Combining Multiple Classifiers using Ensemble Method for Anomaly Detection in Blockchain Networks: A Comprehensive Review

Sabri Hisham, Mokhairi Makhtar and Azwa Abdul Aziz
Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, 22000 Terengganu, Malaysia

***Abstract*—Blockchain is one of the most anticipated technology revolutions, with immense promise in various applications. It is a distributed and encrypted database that can address a range of challenges connected to online security and trust. While many people identify Blockchain with cryptocurrencies such as Bitcoin, it has a wide range of applications in supply chain management, health, Internet of Things (IoT), education, identity theft prevention, logistics, and the execution of digital smart contracts. Although Blockchain Technology (BT) has numerous advantages for Decentralized Applications (DApps), it is nevertheless vulnerable to abuse, smart contract failures, security, theft, trespassing, and other concerns. As a result, using Machine Learning (ML) models to detect anomalies is an excellent way to detect and safeguard blockchain networks from criminal activity. Adapting ensemble learning methods in ML to create better prediction outcomes is a viable approach for anomaly identification. Ensemble learning, as the name implies, refers to creating a stronger and more accurate classification by combining the prediction results of numerous weak models. As a result, an in-depth evaluation of ensemble learning methodologies for anomaly detection in the blockchain network ecosystem is applied in this paper. It comprises numerous ensemble methods (e.g., averaging, voting, stacking, boosting, bagging). The review collects data from three established databases, which are Scopus, Web of Science (WoS), and Google Scholar. Specific keywords are employed, such as Blockchain, Ethereum, Bitcoin, Anomaly Detection, and Ensemble Learning, employing advanced searching algorithms. The results of the search found 60 primary articles from 2017 to 2022 (30 from Scopus, 20 from the WoS, and 10 from Google Scholar). Based on these findings, we decided to divide our debate into three primary themes: (1) the fundamentals of Blockchain Technology (BT), (2) the overview of ensemble learning, and (3) the integration and analysis of ensemble learning in blockchain networks for anomaly detection. In terms of awareness and knowledge, the results are also discussed in terms of what they mean and where future research should go.**

Keywords*—Blockchain; Ethereum; Bitcoin; ensemble; anomaly detection

I. INTRODUCTION

Nowadays, most agencies have started evaluating Blockchain Technology (BT) in various sectors such as pharmaceuticals, automotive, agri-food, livestock, supply chain, health, and government digital initiatives [1]. This scenario has an impact in the context of traceability, transparency, and trustworthiness values in distributed and decentralized ecosystem environments [2]. A Blockchain operates based on a data structure storage method consisting of blocks that are interconnected with each other using a cryptography hash mechanism. Technically, each block stores information such as timestamp, Merkle root, nonce, previous hash and difficulty in the block header [3]. From the point of view of decentralized Blockchain applications, the world of cryptocurrency has become popular and dominant. Thus, Bitcoin BT has forged success by producing the first cryptocurrency application. It is different from Ethereum, which introduced smart contracts, and Ether has been declared the second largest cryptocurrency after Bitcoin [4]. Additionally, Ethereum was created to address the Bitcoin protocol's functional insufficiency [5]. Technically, the Ethereum network hosts smart contracts, which are collections of code that run on the Blockchain and carry out a set of instructions. These contracts are what power Decentralized Applications (DApps), which are akin to smartphone apps that operate on Google (Android) or Apple (iOS) operating systems.

In a public blockchain network, all transactions are transparent and are publicly available. Hence, anyone in the network can examine these transactions and may cross-verify any fraudulent behavior. Along with its rapid development, BT has encountered several security issues and shortcomings, including majority attacks, forking, and bugs in smart contracts. Wallet attacks, Ponzi Schemes, Proof of Work (PoW) vulnerabilities, and crypto-jacking are all challenges that need to be addressed. For instance, the Ethereum Blockchain has increased in prominence. Nevertheless, it has been beset by security vulnerabilities such as phishing scam, which has accounted for nearly half of all criminality on the platform since 2017 [6]. Therefore, for an efficient functioning of a blockchain network, it is vital to detect these vulnerabilities in the most precise and timely manner. To enable the successful identification and prediction of such attacks over Blockchain, the field of anomaly detection models in the Machine Learning (ML) method for Blockchain comes into play.

In general, an attempt to detect an anomaly in a pattern or thing that is different from the norm is termed anomaly detection. [7]. This demonstrates that combining ML and BT has a good impact and is widely employed in industries such as automotive, health, decentralized finance (DeFi), supply chain, agriculture, and the Internet of Things (IoT). Both technologies are combined for goals such as detecting suspicious activity, cybercrime and fraud. Besides, a

Blockchain system that can handle massive data sets is compatible with ML approaches to data analysis and can increase data security [8]. Therefore, a huge variety of anomaly detection models are being designed and deployed by researchers for various Blockchains. However, one of the most difficult aspects of detecting fraud on the Blockchain is that it is anonymous [9].

Overall, it is necessary to note that anomaly detection is one of the important areas for protecting future blockchain networks and that a considerable amount of work is being undertaken on this subject from many views, which will be described in this paper. Ensemble approaches are prominent ways of increasing the prediction capacity of an ML model for anomaly detection. In theory, ensemble learning techniques use multiple classifier methods to improve experimental outcomes. Conventional methods that use a single classifier to perform predictive analysis are ineffective. Therefore, combining individual classifiers in an ensemble can produce higher accuracy values [112]. For instance, strategies include stacking, averaging, bagging, and boosting approaches [10].

This research focuses on the fundamentals of BT, ML classification, and the combined contribution of ML and Blockchain to detect irregularities utilizing ensemble techniques. To aid comprehension, the study is divided into three sections: (2) Blockchain principles, (3) an overview of ensemble learning classification, and (4) developing the ensemble learning method for anomaly detection in blockchain networks.

II. Blockchain Technology

A. Overview

Blockchain is presently one of the most promising technology trends, with great possibilities across many useful applications. It is basically a distributed and encrypted variation of a database, which can solve several difficulties connected to online security and trust. As a result, the Blockchain feature of securely and decentralized data management makes Blockchain known in the world of cryptocurrencies such as Bitcoin and Ether (Ethereum). Historically, the goal of producing interference-proof texts led to the development of a cryptographic hash formatting system for storing documents in a chain of blocks [11]. In this endeavour, hash-based cryptographic algorithms are used to store a collection of verified documents in Merkle tree format in each block [12]. Moreover, since it was invented and exploited in cryptocurrencies like Bitcoin, which was presented by Nakamoto [1] in 2008, this technology has become well-known. This has helped popularize Bitcoin as the first digital electronic payment mechanism that operates on a peer-to-peer (P2P) basis and in a decentralized ecosystem. The field of Blockchain has been divided into four categories: Private, Public, Hybrid, and Consortium. Fig. 1 depicts the categorization of Blockchain.

