Phishing Attack Detection in Ethereum Transactions with PCA-Enhanced Machine Learning

Khuushi Maheshwari[1], Srujan Kumar Ch.[1], Y.V. Srinivasa Murthy[1], and Anand Paul[2]
[1]Dept. of IT, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, Karnataka - 575 104, India.
[2]Biostatistics and Data Science, Health Sciences Center, Louisiana State University, LA 70112, USA.
Email IDs: *khuushi.mitblr2023@learner.manipal.edu, chilveri.mitblr2022@learner.manipal.edu, vishnu.murthy@manipal.edu,* and *apaul4@lsuhsc.edu*

***Abstract*—Phishing attacks in Ethereum transactions pose a significant threat to the security and integrity of blockchain-based systems, as these scams exploit user vulnerabilities to extract sensitive information or cryptocurrency assets. In contrast to the approaches proposed by the various research works to tackle phishing detection, many struggle with high-dimensional datasets, leading to computational inefficiencies and overfitting. To address these gaps, this study applies principal component analysis (PCA) for dimensionality reduction, helping in the development of more efficient and robust machine learning models by reducing data complexity and enhancing model generalization. A comparative analysis is conducted using multiple algorithms, including support vector machines (SVMs), decision trees (DT), XGBoost, and multi-layer perceptron (MLP). By evaluating their performance using standard metrics such as accuracy, F1 score, precision, recall, and ROC-AUC, the MLP model demonstrates superior accuracy and generalization, establishing its efficacy for phishing detection in Ethereum transactions. This work highlights the importance of feature reduction techniques and neural network models in enhancing the accuracy and efficiency of phishing detection systems, paving the way for future advancements in blockchain security.**

***Index Terms*—Phishing Detection, Ethereum, Principal Component Analysis (PCA), Machine Learning, Blockchain Security.**

I. Introduction

Ethereum works on the principles of blockchain [1]. In recent years, blockchain technology has experienced significant growth. Global spending on blockchain solutions is projected to reach €3 trillion [2]. The decentralized, unchangeable digital ledger known as blockchain technology has completely transformed a number of sectors, most notably banking. Through cryptographic hashing, it enables the possible safety for data storage and guarantees transparency while guarding against unwanted manipulation. By adding smart contracts, Ethereum, which is one of the most well-known blockchain platforms expands these features and enables programmers to create decentralized apps (DeFi) [3]. Ethereum is a prominent platform for blockchain-based innovations because its own cryptocurrency, Ether, facilitates these functions [4].

Ethereum is extremely vulnerable to cyberthreats, especially phishing assaults, despite its decentralized and transparent nature. The openness property of blockchain makes transaction records available to the general public, which promotes transparency. However, gives bad actors a way in. Phishing attacks employ this visibility to target consumers by impersonating reliable organizations and deceiving them into disclosing private information [5]. Phishing is a type of social engineering assault in which victims are tricked into divulging secret keys or credentials. This technique is frequently used to steal money or private information. These schemes have serious financial consequences and frequently cause large losses for both people and businesses.

For instance, in 2022, the Federal Bureau of Investigation (FBI) revealed that over 300 thousand people had fallen victim to phishing, and that they had lost over $52.1 million [6]. Over 500 million phishing attacks were reported. The 2021 Microsoft 365 phishing effort is a prominent example, in which hackers pretended to be Microsoft and fooled users into clicking on malicious uniform resource locators (URLs) in order to gain login credentials [7]. Phishing scams frequently use phony addresses or transactions in the Ethereum ecosystem, taking advantage of users faith in decentralized systems.

In this paper, we have performed a comparative study and aimed to gather all the various data cited in many papers, testing them with some of the most famous algorithms and getting their benchmarks. We mainly focus on problems such as phishing transaction detection and fraudulent transaction detections. Also, a comparative study on phishing transaction detection in Ethereum, focusing on analyzing various machine learning algorithms and finding the effectiveness in identifying fraudulent activities within blockchain transactions. A phishing transaction dataset has been used to evaluate the performance of several popular models that include support vector machines (SVMs), multi-layer perceptron (MLP), gradient boost (XGBoost), Decision Trees (DTs), and Random Forest (RF). This approach involves the application of principal component analysis (PCA) for dimensionality reduction, aiming to improve the models accuracy by reducing overfitting and enhancing computational efficiency. The focus is to identify the most effective algorithm for phishing transaction detection and evaluate its suitability for real-world applications in blockchain networks like Ethereum.

