Predicting Phishing in Blockchain using Machine Learning

Abstract

Security in Blockchain Environment has become a topic of concern with recent developments in the field. One of the most common attacks carried out by attackers is a Phishing Attack wherein the attacker tricks the Miner into adding a malicious block to the blockchain under genuine conditions to avoid detection and potentially destroy the entire Blockchain. The current attempts at detection includes Consensus Protocol, but it fails when a genuine miner is trying to add a new block to the blockchain. Zero-Trust Policies have been starting making rounds in the field to ensure complete detection of phishing attempts, but it is still in process of deployment and might take a significant time to deploy. A more accurate measure of phishing detection involves Machine Learning models using particular features to automate the entire process of classification of an attempt as a phishing or safe attempt. This paper highlights a few models which might be able to give safe results and might help in eradicating phishing attempts in Blockchain Environment.

Keywords- Blockchain, Machine Learning, Phishing, Cyberattacks, Ethereum

1. Introduction

Security in Blockchain has become a major concern with recent developments in the field. One of the most common and easiest attack carried out by the attacker is the Phishing Attack where the attacker tricks the miner into mining a malicious block into the Blockchain Environment using Social Engineering techniques. Many blockchain projects with high-grade security implications, were found to be susceptible to Phishing Attacks and many members of the community and its investors have lost a lot of money [1]. A report in 2017 showed that more than 50% of attacks carried out on Ethereum, were phishing attacks [2]. Due to Phishing Attacks being so common, it poses a threat to the three pillars of security: Privacy, Integrity and Availability.

A Phishing attack is normally done for a few aims, either the attacker has some monetary gains in the same or is getting it done to install a malicious code in the environment. It is majorly carried out by Social Engineering- where the miner is made to think the malicious block is a genuine block and would be benefitted by linking it in the blockchain, but is proven wrong, when the miner doesn't either gets the worth of his share in blockchain, or the blockchain becomes dysfunctional and no further linking is available.

There are many available methodologies to prevent this kind of attacks currently being used. The major methodology is Consensus Protocol, wherein a block is only linked to the blockchain if all the block-owners in the blockchain agree upon its addition. But it is a very time-consuming method and rigged with its own drawbacks and biases. [3] It led to Zero-Trust Policies in Blockchain Environment and research upon the same. It has been implemented in major Blockchains such as Bitcoin and Ethereum and have shown very promising results. [4] It is very complicated mechanism to deploy and still many new blockchains are trying to settle their foot in the same territory. Zero-Trust Policies are still being researched upon and might hold a promising future as well. But the attackers don't wait for its implementation to start their attacks. For the present, a newer and more efficient methodology is required to prevent Phishing Attempts. This can

be done using Machine Learning models and methodologies. This paper defines a few models which might be helpful for the same and have shown some promising results for the same as well. The researchers were able to reach 98.28% accuracy in Blockchain Phishing Attempt Detection using certain features in Blockchain Environment.

This paper is divided into 7 sections. The first section is the introduction to the project and the topic of interest. The second section covers the literature review and the third section defines the dataset. The fourth section defines the procedure of the experiment. The fifth section are observations of the experiment and the sixth section is the conclusion of the paper. Seventh section describes a few limitations of the research and possible future work.

This work is detailed research on Machine Learning Algorithms which can accurately predict Phishing Attempts in Blockchain environment. The researchers have developed models based on pre-known knowledge and with help of some experts as well. Hence, the paper dwells on the following Research Question:

RQ1 How accurately can Machine Learning predict a Phishing Attempt in a Blockchain Environment and what is the Model which brings out the best results?

2. Literature Review

Ye, Li, Cai, Gu and Fukuda [5] et al. researched and simulated a stable security system in blockchain which can withstand attacks such as 51%-attacks and other attacks. This research came into play after the WannaCry Ransomware outbreak and security in blockchain became a hot topic. Their simulation made an attacking state distinguishable from honest state. Their research also concluded that with higher power of attacks, the blockchain environment became more and more weaker and vulnerable to other attacks. Zhang, Xue and Liu [6] et al. in their paper wrote about Security and Privacy in Blockchain. Security practices in Blockchain have been termed as breakthrough in Cryptography and Cybersecurity by them. The notion of Smart Contracts and Smart Grids have improved security by implying Zero-Trust Policies and other security practices in all fields of Computer Science.