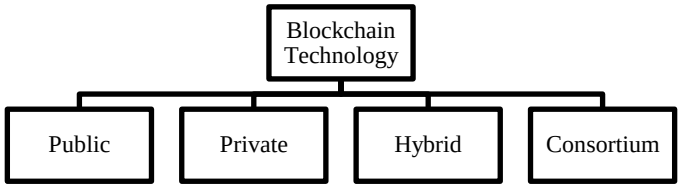

Fig. 1. Blockchain Classification.

A Public Blockchain is non-restrictive and permissionless [13]. This means anyone can do the mining process, and these transactions involve the addition of new blocks settled through a consensus mechanism. This concept has been fundamental to the existence of Bitcoin and other cryptocurrencies in the Distributed Ledger Technology (DLT) ecosystem [1]. As a result, the weakness of the centralized operating system faces challenges in terms of low-security level and no value of transparency (dependence on third parties). Regarding data storage, the DLT ecosystem stores data distributed across nodes linked by a blockchain network, as opposed to a centralized system that stores data in a single location. Technically, the consensus mechanism is an important algorithm in Blockchain operations to ensure that members joining a blockchain network agree on certain conditions before the ledger is updated. Proof of work (PoW) is a common consensus algorithm used in Public Blockchain environments. One of the benefits of this consensus is that as the number of miners grows, attacks can be reduced to 51 percent [16].

In contrast to the Public Blockchain, the Private Blockchain operates based on an organization through access granted only to be allowed to enter the network. Therefore, they are also called "permissioned blockchains" or "business blockchains" [17]. It has the same properties as a Public Blockchain that is distributed, decentralized, and operates in a P2P environment. Typically, a Private Blockchain is used in a network environment with a small organization compared to a Public Blockchain, where anyone has the right to enter a public network. The consensus algorithm used in the Private Blockchain (permissioned) is Practical Byzantine Fault Tolerance (PBFT).

Using both Public and Private Blockchain features in Blockchain development is necessary in the real world. As a result, a Blockchain ecosystem known as Hybrid Blockchain [18] has emerged. Elements from the Private Blockchain (permissioned) are employed in the enterprise context. On the other hand, a Public Blockchain is ideal for practice since the data requirements are open or public (permissionless). The addition of the participation of several organizations from a single organization so that the value of collaboration is higher in a Private Blockchain environment is termed a "blockchain consortium" [18]. It combines features of a Public and Private Blockchain and is very similar to a Hybrid Blockchain. An important goal is to eliminate access gaps limited to a single organization in a Private Blockchain environment.

B. Blockchain Architecture

In a decentralized ledger, all transactions in a Blockchain are stored in interconnected blocks. Each block contains a block header that stores critical information, including the timestamp, nonce, difficulty, block hash, and Merkle root tree, to keep these blocks related. This method guarantees the security of the data within the blocks, and the size of the witness determines the size of each block. One Bitcoin block, for example, is 1 MB in size [1]. Meanwhile, the Merkle tree employs the hash technique for each block transaction, as shown in Fig. 2. From an operational point of view, each block stores the address of the parent block or the previous block in the form of a hash value. This mechanism can help to identify the chain sequence between these blocks. Blocks generated in the early stages of blockchain network construction are termed "block genesis." To ensure the uniqueness of each block, the timestamp information is crucial to store the time differentiation generated on each block. For example, the current block has a more recent timestamp value than the timestamp of the previous block. This mechanism can prevent the occurrence of double-spending cases.

Blockchain environments, especially Bitcoin, are known for mining processes using pseudo-random numbers (nonce) and are used only once throughout the mining process. Note that it is difficult to keep the value of the difficulty level based on a threshold with a specific target. For example, the difficulty level rises when the number of transactions increases. As a result, block formation becomes increasingly complex (mining process) and slower. It also affects cyber attackers and greedy miners who want to take advantage of many transactions and slow the processing. The Merkle tree cryptographically manages the hash mechanism on transactions in blocks. This is described as a tree consisting of leaves as well as twigs. Conceptually, the hash in the brand tree is constructed based on a combination of left and right hashes to produce the parent hash. The generation of interconnected hashes forms a chain called a Blockchain. Therefore, an abnormality in the Merkle tree indicates something is happening in the chain, and appropriate action is taken immediately [19].

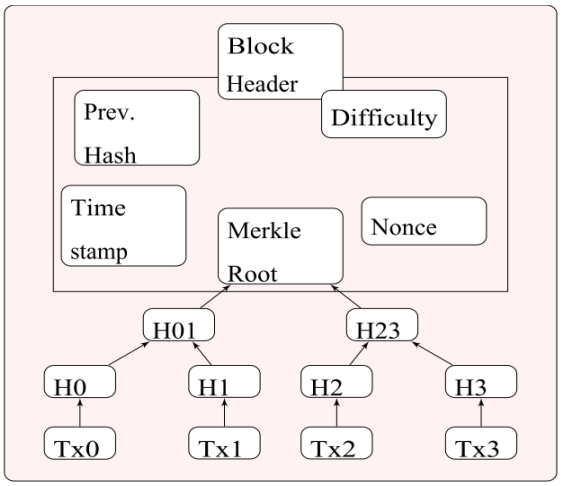

Fig. 2. Block Architecture [2].

C. Blockchain Layers

From the Blockchain Technology (BT) layer perspective, there are six layers in the blockchain network, as depicted in Fig. 3. The blockchain network contains several layers to execute specialized activities [20,21]. The data layer provides cryptographic techniques that store data in the hash, Merkle tree, and timestamp value forms in both on-chain (Blockchain) and off-chain (database) settings. The network layer manages all of the nodes in the blockchain network. At the network layer, this level of security and privacy is made sure to stay in place by a decentralized P2P environment. At the same time, transaction consistency is managed by consensus mechanisms located at the consensus layer. The mining process rewards successful miners. It is managed in the incentive layer. The condition of the smart contract in the Blockchain ecosystem is important to ensure that the security aspects are guaranteed, bug-free, and free from any vulnerabilities. Therefore, the smart contract programme is implemented at the contract layer. The application layer, which connects the end-user to the blockchain network, is the final layer. This layer comprises Blockchain applications (Decentralized Applications (DApps)) that were designed and constructed based on the business case in various sectors.

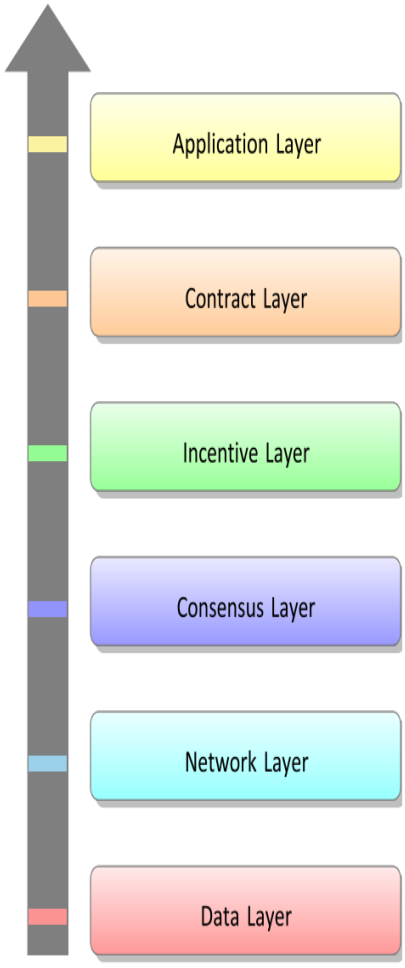

Fig. 3. Six Blockchain Layers [3].