The rest of the paper is organized as follows: Section II covers the various works done in the phishing attack detection. The proposed methodology is detailed in section III. Result analysis with detailed observations made with and without dimensionality reduction techniques are given in section IV. Section V concludes the work with some possible open

problems.

II. Literature Review

In the past few years, the rapid rise in popularity and economic significance of blockchain environments, particularly Ethereum, has led to a heightened focus on the development of effective phishing detection techniques [REF]. Numerous methods, each with unique advantages and disadvantages, have been devised to detect phishing accounts. These initiatives can be broadly divided into two groups: (i) network-based and (ii) feature-based detection. To detect phishing activity, feature-based approaches examine particular aspects of accounts and transactions. Metrics like the quantity of transactions, account balance, average transfer amount, etc are frequently included in these attributes [REF]. Models can spot trends that set phishing accounts apart from authentic ones by looking at these characteristics.

In order to detect phishing schemes on Ethereum, the study by Bartoletti *et al.* developed a novel hybrid deep neural network model called LBPS [8]. The model combines transaction record analysis and manual feature engineering for feature extraction. LSTM-FCN was used to record temporal information, while back propagation neural networks (BPNN) have been used to uncover implicit correlations between features. According to experimental results, the LBPS model outperformed baseline and conventional approaches, achieving a high F1-score of 0.97. However, the model has high computing demands due to its hybrid nature, and depends significantly on labeled datasets, which are scarce in Ethereum. Furthermore, bias may be introduced by the manual feature engineering process, which compromises generalization skills [9].

It is essential to estimate the temporal information to enhance the identification process of phishing scams. Temporal Graph Attention Networks (TGAT) have been used in phishing detection using the temporal graph attention (PDTGA) model to enhance the identification of phishing scams in Ethereum transactions [10]. PDTGA models the Ethereum transaction graph's temporal evolution as a function of continuous time, in contrast to traditional approaches. Through a self-attentive method, the model integrates temporal signals, node properties, and edge data to improve the identification of dynamic phishing behaviours. The study showed that PDTGA outperformed current graph-based techniques, achieving an AUC score of 94.78% and a recall score of 0.89 on Ethereum phishing datasets.Graph neural networks have high processing costs and face challenges in real-time deployment due to the need for intricate temporal modeling. [11].

A Ponzi scheme is one where returns for earlier investors are paid using the capital from newer investors, rather than from legitimate profits. Focusing on such Ponzi schemes as a crucial issue, fraudulent patterns within them have been primarily detected. A dataset of 184 smart contracts implementing Ponzi schemes was constructed, and manual feature analysis was performed to identify common fraudulent patterns. The study highlighted vulnerabilities in Ethereum's openness, which fraudsters exploit to create *trustworthy* scams. Although this work provides foundational knowledge on blockchain scams, these methods rely heavily on manual inspection, which limits scalability. The absence of advanced machine learning techniques for automation restricts its applicability to dynamic and large-scale phishing detection [12].

Given the high complexity of machine learning algorithms, it is essential to build light weight models for embedding applications in devices with low computational power. Hence, the use of light gradient boost machine (LightGBM) in conjunction with a graph-based cascade feature extraction technique is considered to detect phishing scams [13]. The method uses a two-order transaction graph to extract account features and dual-sampling approaches to alleviate class imbalance. Although the approach performed well in categorization, it is limited to analyzing static graphs and does not account for the temporal dynamics of transaction patterns. Due to this restriction, it is less effective in capturing the changing patterns of phishing accounts, which is crucial for Ethereum real-world applications. To enhance the scam detection in phishing attacks and make it available in real-time applications, a technique that analyzes smaller subgraphs as opposed to the full transaction network has been proposed [14]. Particularly for large and dynamic datasets, this method speeds up and increases the scalability of the process. TGC employs two contrastive learning modules to differentiate phishing nodes from their typical neighbors and identify patterns in sparsely distributed phishing nodes, setting it apart from competing methods. Although the results demonstrate notable advancements over current techniques, the proposed framework efficacy is highly dependent on the caliber of training data, and it may struggle to adjust to changing phishing techniques or real-time situations [15].

