcp:ldap	389	29795	473	TCP	08/21/19 10:55	Packet Hijacking
10.42.0.42-66.198.178.91-52456-443-6	32:00.9	149.171.126.16	175.45.176.1	sll:ethertype:ip:tcp	80	8630	48	TCP	08/21/19 00:48	Packet Hijacking
10.42.0.151-10.42.0.1-9903-53-17	32:14.1	149.171.126.16	175.45.176.3	sll:ethertype:ip:tcp	25	55442	48	TCP	08-08-19 6:09	Packet Drop


Block Hunter Federated Learning for Cyber Threat Hunting
Networks
Test IIOT Network Datasets
View Trained and Tested IIOT Network Datasets Accuracy in Bar Chart
View Trained and Tested IIOT Network Datasets
yber Threat Hunting Type Ratio
Download Predicted Data Sets
View Cyber Threat Hunting Type Ratio Results
View All Remote Users
Logo

View Cyber Threat Hunting Type Type Found Ratio Details

Cyber Threat Hunting Type	Ratio
Packet Drop	60.0
Packet Hijacking	40.0

VII.CONCLUSION

In this paper, we developed the Block Hunter framework to hunt anomalies in block chain-based IIOT smart factories using a federated learning approach. Block Hunter uses a cluster-based architecture to reduce resources and improve the throughput of block chain-based IIOT networks hunting. The Block Hunter framework was evaluated using a variety of machine learning algorithms (NED, IF, CBLOF, K-means, PCA) to detect anomalies. We also examined the impacts of block generation interval, block size, and different miners on the performance of the Block Hunter. Using generative adversarial networks (GAN) to design and implement a block hunter like framework would be an interesting future research work. Furthermore, designing and applying IIOT-related block chain networks

ISSN 2319-5991 www.ijerst.com
Vol. 16, Issue.1, Jan 2023

with different consensus algorithms would also be worth investigating in the future.

VIII.REFERENCES

[1] J. Wan, J. Li, M. Imran, D. Li, and F. e Amin, “A blockchain-based
solution for enhancing security and privacy in smart factory,” IEEE
Transactions on Industrial Informatics, vol. 15, no. 6, pp. 3652–3660,
2019.

[2] F. Scicchitano, A. Liguori, M. Guarascio, E. Ritacco, and G. Manco,
“Blockchain attack discovery via anomaly detection,” Consiglio
Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni
(ICAR), 2019, 2019.

[3] Q. Xu, Z. He, Z. Li, M. Xiao, R. S. M. Goh, and Y. Li, “An effective
blockchain-based, decentralized application for smart building system
management,” in Real-Time Data Analytics for Large Scale Sensor Data.
Elsevier, 2020, pp. 157–181.

[4] B. Podgorelec, M. Turkanovi´c, and S. Karakatiˇc, “A machine learningbased
method for automated blockchain transaction signing including
personalized anomaly detection,” Sensors, vol. 20, no. 1, p. 147, 2020.

[5] A. Quintal, “Veriblock foundation discloses mess
vulnerability in ethereum classic blockchain,” VeriBlock
Foundation. [Online]. Available: https://www.prnewswire.
com/news-releases/veriblock-foundation-discloses-mess-vulnern
ability-in-ethereum-classic-blockchain-301327998.html

[6] M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang,
and D. Mohaisen, “Exploring the attack surface of blockchain: A
comprehensive survey,” IEEE Communications Surveys & Tutorials,
vol. 22, no. 3, pp. 1977–2008, 2020.

[7] R. A. Sater and A. B. Hamza, “A federated learning approach to anomaly
detection in smart buildings,” arXiv preprint arXiv:2010.10293, 2020.

[8] O. Shafiq, “Anomaly detection in blockchain,” Master’s thesis, Tampere
University, 2019.

[9] A. Yazdinejadna, R. M. Parizi, A. Dehghantanha, and H. Karimipour,
“Federated learning for drone authentication,” Ad Hoc Networks, p.
102574, 2021.