D. Consensus Algorithm

The blockchain network must verify each software's ledger for consistency and clarity. This is performed by a few steps that follow certain rules during the transaction process. The verification process is carried out decentralized, with transactions completed in a distributed environment managed by P2P-connected nodes in the network. The approaches or algorithms utilized to reach a consensus are called consensus algorithms. Fig. 4 shows various widely used consensus algorithms, including Proof of Authority (PoA), PoW, Proof of Stake (PoS), and PBFT. Each node seeking to participate (mining) in the PoW consensus process must contribute resources by completing mathematical problem challenges [14]. This problem has a different level of difficulty. It is a consensus technique used in Bitcoin [1] and Ethereum [22]. In PoS, only one miner can generate new blocks from all participating nodes, while other miners waste incentives and energy resources on the blockchain network [15].

As a result, PoS works better when only those nodes can verify that their shareholders are permitted to participate. It avoids the circumstance where one node owns the network since no single node may hold 51 percent of the network's money[23]. As a result, PoS can efficiently cut energy consumption and reduce the number of miners, and the transaction speed can be boosted compared to PoW. It is critical to obtain mutual understanding in the PoA consensus to ensure the transaction is valid. The node's blocks must be certified by the verified node, and the process continues through the successive rounds as planned [24]. The PBFT consensus refers to a Byzantine military analogy that is difficult to reach consensus if no nodes have reached an agreement. The effort to reach this agreement based on the leaders with the most weight is called the PBFT consensus [25].

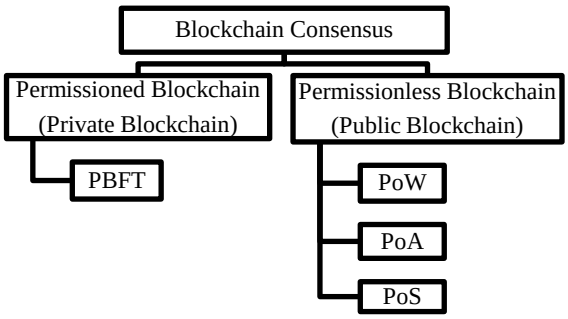

Fig. 4. Blockchain Consensus Algorithm.

E. Bitcoin

Cryptocurrency is one of the most extensively used Blockchain applications today and is used worldwide. Bitcoin and Ether (Ethereum) are two digital currencies commonly used in the crypto realm. Satoshi Nakamoto was the first to introduce Bitcoin, successfully solving the double-spending problem while introducing digital currency use [1]. The Blockchain controls each transaction using a cryptographic process based on hash values on input and output sets from an operational standpoint. Only one input transaction from the whole blockchain network is used to generate the output [26].

Aside from that, Blockchain is linked to a P2P ecosystem for transaction management and network ownership. The decentralization of Blockchain is a clear distinction between traditional databases and Blockchain. This implies that each network node is accountable for storing a copy of the ledger [19]. In the Bitcoin ecosystem, anyone can participate in the network. This feature is why Bitcoin is known by the term "incentive" or "reward" through the PoW consensus given to miners who successfully perform the mining process. As such, this Blockchain operates in a decentralized manner, which means it does not require a centralized body compared to traditional financial systems, which are centralized in nature. In this process, the miner gets paid a few Bitcoins after completing the operation.The mining process is secure because it involves hashed and encrypted transactions using the SHA-256 cryptographic technique. The popularity of Bitcoin as a Blockchain application for managing cryptocurrencies has prompted the development of several other crypto and DApps.

F. Ethereum

Buterin's paper [27] launched Ethereum and solved various problems with Bitcoin's scripting language. Ethereum had added transaction list and state information in the block header compared to before, which only contained information such as nonce, difficulty, and block number. A new state will be formed based on the previous state in the transaction list. The notable difference between Bitcoin and Ethereum is the cryptographic protocol used. Ethereum uses Keccak 256 bits while Bitcoin uses SHA-256. Thus, the header block in Ethereum consists of hashes for gas fee information, timestamp, parent block header, root state, and additional hashes for verification process purposes [28]. Ethereum provides a decentralized ecosystem for developers to develop products using the Solidity language and Ethereum Virtual Machine (EVM). The Solidity language is used to develop smart contract programmes based on business cases to be executed and converted to byte code in EVM [26].

G. Smart Contract

Historically, the idea of contract management has traditionally inspired the introduction of digital smart contracts by the founder of smart contracts, Szabo [29]. The main purpose of digital smart contracts is to automate traditional contract management. This smart contract is referred to as computer technology with the help of writing programme code to be implemented to automate the contract process. For operational purposes, smart contracts are integrated with Ethereum to be executed and stored in a decentralized ledger. Recently, the use of smart contracts has been widely used in conjunction with BT in various fields [61, 62]. Furthermore, the EVM environment and the Solidity programming language facilitate the development of smart contracts within Ethereum. This development has also attracted researchers to explore smart contracts on the Blockchain.

III. Ensemble Method

Machine Learning (ML) algorithms have been widely applied in both supervised learning and unsupervised learning situations to construct systems capable of making realistic decisions in light of past data. Numerous classification-based ensemble methods have been developed to boost the accuracy of supervised Learning Algorithms (LAs). Therefore, ensemble methods are prominent solutions for boosting the prediction capacity of an ML model. In the competition aspect, the ensemble approach has succeeded in several ML model competitions in which it has participated. For instance, the winner employed an ensemble method to create a robust collaborative filtering algorithm in the popular Netflix Competition [30]. Another example is Knowledge Discovery in Databases (KDD) 2009 when the winner also used ensemble methods [31].

Conceptually, the ensemble approach combines several trained individual classifiers to produce a new classifier. Typically, these individual classifiers are termed weak learners, and their ensemble combination aims to make this model stronger in terms of accuracy. However, among the challenges of using the original model individually is exposure to high variance and bias factors. Therefore, the ensemble strategy can reduce the bias and variance gaps to produce new combinations with better performance results, as illustrated in Fig. 5.

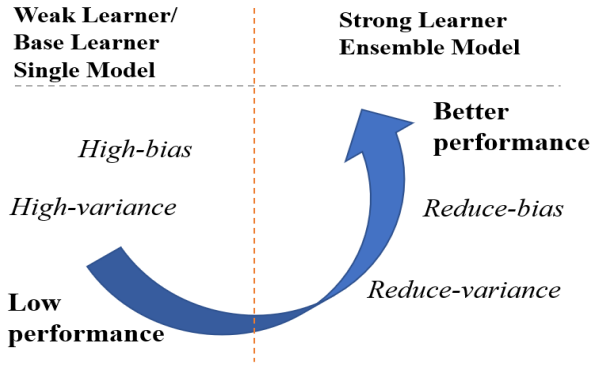

Fig. 5. Weak and Strong Learners.