With the recent advancements in deep learning, it is found that graph neural network perform well in analyzing the data connected in the form of nodes. A graph neural network architecture called Ethident is introduced [16]. Ethident records intricate behavior patterns at the node and subgraph levels using a hierarchical graph attention encoder (HGATE). Additionally, it makes use of self-supervised learning that addresses the problem of sparse labeled data. It is not designed primarily for detecting phishing addresses, which frequently utilize more dishonest and sparsely connected patterns but it does a good job of recognizing different account behaviors and types. Furthermore, in circumstances when labeled datasets are limited, it's performance may get degraded as it is dependent on labeled datasets.

Collectively, these studies demonstrate promising advancements in addressing blockchain security challenges through machine learning and graph-based methods. While these methods show improvements, they also highlight areas requiring further development."Incorporating Principal Component Analysis (PCA) and other dimensionality reduction techniques can enhance the efficiency and adaptability of these approaches in dynamic environments. Additionally, lightweight machine learning models have not been widely explored for phishing attack detection in the existing literature. Dimensionally reduced

features that are fed to prominent machine learning models that include ensemble methods may enhance performance. Therefore, we experimented with various machine learning models using PCA-applied feature vectors and observed interesting results.

III. Proposed Methodology

This work focuses on the comparative analysis of various machine learning models for the task of phishing detection. The flow diagram shown in in in Fig.1 summarizes the methodology.

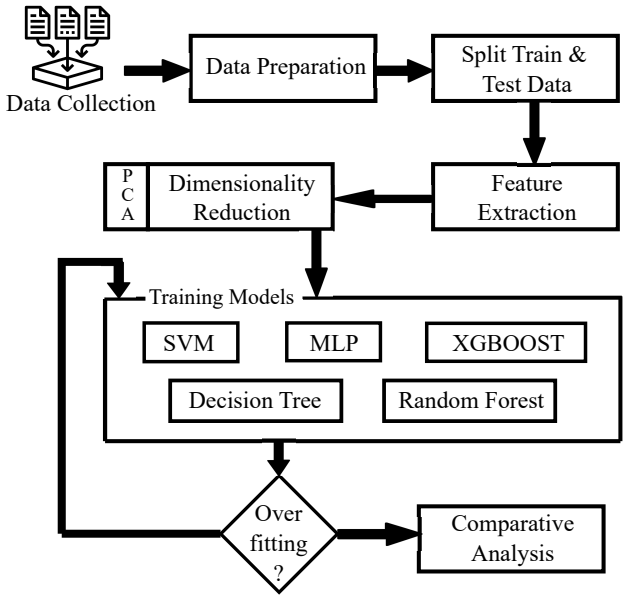

Fig. 1. Proposed flow diagram that compares various models for phishing detection.

The key steps include:

- Preprocessing dataset to handle missing values and normalize features.
- Applying PCA to reduce the dataset's dimensionality.
- Training and evaluating various models, including SVM, MLP, xgBoost, decision trees, and Random Forest.
- Selecting the model with the best performance based on accuracy and generalization metrics.

The details on each block have been given in the following subsections.

A. Data Collection

The dataset for this work comes from Etherscan, a blockchain explorer for Ethereum transactions. It was originally created using publicly available phishing reports on Etherscan, where certain Ethereum addresses were flagged as phishing accounts. To expand on this, a second-order bredth-first search (BFS) crawler has been used to collect a much larger Ethereum transaction network, giving a broader picture of how phishing and non-phishing accounts interact. The original dataset had 2,973,489 nodes (Ethereum addresses) and 13,551,303 edges (transactions). Each node has an inernet servise provider (ISP) attribute that marks if it is a phishing account. Also, each transaction includes details like amount and timestamp, helping capture fund movements across the network. This dataset was processed and reduced to 50 key features, keeping the most important transaction and behavioural patterns. The refined dataset has been included in the GitHub repository: https://github.com/kofuuku/Ethereum-Phishing-Detection, which had originally sourced it from Kaggle[1].