Aggarwal and Kumar [7] et al. did a brief analysis of attacks on Blockchain and came to a conclusion that the difficulty of attacking a blockchain is directly proportional to the number of miners mining in the blockchain. They analysed various kinds of attacks such as 51%-attacks, Sybil attacks, DoS attacks etc. The attacks are also dependent on the hash computing power of Computers on which the blockchain is hosted. Phillips and Wilder [8] et al. claimed Phishing attacks on Blockchain to be the most common kinds of attacks in blockchain environment. Many scammers try to redirect noble users to malicious websites which ask them for advanced fees and scams them into believing them to be mining for blockchain. They have also used Machine Learning clustering technique DBSCAN for detection of Phishing. Holub and O'Connor [9] et al. developed a phishing Detection system called COINHOARDER which traced and decided whether a Bitcoin

transaction was either a phishing attempt or safe. Their project was limited only to a Ukrainian based cryptocurrency. Fu, Yu and Feng [10] et al. also developed a Phishing Detection System using Machine Learning models, namely Convolution Networks, to predict an Ethereum transaction to be either fraud or safe. Their models received an 88.02% accuracy. Yuan, Yuan and Wu [11] also developed a transaction-based phishing detection model, which detected and classified models to be either safe or phish by mapping their transactions to subgraphs. They also worked on a Ethereum Blockchain and improvised a Graph2Vec and made stronger classifications. Zhang and Chen [12] used Multi-Channel Graph Classification in their Phishing Detection models and used Graph Neural Networks to map attempts as either phish or safe. They received 91% accuracy in the attempt. Bhowmik, Chandana and Rudra [13] et al. did a comparative study on various machine learning algorithms such as Naïve Bayes Classifier, Decision Trees etc. on Fraud Detection in Blockchain and found Decision Trees to be the most accurate models in Phishing Detection. Arshey and Viji [14] published their work in thwarting Security issues in Blockchain environment, with Machine Learning solutions. Their work was more theoretical but had many practical applications such as Covid-19 Data Storage and Cloud technology.

3. Dataset Used in Project

The dataset used in the project was obtained from Kaggle Open Source. [15] The dataset contains detailed data about transactions in Ethereum Network. It classifies the transactions as either Phishing Attempt or a safe Attempt. The features involved in the dataset include features such as Average Minutes between Sent and Received Transactions, Contracts created, Received and Sent Transactions over an Address, Minimum Value and Maximum Value of sent and received transactions over an address and Ethereum related details such as total Ethereum sent, unique Ethereum tokens, Ethereum contracts created etc.

It involved 829 empty data points, all missing in the same transaction, in a dataset with over 10,000 data points. These data points were removed during preprocessing.

The number of Phishing Attempts in the dataset were 1350 and safe attempts were 7662. Hence the dataset was imbalanced. Here is a representation of distribution of outputs over the dataset.

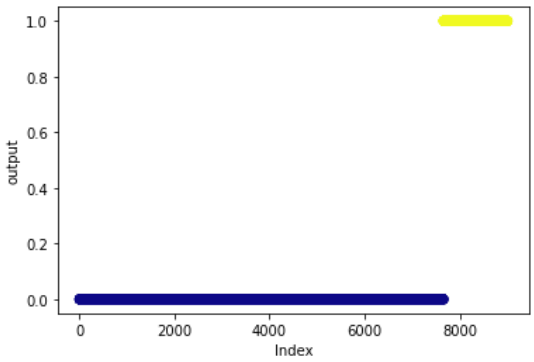

Fig 1. Distribution of Output with respect to Index in the database

Many of the features were either co-dependent on each other and many other features were either having same value or having small number of outliers and all other datapoints lying in the same range. All of such features were eliminated in the preprocessing stage. Some of such features are illustrated below:

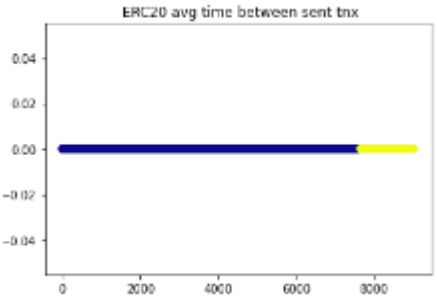

Fig 2. Illustrating all Datapoints having same value for Feature: ERC20 avg time between sent tnx

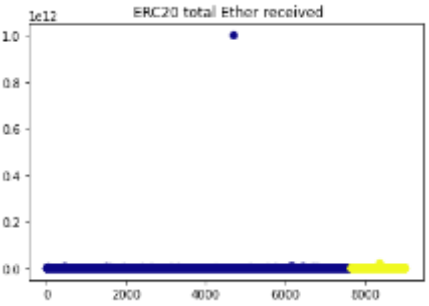

Fig 3. Illustrating a Single Outlier while all other datapoints having the same value for feature: ERC20 total Ether received

4. Procedure of Experiment

Before predicting the Phishing Attempts in Blockchain Environment, a machine learning project was done on a similar domain, phishing attempt prediction in websites. The observations made from the project were later on used in this project. The observations included:

1) Best accuracy in Random Forest and XGBoost Algorithms
2) Best accuracy achieved when total number of features used in model: 3N/5, where N is the total number of features in dataset.