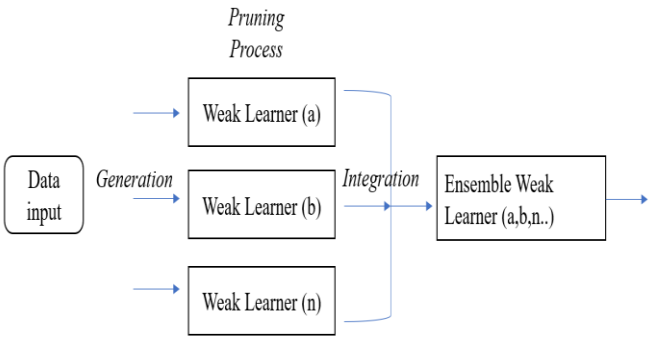

Fig. 6. Typical Process of the Ensemble Method.

With reference to Fig. 6, the process of ensemble generation from data input takes place in the first phase to produce weak learners. Next, the pruning and cleaning process is done for the weak learners. Finally, the combined integration of the weak learners is implemented in the last phase using the selected model. Past research has proven the ensemble approach successfully produces more accurate study results and lower false positive (FP) metrics than individual classifiers. The study also shows that popular ensemble strategies are stacking, bagging, and boosting. The authors [10] has described the ensemble as a variety of combined approaches consisting of the voting method, the averaging method, the stacking method, the bagging method, and the boosting method. According to [32], the ensemble approach can address the shortcomings of traditional ML, such as mathematical, computational, and representation problems. Fig. 7 depicts the ensemble learning methodology and methods. Moreover, the authors explain an ensemble as a model that incorporates the results from numerous other models to remedy the flaws of every situation. Most of this strategy's options can be classified as bagging or boosting [33]. In the averaging approaches, the authors [34] tests with different alternatives of anomaly detection models. The authors believe that choosing a simple average score between different algorithms is a simple and successful solution. Apart from that, the authors define combining the multiple models as needed because they address the problem from diverse aspects [34]. Using ensemble learning, the combination of Random Forest (RF), Extra Trees, and Bagging classifier demonstrated a possible performance by gaining the predictions based on averaging the probabilities derived from these methods [35]. The authors [36] describe how the results generated from the individual classifiers have enhanced their capabilities and have shown improved performance on the study results through the ensemble method. Meanwhile, the study by [113] used a Deep Learning (DL) approach to produce prediction analysis with an ensemble combination for a single classifier based on medical datasets. The study results show that the ensemble technique produces high accuracy values compared to the individual classifiers. Nowadays, more studies lead to new methods or techniques for model optimization compared to before, which is more to developing new models. Among them is a study conducted by the authors [114] using ensemble techniques to develop a new model optimization method for the prediction of taxological applications. The experimental results in this study show that the ensemble technique produces better results than the single classifier.

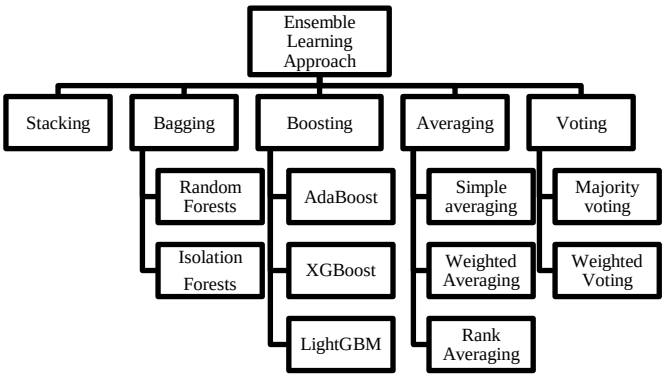

Fig. 7. Ensemble Learning Approach.

A. Voting

Voting is the easiest ensemble procedure. Among the main techniques of the voting ensemble is the majority voting ensemble, sometimes called the max voting ensemble. This is an ensemble strategy that combines multiple different types of individual classifiers. The desire to increase performance from individual models is an essential strategy. For classification and regression, ensemble voting might be used. The mean value of the forecast is derived using the regression approach. In the classification approach, labelling is based on the number of prediction outcomes tagged and the majority of votes. In practice, ensemble voting is appropriate when all individual models show good performance. Fig. 8 illustrates a voting ensemble learning illustration. In a study by [37], the majority voting-based ensemble model method was used. The results successfully detected network traffic as if there had been an attack on the Intrusion Detection System (IDS). In this research, the authors [37] mentioned that many classifiers were employed for training and testing, and final findings were attained utilizing the voting approach. Aside from the majority vote approach, the researchers chose to perform the investigation using the weighted voting method. Repeated calculations on the model prediction are used in the weighted voting method to produce a favourable result from the standpoint of the ballot weights. In the current work, weighted majority voting was used to categorize the data, where Particle Swarm Optimization (PSO) was employed for allocating weights to several classifiers [37].

B. Averaging

Using the averaging method, the simplest strategy for making predictions from dataset inputs is based on average values. In general, this method generates a better regression model and reduces overfitting. Nevertheless, this averaging variant is slightly modified to be a weighted average model. The prediction generated from this model is calculated based on the average value generated from the multiplication operation by the weights on each model. Rank averaging is the process of allocating ranks to individual models based on the weight to be assigned to each model. The method of averaging and determining the maximum score is one of the combination methods that can be used. The findings of the pilot experiment reveal that weighted averaging has been utilized to normalize the anomaly scores. This is done before combining the method to balance the results of unbalanced for different algorithms with different datasets [38]. The weighted average is the result of the study's final output based on the method of grouping the list of scores and assigning a weighting value that is inversely proportionate to the group size possessed by each list of scores, according to [39]. Fig. 9 illustrates the average ensemble learning demonstration.

C. Stacking

Stacking, or layered generalization, is an alternative way of integrating numerous models. In the stacking technique, various individual (multiple) models have been integrated. Among them are logistic regression (LR), Naïve Bayes (NB), and Decision Tree (DT). The learning approach of stacking is for merging the expectations of several classification models into a single meta-classifier [31]. Meanwhile, the authors [40] explained that stacking techniques in the ML approach could produce a more powerful model. This is implemented through training on datasets on individual models to improve accuracy. Basically, the stacking method uses the predictions made by a single model to make another model.

From an operational point of view, the stacking technique is carried out sequentially. The process begins by training several selected individual models using a dataset sample. Subsequently, the production probability results from each individual model go through a fine-tuned process before being combined into a final model. This procedure is performed repeatedly depending on the number of stacking layers you want to use. Finally, the final output is formed based on the final output generated by several individual models in the last layer. Therefore, the individual models generated at this end layer are known as meta-classifiers. According to [41], the learning output at the base layer determines the final output produced by the stacking method. Fig. 10 depicts the usual two-layer stacking modelling approach.