B. Dimensionality Reduction: PCA

In order to decrease the dimensionality of the data without losing essential information, PCA has been applied. PCA transforms the initial feature space to a new space of principal components(PCs) that explain the most variance in the data. Variance is used because it quantifies the spread of data. guaranteeing the maintenance of the most informative patterns while eliminating non-essential components and distractions. Searching for routes with the highest variance, PCA captures the pattern of the data in a lower-dimensional setting. A lower-dimensional space enhances model efficiency by reducing computational complexity, speeding up training, and mitigate the curse of dimensionality, which can negatively impact performance. To determine the optimal feature vector, the variation in the number of ingredients was from 1 to 46 depending on the ratio of explained variance. This methodology enhanced computational efficiency with little information loss. Because high-dimensional data is more prone to overfitting, PCA improves generalization by eliminating correlated and redundant features. The effect of dimensionality reduction on model performance has been assessed by training several classifiers on several PCA-transformed datasets.

C. Machine Learning Models

In this work, machine learning models such as SVM, decision tree, random forest, xgboost and multi layer perceptron (MLP) have been considered. The details of each model is given below:

1) Support Vector Machines: The support vector machine model is a supervised machine learning model often used for classification and regression tasks and has been used in this paper to classify transactions as phishing or non phishing. It does so by finding the optimal decision boundary to seperate the two classes. SVM constructs a hyperplane in a high-dimensional space, which maximizes the margin. The margin is the distance between the hyperplane and the nearest data points from each class, known as support vectors. A larger margin enhances generalization by ensuring effective classification of unseen data while also minimizing the risk of overfitting.

Though SVM is particularly effective for datasets with high-dimensional feature spaces as the model performs well when the classes are well-separated , in real-world phishing detection, data distribution is usually highly imbalanced, with overlapping classes and non-linear patterns. To overcome this,

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

three different kernels: (i) Linear, (ii) Radial Basis Function (RBF), and (iii) Sigmoid were used with to determine the most suitable approach. The linear kernel efficiently handles linearly separable data while maintaining interpretability. The RBF kernel captures complex, non-linear relationships by mapping the data into a higher-dimensional space. The sigmoid kernel is similar to the activation functions of neural networks and was tested to model intricate decision boundaries.

2) Multi Layer Perceptron: A Multi-Layer Perceptron (MLP) was utilized to simulate complicated, non-linear phishing relations among phishing transactions. MLP consists of an input layer, one or multiple hidden layers, and an output layer, where every neuron applies an activation function to introduce non-linearity. Learning is done using backpropagation where errors are back-propagated to adjust weights and minimize the loss function. The Adam optimizer, which contains momentum and learning rate adaptation, was used to facilitate stable training and fast convergence. MLP turned out as the best for phishing detection as it effectively builds intricate structures in transactional data sets. The model was trained on PCA-transformed data and tested on different PCA dimensions to evaluate its efficacy in differentiating between phishing and non-phishing transactions.

3) Decision Tree: Decision Trees (DT) were used to classify phishing transactions recursively by dividing the feature value-based dataset. A decision tree is composed of nodes for feature tests, branches for decision output, where the leaf nodes are the class labels. The model selects the best feature for splitting iteratively, seeking to acquire maximum information and create a hierarchical decision structure. In a comparison of the impacts of different splitting criteria, two Various models were attempted, one being based on Gini impurity measure (DCT Gini) and another using entropy (DCT Entropy). Gini Impurity calculates the probability of misclassification, preferring divisions into homogeneous subsets, while entropy measurements recognizes the irregularities in the dataset and guides the divisions in order to minimize uncertainty.