Fig 4. Flowchart explaining the flow of Procedure

4.1 Data Preprocessing

Data Preprocessing step involved two phases: removal of null datapoints and removal of unwanted features. The size of dataset was reduced from 54 columns to 30 columns by removing columns having categorical unique data and features having same values for all datapoints or a few outliers. All such features were removed which contributed nothing to the actual model.

There were 829 null datapoints, belonging to a mixed category of outputs. They all had to be removed for smooth flow of model. They could not be filled by mean or binning methods, as it would have made them outliers and would have been ignored by the model.

4.2 Standardisation

Standardisation was done by using Standard Scaler on Training Dataset.

The formula of Standardisation is as follows:

$(V_c-M_c)/ SD_c$

Where V_c represents Value in column, M_c represents Mean of column and SD_c represents Standard Deviation of column. The standardisation used was Column Standardisation.

4.3 Train Test Split

This step involves splitting the dataset into training and testing datasets. Training dataset is the dataset which is fed to the model to train it based on input values. Testing dataset is the dataset which is used to check the accuracy of model in predicting correct output values.

We had not taken a cross validation dataset because we are not going to do iterations in the model. The dataset was divided into 80% training and 20% testing data

4.4 Models for classification

The project focused on four different models of classification namely Random Forest Classifier, XGBoost Classifier, Decision Tree using Gini Index and Decision Tree using Entropy. Random Forest provided with the best accuracy with 98.5% accuracy while XGBoost Classifier came a close second with 98.33% accuracy. The models were used based on their implementations in sklearn library in Python.

5. Models Used in Classification

5.1 Decision Tree Classifier using Gini Index

A decision tree classifier is a tree-based classifier which divides the data based on the feature which gives us best classification at a particular step. The best feature which classifies the dataset is chosen using a mathematical function. Gini index is one such mathematical function. The leaf nodes of this tree give us classified examples. The error is calculated based on number of wrong samples classified. Features with lower Gini Index are selected. [16]

The Gini Index is calculated as:

$$\text{Gini(D)}=1-\sum_{i=1}^{m} \text{Pi}^2$$

Fig 5. Gini Index Formula

Where D is the tuple in consideration, P_i is the probability of tuple belonging to class i and m is the total number of classes

After pre-processing, our Decision Tree using Gini Index gave us a 97.33% accuracy which is pretty high, but we received better results with other options.

5.2 Decision Tree Classifier using Entropy

Entropy is another mathematical function that is used in feature selection for classification in Decision Trees. Entropy is a more complex mathematical notation than Gini Index and it gives us an Information Gain. Over here the features with higher Information Gain are selected. It makes use of logarithmic function to calculate the predictability of a certain feature. [16]

The Entropy is calculated as:

$$\text{Info(D)}= -\sum_{i=1}^{m} pi \log_2 pi$$

Fig 6. Information Gain formula

Where D is tuple in consideration, pi is the probability of tuple

belonging to class i and m is total number of classes

After pre-processing, our Decision Tree using Entropy gave us a 97.22% accuracy, just a little bit lesser than Decision Tree using Gini Index. It classified one sample as a false negative.

5.3 Random Forest Classifier

Random Forest Classifier is a more complex version of Decision Tree Classifiers. It is based on Bagging Method; wherein random subsets of features and examples are selected in a predefined format and they are made into small decision trees. It accurately measures the importance each feature gives to the decision trees and creates a final decision tree using all the subset decision trees. The final decision tree has better accuracy than the subset decision trees. [17]

Random Forest Classifier in our case gave us the best accuracy of 98.5% and can be implemented in real life scenarios as well

5.4 XGBoost Classification

XGBoost Classification uses the Boosting Method as well along the Bagging Method used in Random Forest Classification. The Boosting Method gives weights to each and every subset decision tree branches and adjusts them accordingly to reduce the error faced. It first develops a decision tree using a subset, it then boosts the decision tree by reducing its error and changing features and data variables in the subset decision tree. It does so in a predefined manner and till we reach an error rate lesser than defined threshold. Using XGBoost algorithm might increase the bias in the model, but reduces the variance rate. [18]

Our XGBoost Classifier provided with almost same accuracy as Decision Tree classifier with a 98.33%, but XGBoost can be implemented in real time with more efficiency because it gave us smaller False Negative rate.