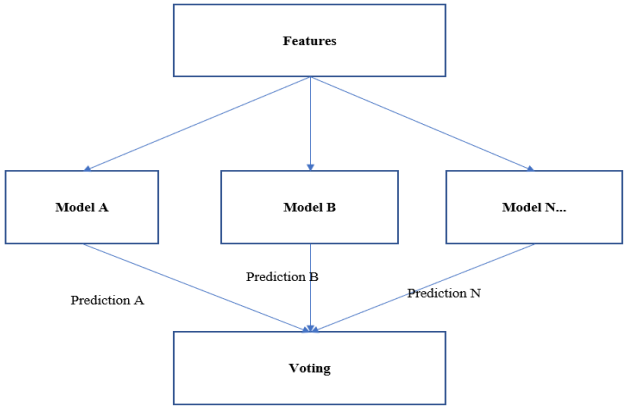

Fig. 8. Voting Ensemble Learning Illustration.

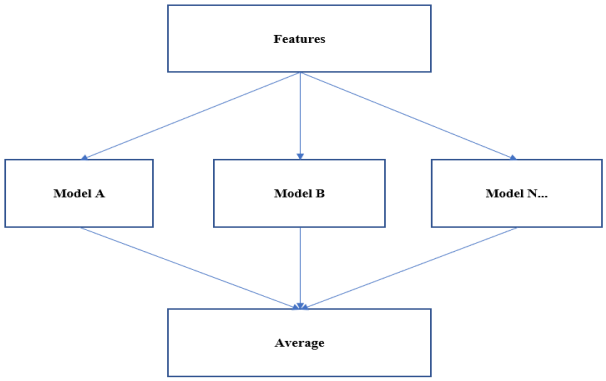

Fig. 9. Averaging Ensemble Learning Illustration.

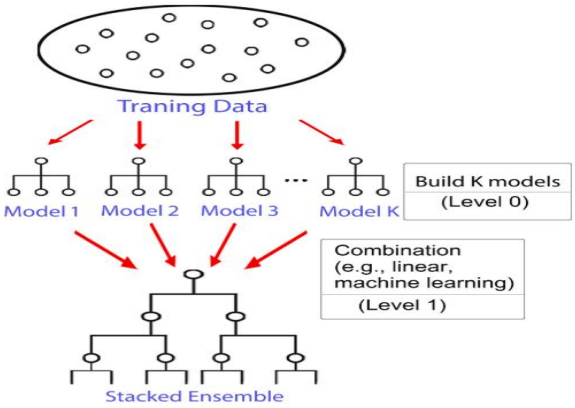

Fig. 10. Stacking Ensemble Learning Illustration[6].

D. Bagging

The bootstrap aggregating (bagging) was first described in [43]. It is one of the simplest ensemble approaches and is best suited for issues involving small training datasets. Sequential and parallel ensemble methods are the two predominant paradigms for constructing ensemble models. Technically, various series of datasets are formed through random extraction from samples of the original data set, and these data sets are used to train different models. Then, voting is used to aggregate the results of the models to form a single output. Bagging is used in regression and classification to improve the precision of ML algorithms. Besides, bagging also utilizes the most prevalent techniques for combining the outputs of base learners, namely averaging for regression issues and voting for classification tasks. Among the algorithms commonly used in the bagging technique is the DT. According to [44], this algorithm can be compatible with weak models and have high variance. However, apart from the DT, other model classifications such as K-Nearest Neighbour (KNN) and NB are also used in the bagging technique. Furthermore, creating a model using a simple method that incorporates large and complex data is impossible. Consequently, bagging approaches are ideal for managing both high-dimensional and large-capacity data. Fig. 11 depicts an illustration of the Bagging algorithm procedure.

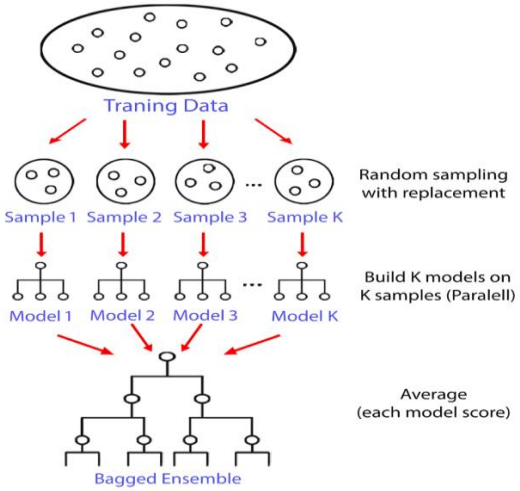

Fig. 11. Bagging Ensemble Learning Illustration[6].

1) Random Forest (RF): RF was introduced in 2002 by Breiman. The Random Forest is, as its name suggests, a forest comprised of numerous trees. In general, RF (Tree-Based) use a DT as an individual model, which generates a set of random parameters as the value of dependence on each tree. Similar to other ensemble algorithms, RF produces predictions by combining numerous separate models. Basically, the RF procedure consists of multiple steps. First, bootstrap samples were randomly generated from the dataset. Then, the prediction results of each tree will be obtained from the construction of the DT based on the data sample. Lastly is the implementation in the voting phase to produce the final output. In this last phase, the model that gives the most accurate prediction results will be selected [45].

2) Isolation Forest (IF): The Isolation Forest (IF) algorithm was first proposed in 2008 [46]. Like any other tree ensemble method, this approach is based on DT. It operates on the premise that an individual who is easier to distinguish from others in a random sub dataset of the feature space must be an outlier. It begins by drawing a random sample from the dataset and selecting a random dimension. Correspondingly, a random value within the range of that dimension is selected to precisely divide the sample into two pieces. Next, the root node of a tree is built using the selected dimension and splitting point. Further nodes are produced recursively for subsamples until a subdivision is impossible or an arbitrary tree depth is attained. In this tree, a point closer to the root node correlates to a situation more likely to be isolated. Nevertheless, this could be due to random chance. Therefore, the entire tree generation technique is repeated for additional samples until the necessary number of trees is achieved. Note that the anomaly score is computed using the mean traversal path length of the trees. The authors of [46] claim that their algorithm is superior to other alternatives for addressing masking difficulties (clusters of anomalies) and swamping problems (mistakenly identifying normal situations as being surrounded by anomalies).

E. Boosting

Boosting is a strategy for enhancing the performance and accuracy of the ML approach by transforming weak base learners into strong ones [47] as shown in Fig. 12. The fundamental premise of the boosting strategy is to sequentially add new models to the ensemble. In general, the boosting technique generates a sample of training data randomly with the replacement of the main dataset sequentially. In this procedure, a sequence of models is learned. The process begins by providing training on the weak model using a training dataset to produce a second model after fixing the weaknesses in the first model. Subsequently, a third model was produced that overcame the weaknesses of the previous two models. This process will continue until all the mistakes are fixed and the final model is made. Last, a technique weighted majority voting was used to build the final model from the weak model [48,49]. Boosting techniques have been proven to increase accuracy and reduce bias and variance. Among the algorithms widely used in boosting techniques are

Adaptive Boosting (AdaBoost), Extreme Gradient Boosting (XGBoost), and Light Gradient Boosted Machine (LightGBM).