4) Random Forest: Random Forest works like a collection of decision trees together leading to a conclusion. It constructs a complete "forest" of decision trees, trained on a precise subset of the data collection, as opposed to using a single decision tree. The winner is determined through majority vote after these trees work on their own and produce their own predictions. From this, Random Forest ascertains the ability to handle a wide range of data patterns and lowers the possibility that the model would overfit—that is, learn the noise in the data rather than the patterns themselves. The Random Forest technique is a powerful option for phishing detection as it is able to handle the complex interdependencies between the different properties of Ethereum transactions. The final selection is made based on the dominant characteristics of the trees. to determine, and the model produces predictions using the data obtained from several trees, each of which focuses across various parts of the data set. It can be optimized. for maximum performance by fine-tuning parameters like the the number of trees or depth of each tree.

5) XGBOOST: XGBoost is a potent algorithm designed for speed and effectiveness. It functions by successively building trees, each of which attempts to correct the errors of the one before it. Because of this, XGBoost is incredibly good at learning from mistakes and gradually enhancing its forecasts. Additionally, it works well with data that is unbalanced, where one class—such as phishing transactions—may be underestimated. PCA was used to reduce the dataset's dimensions before XGBoost was trained on it for this challenge. To ensure that the model didn't overfit and could effectively generalize to new data, we experimented with a number of parameters, such as the learning rate and the tree depth. We evaluated XGBoost's performance against the other models using criteria such as recall, accuracy, precision, recall, and F1 score to see which one performed best.

IV. Results and Observations

This section presents the results obtained from the various classification models evaluated on the collected dataset across the two stages of experimentation. Stage 1 focuses on classification without applying any dimensionality reduction techniques, while Stage 2 incorporates Principal Component Analysis (PCA) to reduce the dataset's dimensionality. Although multiple dimensionality reduction techniques exist, PCA was chosen for this study as it is computationally economical. The performance of models in both stages is summarized in Tables I and II. The experiments were conducted on a publicly available Ethereum transaction dataset, with performance evaluated using accuracy, precision, recall, F1-score, and ROC-AUC score.

Accuracy: Measures the proportion of correctly classified transactions.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

Precision: Indicates the proportion of correctly identified phishing transactions among all transactions classified as phishing.

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

Recall (Sensitivity): Represents the proportion of actual phishing transactions correctly identified by the model.

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

F1-Score: The harmonic mean of precision and recall, which gives a balanced measure for imbalanced datasets.

$$F1\text{-}Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

ROC-AUC Score: Evaluates the model's ability to distinguish between phishing and non-phishing transactions.

The True Positive Rate (TPR) and False Positive Rate (FPR) are defined as:

$$TPR = \frac{TP}{TP+FN} \tag{5}$$

TABLE I

THE PERFORMANCE OF VARIOUS CLASSIFICATION MODELS THAT ARE FED WITH COMPLETE FEATURES. *Note:* CELLS HIGHLIGHTED IN BOLD AND GREY INDICATE BETTER PERFORMANCE.

Model	Accuracy	F1 Score	Precision	Recall	ROC AUC
SVM Linear	0.8663	0.0398	0.5556	0.0207	0.8329
SVM RBF	0.8702	0.0714	0.9000	0.0372	0.9462
SVM Sigmoid	0.8125	0.0000	0.0000	0.0000	0.5671
MLP	0.9196	0.7627	0.6314	0.9628	0.9759
XGBoost	0.9506	0.8355	0.7559	0.9339	0.9835
Decision Tree (Gini)	0.7277	0.2243	0.1816	0.2934	0.5442
Decision Tree (Entropy)	0.9107	0.6005	0.7516	0.5000	0.7372
Random Forest	**0.9024**	0.4395	0.9583	0.2851	0.9463

TABLE II

THE PERFORMANCE OF VARIOUS LIGHT-WEIGHT MACHINE LEARNING MODELS WITH PCA. *Note:* CELLS HIGHLIGHTED IN BOLD AND GREY INDICATE BETTER PERFORMANCE.