[10] D. Preuveneers, V. Rimmer, I. Tsingenopoulos, J. Spooren, W. Joosen,
and E. Ilie-Zudor, “Chained anomaly detection models for federated
learning: An intrusion detection case study,” Applied Sciences, vol. 8,
no. 12, p. 2663, 2018.

[11] L. Tan, H. Xiao, K. Yu, M. Aloqaily, and Y. Jararweh, “A blockchainempowered
crowdsourcing system for 5g-enabled smart cities,” Computer
Standards & Interfaces, vol. 76, p. 103517, 2021.

[12] L. Tseng, X. Yao, S. Otoum, M. Aloqaily, and Y. Jararweh, “Blockchainbased
database in an iot environment: challenges, opportunities, and
analysis,” Cluster Computing, vol. 23, no. 3, pp. 2151–2165, 2020.

[13] M. Signorini, M. Pontecorvi, W. Kanoun, and R. Di Pietro, “Bad:
a blockchain anomaly detection solution,” IEEE Access, vol. 8, pp.
173 481–173 490, 2020.

[14] S. Iyer, S. Thakur, M. Dixit, R. Katkam, A. Agrawal, and F. Kazi,
“Blockchain and anomaly detection based monitoring system for enforcing
wastewater reuse,” in 2019 10th International Conference on
Computing, Communication and Networking Technologies (ICCCNT).
IEEE, 2019, pp. 1–7.

[15] S. Sayadi, S. B. Rejeb, and Z. Choukair, “Anomaly detection model over
blockchain electronic transactions,” in 2019 15th International Wireless
Communications & Mobile Computing Conference (IWCMC). IEEE,
2019, pp. 895–900.

[16] Z. Il-Agure, B. Attallah, and Y.-K. Chang, “The semantics of anomalies
in iot integrated blockchain network,” in 2019 Sixth HCT Information
Technology Trends (ITT). IEEE, 2019, pp. 144–146.

[17] 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.

[18] T. R. Gadekallu, Q.-V. Pham, D. C. Nguyen, P. K. R. Maddikunta,
N. Deepa, B. Prabadevi, P. N. Pathirana, J. Zhao, and W.-J. Hwang,
“Blockchain for edge of things: Applications, opportunities, and challenges,”
IEEE Internet of Things Journal, vol. 9, no. 2, pp. 964–988,
2022.

ISSN 2319-5991 www.ijerst.com
Vol. 16, Issue.1, Jan 2023

[19] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and
A.-R. Sadeghi, "D¨ıot: A federated self-learning anomaly detection system
for iot," in 2019 IEEE 39th International Conference on Distributed
Computing Systems (ICDCS). IEEE, 2019, pp. 756–767.

[20] S. Li, Y. Cheng, Y. Liu, W. Wang, and T. Chen, "Abnormal client behavior
detection in federated learning," arXiv preprint arXiv:1910.09933,
2019.

[21] R. Kumar, A. A. Khan, S. Zhang, W. Wang, Y. Abuidris, W. Amin, and
J. Kumar, "Blockchain-federated-learning and deep learning models for
covid-19 detection using ct imaging," arXiv preprint arXiv:2007.06537,
2020.

[22] H. Chai, S. Leng, Y. Chen, and K. Zhang, "A hierarchical blockchainenabled
federated learning algorithm for knowledge sharing in internet
of vehicles," IEEE Transactions on Intelligent Transportation Systems,
2020.

[23] M. Alazab, S. P. RM, P. M, P. K. R. Maddikunta, T. R. Gadekallu, and
Q.-V. Pham, "Federated learning for cybersecurity: Concepts, challenges,
and future directions," IEEE Transactions on Industrial Informatics,
vol. 18, no. 5, pp. 3501–3509, 2022.

[24] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson, and
B. Ag¨uera y Arcas, "Communication-efficient learning of deep networks
from decentralized data," ArXiv e-prints, pp. arXiv–1602, 2016.

[25] J. Koneˇcn`y, H. B. McMahan, F. X. Yu, P. Richt´arik, A. T. Suresh, and
D. Bacon, "Federated learning: Strategies for improving communication
efficiency," arXiv preprint arXiv:1610.05492, 2016.