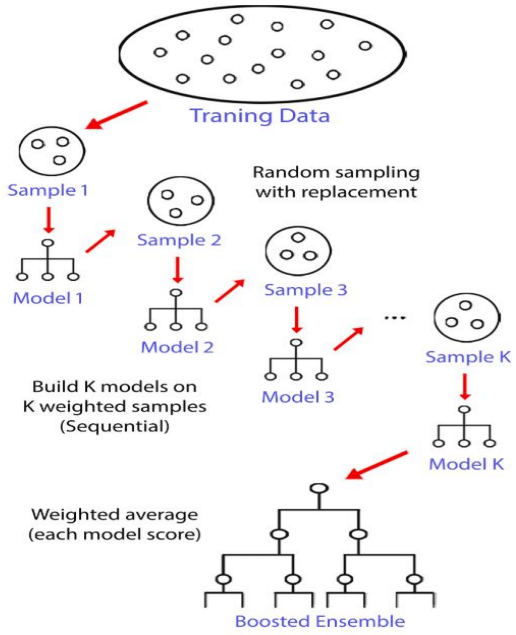

Fig. 12. Boosting Ensemble Learning Illustration [6].

1) Adaptive Boosting (AdaBoost): AdaBoost was the first truly successful binary classification boosting method. It was originally referred to by its inventors as AdaBoost.M1 [50]. Recently, it has been referred to as "discrete AdaBoost" because it is utilized for classification instead of regression. AdaBoost, like other approaches, may be used to increase the performance of any ML model and can be used for learners with low intelligence. This strategy works by turning weak learners into strong ones by getting rid of them by correcting their mistakes over and over again iteratively. The weighted training dataset is used to train weak learners in succession. Subsequently, numerous weak learners are joined to become a single powerful learner. Finally, the weight voting method on the weaker model was used to determine the stronger final model [50]. Besides, one-level DT is the best-suited and, thus, the most popular algorithm employed with AdaBoost. Since these trees are so short and contain only one classification decision, they are often referred to as decision stumps.

2) Extreme Gradient Boosting (XGBoost): Extreme Gradient Boosting, or XGBoost, is a scalable ML approach for tree boosting that was presented by Chen and Guestrin [51]. XGBoost is a gradient boosting-based model that uses additional boosting strategies to produce predictions more accurately compared to other gradient boosting models [52]. Therefore, the advantages of this technique have been acknowledged in various fields of ML and data science. For example, a total of 17 winners used the XGBoost technique out of a total of 29 winners to complete one solution contest as well as be featured in the Kaggle blog [53]. XGBoost uses the advantages of boosted tree algorithms to produce accurate and scalable boosting gradients. Moreover, XGBoost has been designed with fast computer processing and improved ML model performance in mind. In general, XGBoost works in parallel to generate trees. This process is implemented level by level to produce predictions on each iteration from weak learners. As a result, each of these iterations can improve the errors of their predecessors. The final result of prediction with a combination of individual models and these mechanisms is the same as with other ensemble approaches.

3) Light Gradient Boosted Machine (LightGBM): LightGBM, or Light Gradient Boosted Machine, was described by Guolin Ke et al. in 2017 [54]. LightGBM is a gradient boosting implementation aimed to be efficient and possibly more successful than previous gradient boosting implementations. According to the authors [54], the solution includes two main concepts: 1) Gradient-based One-Side Sampling (GOSS); and 2) Exclusive Feature Bundling (EFB). GOSS is a variation on the gradient boosting approach that prioritizes training samples that provide a greater gradient, accelerating learning and minimizing the method's computing complexity. In contrast, EFB is a method for combining sparse (mainly zero) mutually exclusive features, such as one-hot encoded categorical variable inputs. Consequently, this is a form of automatic feature selection. Through this concept, LightGBM has adapted a tree algorithm capable of producing high performance, classification, ranking, and various tasks in ML. Besides, LightGBM is a fast, more efficient, less memory-intensive, more accurate than any other boosting algorithm, compatible with large datasets, and gradient boosting framework. Normally, the DT through the boosting method is determined based on their level or depth. Nevertheless, this approach differs from LightGBM, which divides the tree based on the optimal leaf. Therefore, this approach provides a high level of accuracy by minimizing the level of loss and is an achievement that is rarely achieved by any existing booster algorithm.

IV. Ensemble Anomaly Detection in Blockchain

Nowadays, the development of Blockchain Technology (BT) is not just focused on the world of cryptocurrency but its expansion to Decentralized Applications (DApps) in various fields. Following this, the features available in BT have provided advantages in terms of transparency, immutability, enhanced security level, fast transactions, and high privacy. As a result, we see many applications that use BT in various sectors, namely finance, supply chain, halal products, pharmaceuticals, education, government, etc. In cryptocurrency, Bitcoin and Ethereum are the most popular and widely used applications due to their high market capitalization and trading volume. Apart from that, Bitcoin constitutes about 39.53 percent of the market's entire value [55]. At the same time, Ether is the second-biggest cryptocurrency [3]. Meanwhile, Ethereum is the largest and most widely used decentralized Blockchain platform for smart contract adaptation. The widespread use of Bitcoin, as well as Ethereum, has given rise to some critical issues in the aspects of cybercrime and security. As a result, many have become

victims of various frauds, such as phishing and Ponzi Schemes, after detecting more than 10 percent of Initial Coin Offering (ICO) on Ethereum. Generally, the Ethereum blockchain network is a public distributed ledger with around 1.158 million daily transactions [56] and is categorized as big data. Therefore, manually combing through all of these transactions to find any transactions suspected of exhibiting unusual characteristics would be impracticable and interminable. Based on this scenario, Machine Learning (ML) algorithms would help differentiate between transactions that exhibit normal and abnormal behavior among user accounts by learning the attributes that correspond to either normal or abnormal conduct.Therefore, an approach to detecting transactions that show abnormalities was introduced, known as the abnormal detection method. Nowadays, this method is increasingly used in various fields to detect patterns of abnormalities, especially its role in the Blockchain ecosystem. The detection model developed using the ML model helps detect and predict the initial attacks on the blockchain network. Fig. 13 offers data visualization for normal and anomalous transactions to better understand anomaly transactions. Oddities or unusual occurrences have the same meaning as deviations, noise, novelties, exceptions, and outliers [7]. Clearly, the combination of Blockchain and ML technology positively benefits both parties, as shown in Fig. 14. The Blockchain ecosystem is known for its overly large data storage nature and can be declared big data. There is also data from external sources such as smart devices, the Internet of Things (IoT), and external applications that store data in a database (off-chain). Thus, data from various sources is analyzed using ML techniques to produce analytical dashboards, predictions, visualizations, and others that can help with planning, monitoring, and decisions.

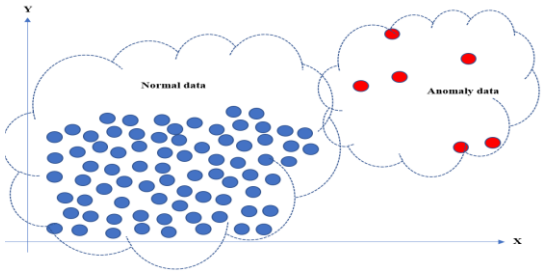

Fig. 13. Data Visualization for Normal & Anomaly.