Model	#Dimensions	Accuracy	F1 Score	Precision	Recall	ROC AUC
SVM Linear	29	0.8674	0.0553	0.6364	0.0289	0.6938
SVM RBF	25	0.8708	0.0791	0.9091	0.0413	0.9348
SVM Sigmoid	2	0.8641	0.0000	0.0000	0.0000	0.5472
MLP	43	0.9678	0.8849	0.8511	0.9215	0.9813
XGBoost	30	0.9556	0.8305	0.8522	0.8099	0.9701
Random Forest (Gini)	15	0.9224	0.7071	0.7161	0.6983	0.8277
Random Forest (Entropy)	27	0.9290	0.7344	0.7375	0.7314	0.8455
Decision Trees	30	0.8996	0.4290	0.9067	0.2810	0.9539

$$FPR = \frac{FP}{FP+TN} \tag{6}$$

where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives from the confusion matrix.

It is observed that the classifiers based on SVM, such as SVM linear, radial basis function (RBF), and sigmoid functions, could not perform well in classifying phishing attacks. One can specify that SVMs are always giving poor performance for phishing detection in Ethereum transactions based on the rows in Tables I and II. The performance of SVM classifiers remains consistently lower across all metrics when compared to other models.This is due to their inability to effectively handle imbalanced datasets and non-linear feature interactions. Since SVMs rely on maximizing the margin between classes, they struggle when phishing transactions (minority class) are vastly outnumbered by non-phishing transactions, leading to poor generalization. This is evident in their recall values indicating a failure to correctly identify phishing transactions. Even though SVM models achieve relatively higher precision (0.9091 for SVM RBF with dimensionality reduction), this comes at the cost of missing a large number of phishing transactions. Since phishing transactions are in the minority the model will label most of the transactions as non-phishing (majority class) to reduce errors.

Interestingly, although tree-based models such as Random Forest demonstrate good performance overall; yet, they have poor recall, which reveals that they are also poor at minority-class detection. However, unlike SVM, Random Forest compensates with a high ROC AUC score indicating improved overall separability between phishing and non-phishing transactions. The tree-based models outperform support vector machines since they are able to determine non-linear relationships with recursive feature splitting. Decision Trees build a hierarchical framework by choosing most informative properties, while Random Forest generalizes by aggregating several trees learned on random subsets. XGBoost improves performance through gradient boosting that optimizes trees iteratively to reduce errors.These models attain better accuracy and ROC AUC score since they can learn more intricate decision boundaries. Their greater accuracy yields by decreasing the share of misclassifications per feature ranking of importance and ensemble learning, which makes them more reliable for phishing detection. Multi-Layer Perceptron (MLP) is characterized by its highest recall, which indicates its capacity to recognize phishing transactions while keeping false negatives low. This is because MLP can learn complicated, non-linear patterns with multiple hidden layers and activation functions. Apart from possessing good recall, MLP also excels in accuracy and ROC AUC score, thanks to backpropagation and adaptive optimization (Adam), which assist in refining its weight updates effectively. Its high accuracy demonstrates its capability to discriminate between phishing and non-phishing. transactions well. MLP's performance, however, relies heavily on hyperparameter tuning such as learning rate and layer appropriate sizes, and it needs adequate training data to avoid overfitting or underfitting. Further tuning of such models, such as optimization network topology, activation functions, and training regimen strategies may well increase accuracy and efficiency. Later work can examine these enhancements to enhance phishing detection in Ethereum transactions.

V. Conclusions

In this work, we explored the application of Principal Component Analysis (PCA) for feature selection on a phishing transaction detection dataset. After evaluating several machines learning models, including SVM, XGBoost, Decision Trees, Random Forest, and MLP. The results demonstrated a clear improvement in the models' ability to classify phishing transactions effectively. Among all the models tested, MLP (Multilayer Perceptron) exhibited the best balance of accuracy, precision, recall, and ROC AUC, validating the importance of dimensionality reduction for improved model performance. Given the rise in cryptocurrency-based scams, phishing detection in Ethereum transactions has become crucial to preventing fraudulent activities and protecting users' assets. Efficient detection can help mitigate financial losses, secure user identities, and maintain the integrity of blockchain-based systems. This paper underscores the value of applying machine learning techniques to enhance security in blockchain ecosystems and digital asset transactions. While PCA has shown its effectiveness in improving model performance, future work could explore other feature reduction techniques which might uncover even deeper patterns in the data. Additionally, experimenting with more advanced deep learning models, like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), could further boost the accuracy of phishing detection in Ethereum transactions. Since MLP has provided promising results, trying more complex neural network structures could improve performance, especially when dealing with large amounts of transaction data. With the rapid growth of cryptocurrency transactions and the increasing threat of phishing, continuing to refine these techniques could lead to even more effective detection systems, improving security for users and the entire Ethereum network.