5.5 Artificial Neural Network

An Artificial Neural Network (ANN) is a model copied from human neurons and acts accordingly. It is a mixture of interconnected nodes and layers with first layer being the input layer and the last being the output layer. There are hidden layers in between and each layer has a set of nodes in them. [19] The input layer set of nodes is the same number as input features while output layer set of nodes is same number as output variables. The number of hidden layers and nodes in them are to be decided by the model creator to avoid underfitting and overfitting. The activation function of a layer, takes input and converts it to output as an input for the next layer. [20] Each branch or connection between nodes have a weight which plays an important role in deciding the output.

We used an ANN with 28 input nodes, two hidden layers, the first having 11 nodes and second having 6 and one output node. Each layer used the sigmoid as its activation function. Of three different configurations tried by

us, this configuration gave the best output with 88.8% accuracy.

6. Observations

It was observed that Random Forest Classifier presented with best accuracy at 30 features from all the models tested on the dataset. It provided us with 98.5% accuracy and XGBoost Classifier provided with 98.33% accuracy. These two models are the most accurate Machine Learning implementations and can be used further on for real-time applications in Ethereum networks as well. The following figures show the Confusion Matrix for the two results:

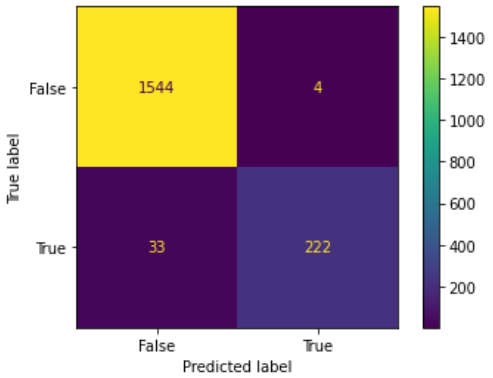

Fig 7. Confuson Matrix for Random Forest Classifier

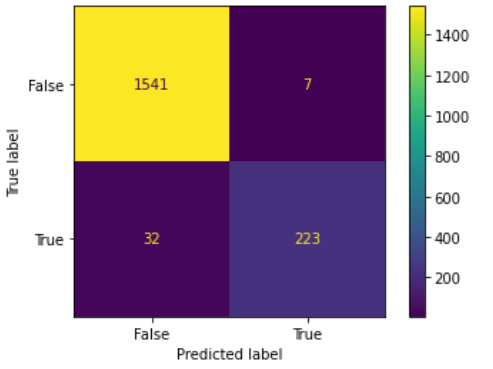

Fig 8. Confusion Matrix for XGBoost classifier

The major worry in the output refers to the notion of False Negatives. False Negatives refer to the values which were a phishing attempt but were classified as safe attempts. This can be dangerous as malicious attackers can use this kind of feature manipulation to their advantage and still be able to successfully carry out a phishing attempt. The case of False Negatives is slightly higher in Random Forest classifier than XGBoost classifier, but XGBoost classifier has higher cases of False Positives, which is still a neglectable outcome.

Observation Table for different classification models

Model	Accuracy
Random Forest Classifier	98.5%
XGBoost Classifier	98.33%
Decision Tree using Gini Index	97.33%
Decision Tree using Entropy	97.22%
Artificial Neural Network	88.8%

With these observations, we can successfully answer the Research Question:

How accurately can Machine Learning predict the Phishing Attempts in Blockchain Environment and which Model is best suitable for the same?

We can say, Random Forest Classification is best suitable for prediction of Phishing Attempts in Blockchain Environment due to it going thoroughly through all possible feature combinations and selecting best feature output in its final Decision Tree. XGBoost Classifier should although be preferred in real-time applications because of it having slightly lower probability of classifying Phishing Attempts as safe.

7. Conclusion

It can be concluded from the given observations that Machine Learning will be a more efficient method to detect Phishing Attempts in Blockchain environment than Consensus Method. It has its own limitations and might not be able to give perfect results as of now, but

with limited number of features and few datapoints, Random Forest Classifier will successfully be able to classify any attempt as either Phishing or Safe.

Some study is still required to ensure no cases of False Negatives occur and all phishing attempts are classified correctly. XGBoost Classifier gives slightly higher accuracy in Random Forest in the field and can be chosen for real-time applications.

The reason why Decision Trees failed infront of Random Forest is because, Decision Trees do not go thoroughly through the features provided to them and they use a Greedy Approach. Decision Trees choose the next best fit for the model and they do not create subsets of features to choose the best fit, hence they failed when in comparison to Random Forest and XGBoost Classifier.