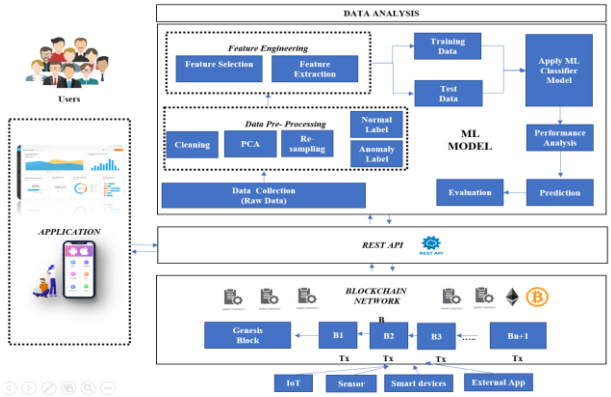

Fig. 14. Connection between Blockchain and ML.

In earlier study, numerous ML algorithms have been applied in supervised [57] and unsupervised learning [58] for anomaly detection in blockchain networks. Random Forest (RF) [59], Decision Tree (DT) (j48) [60], Extreme Gradient Boosting (XGBoost) [61], Adaptive Boosting (AdaBoost) [62], secureSVM [63], Light Gradient Boosted Machine (LightGBM) [64], K-Nearest Neighbour (KNN) [65], Support Vector Machines (SVM) [66], Naïve Bayes (NB) [67] and Isolation Forest (IF) [68] are examples of supervised learning models. Among the models in unsupervised learning that have been utilized are One Class Support Vector Machine (OCSVM) [69], K-means [70], Density Based Spatial Clustering of Application with Noise (DBSCAN) [71] and Long Short Term Memory (LSTM) [72]. This article evaluates the ensemble learning method for detecting anomalous or criminal transactions in blockchain networks. Ensemble learning gave good results and great performance in the experiments for recognizing malicious Ethereum entities [73]. Moreover, the authors execute ensemble learning, a mixture of ML predictors that wins over other classical learning approaches at predicting licit and illegitimate transactions. In the experiment, ensemble learning can be characterized as a classification method based on an average probability ensemble constructed from the collection of best-performing supervised learning methods employed in our experiment [35]. However, individual classifiers are troublesome for processing high-complexity data, according to [74] research. Consequently, this issue has been handled by developing a classification model utilizing the ensemble approach. In a Proof of Concept (PoC) development project for the decentralized unmanned aerial vehicle (UAV), the ensemble stacking method was applied to a variety of individual models to assess its predictive accuracy [75].The completed literature evaluation led to the classification of prior research articles about the addressed applications published from 2017–2022. Publications were divided into four aspects: anomaly detection in cybercrime (see Table I), security (see Table II), information processing (see Table III) and smart devices (see Table IV).

A. An Anomaly in the Aspect of Cybercrime

Cybercrime means using computers, tools or materials with the intent to do illegal things [76]. BT's openness, transparency, and immutability have prompted malicious parties to commit criminal activities. Most cyberattacks are performed for financial benefits. In the cryptocurrency era, hackers are prompted to get their ransoms in cryptocurrencies, as it provides the advantage of anonymity and easy transfer across countries. Therefore, among the effective methods is to use ML techniques to detect abnormalities in blockchain network transactions. Many previous studies have reported detecting transaction abnormalities using the approach of the abnormality detection method. Thus, in this review, we identified 31 publications that apply the cybercrime aspect in the selected papers, as shown in Table I. Referring to Table I, cybercrime aspects are categorized according to the type of application case, namely smart contracts, illicit transactions, scams (pump and dump), fraud detection, ransomware, Ponzi Schemes, money laundering, High Yield Investment Program (HYIP), and phishing, as shown in Fig. 15.

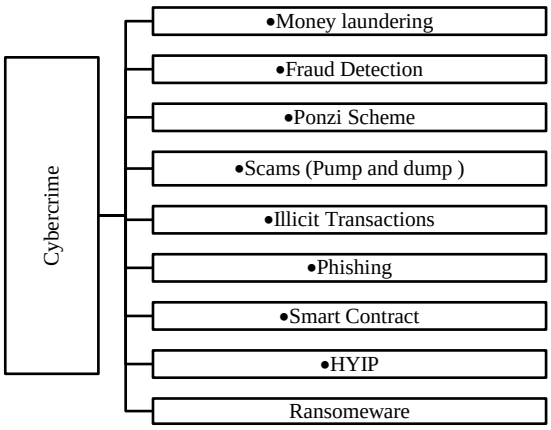

Fig. 15. Classification of Application in the Cybercrime Aspect.

As indicated in Table I, an RF was the most commonly utilized ensemble model (based learner) in the chosen research publications. In addition, 20 research publications utilized bagging as an ensemble approach, and 11 research papers embraced the boosting method. Furthermore, we uncovered 24 research articles utilized in Bitcoin and Ethereum. Since developing an ML model relies on the dataset, we analyzed the data source of ML models for anomaly detection applied in the selected research publications. The analysis of data sources has shown that 30 different types of data sets were used in the experiment. In earlier investigations, it was observed that there are numerous ways employed in the ensemble learning method. Among them is combining ensemble approaches or tactics to produce a good result. The review papers describe numerous techniques for this hybrid scenario, including bagging with voting, bagging with averaging, bagging with boosting and bagging with stacking.

The authors [77] suggested a pre-encryption detection algorithm (PEDA) that seeks to identify ransomware using an ML approach to assess and categorize ransomware using the bagging and voting (majority voting) ensemble learning technique. This research was conducted in Phase 1 and Phase 2 using a dataset created by Resilient Information System Security (RISS) from Imperial College, London. Nevertheless, the focus of this study is the focus on Learning Algorithm (LA) implemented in Phase 1. In general, LA works through an ensemble DT approach. First, the simulations of the LA model were implemented using the Application Programme Interface (API) data generated by suspicious software for inspection. Then, performance measurement analysis was performed by comparing the LA model with three other models, namely NB, RF, and ensemble techniques (RF and NB). Finally, this model was selected using the majority voting method. The results of this experiment have shown that the LA model produces better performance compared to the individual models' RF, NB and the ensemble models (RF and NB). Measurement metrics use detection rate (DR), False Positive Rate (FPR), Under Area the ROC Curve (AUC), and test error values.