References

[1] Mayukh Mukhopadhyay. *Ethereum Smart Contract Development: Build blockchain-based decentralized applications using solidity*. Packt Publishing Ltd, 2018.

[2] Sunhilde Cuc. Unlocking the potential of blockchain technology in the textile and fashion industry. *FinTech*, 2(2):311–326, 2023.

[3] Johannes Rude Jensen, Victor von Wachter, and Omri Ross. An introduction to decentralized finance (defi). *Complex Systems Informatics and Modeling Quarterly*, (26):46–54, 2021.

[4] Alaa Hamid Mohammed, Alaa Amjed Abdulateef, and Ihsan Amjad Abdulateef. Hyperledger, ethereum and blockchain technology: a short overview. In *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–6. IEEE, 2021.

[5] Umer Majeed, Latif U Khan, Ibrar Yaqoob, SM Ahsan Kazmi, Khaled Salah, and Choong Seon Hong. Blockchain for iot-based smart cities: Recent advances, requirements, and future challenges. *Journal of Network and Computer Applications*, 181:103007, 2021.

[6] Trang Thi Thu Horn. *A Study of Cybersecurity Landscape in the United States: Trend, Regional Variations, and Socioeconomic Factors*. PhD thesis, Indiana State University, 2024.

[7] Marshall S Rich. Enhancing microsoft 365 security: Integrating digital forensics analysis to detect and mitigate adversarial behavior patterns. *Forensic Sciences*, 3(3):394–425, 2023.

[8] Massimo Bartoletti, Salvatore Carta, Tiziana Cimoli, and Roberto Saia. Dissecting ponzi schemes on ethereum: Identification, analysis, and impact. *Future Generation Computer Systems*, 102:259–277, 2020.

[9] Atta Ur Rahman, Feras Al-Obeidat, Abdallah Tubaishat, Babar Shah, Sajid Anwar, and Zahid Halim. Discovering the correlation between phishing susceptibility causing data biases and big five personality traits using c-gan. *IEEE Transactions on Computational Social Systems*, 2022.

[10] Lei Wang, Ming Xu, and Hao Cheng. Phishing scams detection via temporal graph attention network in ethereum. *Information Processing & Management*, 60(4):103412, 2023.

[11] Panpan Li, Yunyi Xie, Xinyao Xu, Jiajun Zhou, and Qi Xuan. Phishing fraud detection on ethereum using graph neural network, 2022.

[12] Massimo Bartoletti, Stefano Lande, Andrea Loddo, Livio Pompianu, and Sergio Serusi. Cryptocurrency scams: analysis and perspectives. *Ieee Access*, 9:148353–148373, 2021.

[13] Weili Chen, Xiongfeng Guo, Zhiguang Chen, Zibin Zheng, and Yutong Lu. Phishing scam detection on ethereum: Towards financial security for blockchain ecosystem. In *IJCAI*, volume 7, pages 4456–4462, 2020.

[14] Sijia Li, Gaopeng Gou, Chang Liu, Gang Xiong, Zhen Li, Junchao Xiao, and Xinyu Xing. Tgc: Transaction graph contrast network for ethereum phishing scam detection. In *Proceedings of the 39th Annual Computer Security Applications Conference*, pages 352–365, 2023.

[15] Zhen Chen, Sheng-Zheng Liu, Jia Huang, Yu-Han Xiu, Hao Zhang, and Hai-Xia Long. Ethereum phishing scam detection based on data augmentation method and hybrid graph neural network model. *Sensors*, 24(12):4022, 2024.

[16] Jiajun Zhou, Chenkai Hu, Jianlei Chi, Jiajing Wu, Meng Shen, and Qi Xuan. Behavior-aware account de-anonymization on ethereum interaction graph. *IEEE Transactions on Information Forensics and Security*, 17:3433–3448, 2022.