8. Limitations and Future Work

The research does lag behind some important aspects such as cross validation and thorough comparison.

Here are some possibilities of future work:

1) Using a cross validation dataset along with a training and testing dataset, which might improve the Testing accuracy of the models
2) Hyperparameter Tuning using Cross Validation dataset for choosing the best hyperparamters and higher accuracies
3) Using newer and more advanced Machine Learning algorithms and trying out unsupervised deep learning models on the dataset such as Graph Neural Networks in accordance to improve accuracy
4) More configurations of hidden layers and nodes and activation functions can be experimented with to achieve higher accuracy

9. References

[1] Andryukhin, A. A. (2019, March). Phishing attacks and preventions in blockchain based projects. In *2019 International Conference on Engineering Technologies and Computer Science (EnT)* (pp. 15-19). IEEE.

[2] Wen, H., Fang, J., Wu, J., & Zheng, Z. (2021, May). Transaction-based hidden strategies against general phishing detection framework on ethereum. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1-5). IEEE.

[3] Oyinloye, D. P., Teh, J. S., Jamil, N., & Alawida, M. (2021). Blockchain consensus: An overview of alternative protocols. *Symmetry*, *13*(8), 1363.

[4] Alevizos, L., Ta, V. T., & Hashem Eiza, M. (2022). Augmenting zero trust architecture to endpoints using blockchain: A state-of-the-art review. *Security and Privacy*, *5*(1), e191.

[5] Ye, C., Li, G., Cai, H., Gu, Y., & Fukuda, A. (2018, September). Analysis of security in blockchain: Case study in 51%-attack detecting. In *2018 5th International conference on dependable systems and their applications (DSA)* (pp. 15-24). IEEE.

[6] Zhang, R., Xue, R., & Liu, L. (2019). Security and privacy on blockchain. *ACM Computing Surveys (CSUR)*, *52*(3), 1-34.

[7] Aggarwal, S., & Kumar, N. (2021). Attacks on blockchain. In *Advances in Computers* (Vol. 121, pp. 399-410). Elsevier.

[8] Phillips, R., & Wilder, H. (2020, May). Tracing cryptocurrency scams: Clustering replicated advance-fee and phishing websites. In *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)* (pp. 1-8). IEEE.

[9] Holub, A., & O'Connor, J. (2018, May). COINHOARDER: Tracking a ukrainian bitcoin phishing ring DNS style. In *2018 APWG Symposium on Electronic Crime Research (eCrime)* (pp. 1-5). IEEE.

[10] Fu, B., Yu, X., & Feng, T. (2022). CT-GCN: a phishing identification model for blockchain cryptocurrency transactions. *International Journal of Information Security*, 1-10.

[11] Yuan, Z., Yuan, Q., & Wu, J. (2020, August). Phishing detection on Ethereum via learning representation of transaction subgraphs. In *International Conference on Blockchain and Trustworthy Systems* (pp. 178-191). Springer, Singapore.

[12] Zhang, D., Chen, J., & Lu, X. (2021, August). Blockchain Phishing scam detection via multi-channel graph classification. In *International Conference on Blockchain and Trustworthy Systems* (pp. 241-256). Springer, Singapore.

[13] Bhowmik, M., Chandana, T. S. S., & Rudra, B. (2021, April). Comparative Study of Machine Learning Algorithms for Fraud Detection in Blockchain. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 539-541). IEEE.

[14] Arshey, M., & Viji, K. A. (2021, March). Thwarting cyber crime and phishing attacks with machine learning: a study. In *2021 7th international conference on advanced computing and communication systems (ICACCS)* (Vol. 1, pp. 353-357). IEEE.

[15] https://www.kaggle.com/datasets/xblock/ethereum-phishing-transaction-network

[16] Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, *39*(4), 261-283.

[17] Devetyarov, D., & Nouretdinov, I. (2010, October). Prediction with confidence based on a random forest classifier. In *IFIP international conference on artificial intelligence applications and innovations* (pp. 37-44). Springer, Berlin, Heidelberg.

[18] Chen, Z., Jiang, F., Cheng, Y., Gu, X., Liu, W., & Peng, J. (2018, January). XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud. In *2018 IEEE international conference on big data and smart computing (bigcomp)* (pp. 251-256). IEEE.

[19] Krogh, A. (2008). What are artificial neural networks?. *Nature biotechnology*, *26*(2), 195-197.

[20] Stathakis, D. (2009). How many hidden layers and nodes?. *International Journal of Remote Sensing*, *30*(8), 2133-2147.