TABLE I. SUMMARY OF PREVIOUS RESEARCH USING ENSEMBLE METHOD IN CYBERCRIME ASPECT

Ref.	Year	Blockchain Application	Application	Ensemble method applied	Model/Based learners	Tools/Dataset
[79]	2017	Bitcoin	Fraud detection	Bagging	Random Forest	Public Dataset
[80]	2017	Ripple	Anomaly detection	Averaging	One Class SVM, Gaussian Mixture Models, Isolation Forest	Ripple Transaction dataset
[81]	2017	Bitcoin	HYIP	Bagging, Boosting	Random Forest, XGBoost	Public Dataset
[82]	2018	Bitcoin	Ponzi Scheme	Bagging	Random Forest	Public Dataset Reddit, Bitcointalk.org
[83]	2018	Cryptocurrency	pump and dump scams	Bagging	Random forest	Telegram API Twitter API Crypto Market Data
[84]	2018	Ethereum	Ponzi Scheme	Boosting	XGBoost	Etherscan API/Real Data
[82]	2018	Bitcoin	Ponzi Scheme	Bagging	Random Forest	blockchain.info public dataset (bitcoinponzi)
[85]	2018	Bitcoin	De-Anonymising Entity	Boosting	Gradient Boosting	Chainalysis
[53]	2019	Ethereum	Fraudulent Accounts	Bagging	Random Forest	Etherscan API/Real Data
[86]	2019	Cryptocurrency	Anomalous transactions	Bagging	Random Forest	Etherscan API/Real Data Binance
[86]	2019	Cryptocurrency	pump and dump scams	Boosting	XGBoost	Binance Telegram Data
[87]	2019	Ethereum	Ponzi Scheme	Bagging	Random Forest	Etherscan API/Real Data
[88]	2019	Bitcoin	HYIP	Bagging	Random Forest	WalletExplorer Blockchain.info Xapo.com
[77]	2019	Bitcoin	Crypto-ransomware	Bagging, Voting	Naive Bayes, Random Forest	RISS dataset API Cuckoo Sandbox

						SQL database
[60]	2020	Bitcoin	Illicit entities	Bagging	Tree-based	VJTI Blockchain lab
[89]	2020	Ethereum	Illegal activity	Boosting	XGBoost	Etherscamdb Etherscan API
[90]	2020	Bitcoin	Money Laundering	Bagging	Random Forest	Elliptic
[91]	2020	Ethereum	Fraudulent Behaviour	Bagging	Random Forest	etherscamdb.info
[35]	2020	Bitcoin	Anti-Money Laundering (AML)	Bagging Averaging	Random Forest, Extra Trees, and Bagging classifier	Elliptic
[92]	2020	Ethereum	Honeypot Smart Contract	Boosting	LightGBM	Honeybadger,Ethereum Client,Parity Client
[93]	2020	Ethereum	Ponzi Scheme	Boosting	Ordered Boosting	bitcointalk.org,Google BigQuery,PonziTect
[94]	2020	Bitcoin	Fraudulent Transactions	Bagging	Random Forest	Kaggle
[86]	2020	Cryptocurrency	Fraudulent Transactions	Bagging	Random Forest	Etherscan API
[95]	2020	Cryptocurrency	pump and dump scams	Bagging	Random Forest	Telegram, Twitter, Reddit, BitcoinTalk
[59]	2021	Ethereum	Fraudulent detection	Bagging	Random Forest	Kaggle
[96]	2021	Bitcoin	Fraud Transactions	Bagging	Random Forest	Bitcointalk,bitcoin public dataset
[97]	2021	Ethereum	Fraudulent Detection	Bagging	Random Forest	Google BigQuery Github
[62]	2021	Ethereum	Phishing	Boosting	AdaBoost	Etherscan API
[78]	2021	Ethereum	Fraudulent Transactions	Bagging, Boosting	Random Forest, Adaboost, SVM	node2vec
[98]	2021	Ethereum	Vulnerability Detection	Boosting	XGBoost	Etherscan API
[74]	2022	Cryptocurrency	Anomaly Detection	Boosting, Stacking	SVM, KNN Logistic, DT, MLP	Kaggle

Adapting ensemble techniques has also worked well in networking, where they have been used to predict both licit and illicit transactions [35]. In this experiment, the approach of bagging with averaging technique has been applied to anticipate licit and criminal transactions in the blockchain network. The proposed approach of an ensemble (RF, Extra Trees, and Bagging classifiers) has fared the best with a comparison of RF, Multilayer Perceptron (MLP), and Logistic Regression (LR). In an average probability ensemble, the classification is done by employing numerous pre-trained ML models. The final predictions are formed by averaging the summation of the prediction probabilities received from the LAs. Note that the results demonstrate that ensemble learning is able to execute classification with an accuracy (98.13 percent) and F1 score (83.36 percent) to forecast licit and illegal transactions.

The authors [78] gives a comprehensive evaluation of different supervised ML algorithms, such as bagging models (RF), boosting models (AdaBoost), and others, to prevent fraud. This research concluded that utilizing AdaBoost and RF classifier produced the best performance result among the other seven algorithms.

Feature selection in the ensemble approach plays an important role in producing better results. This has been demonstrated by [74], who conducted studies on the use of feature selection and without feature selection. This simulation is performed by comparing the use of feature selection with that without feature selection in the ensemble classifier (boosting, stacking). The final results have shown that there is an increase in the value of F-Score (7 to 9 percent) and accuracy (2 to 3 percent).

B. An Anomaly in the Aspect of Security

BT does not guarantee freedom from security issues. Therefore, there is a need to establish risk management through a comprehensive cyber security framework and undergo security assessment services to protect against attacks and abuse by hackers. This security issue has been researched and has found a total of 31 research papers involved in the study on the aspect of security, as shown in Table II. This in-depth study uses ensemble techniques to find anomalous transactions in a blockchain network. According to Table II, security elements are largely split into backdoor assaults, vulnerability identification, crypto-jacking, under-priced Denial of Service (DoS) attacks, intrusion detection, miner detection, malware, cybersecurity framework, protection of private information, botnet and malicious account detection, and so on, as shown in Fig. 16.

As shown in Table II, an RF was the most commonly utilized ensemble model (based learner) in the selected research publications. In addition, four research publications utilized bagging as an ensemble approach, two research papers adopted the stacking method, and 1 research study applied to boost and to vote. Moreover, we identified four research publications that have been used in Ethereum. The utilization of datasets is the crucial component of ML model construction. Consequently, this study's analysis considers the datasets utilized in prior studies. As a consequence, it was determined that the selected research utilized five distinct types of data sets. In the ensemble approach, a combination of several ensemble (hybrid) techniques is used to achieve better performance in the study. Among them are: In reviewing investigations for security considerations, it was determined that two research publications used combined ensemble methods or strategies to achieve a decent outcome. In addition, there is one research paper that utilized the stacking with boosting strategy and one paper that used the bagging with the voting approach. The authors [73] offered strategies for detecting malicious entities that employ versions of RF, SVM, LR, and ensemble methods with stacking and boosting (AdaBoost Classifier). With an average F1 score of 0.996, the study's findings demonstrate that the ensemble technique yields effective outcomes. This study's strategy is to establish a framework for identifying entities that potentially do harm to blockchain networks.

The conventional Exploratory Data Analysis (EDA) methodology is implemented via data collection, feature extraction, model training, model testing, and final outcomes evaluation to achieve this objective. The study's results also demonstrated that feature extraction is an effective strategy for achieving positive outcomes.The research on under-priced DoS assaults was proposed by the authors [99]. In this study, the simulation method is implemented on the transaction using several input features, namely pending time, value, gas price, and gas. Several ML models were used in this study, such as NB, SVM, KNN, RF, and DT. While the voting technique, which consists of two criteria, namely majority vote (hard) and average confidence (soft), is practiced. This study concluded that the experimental results had shown good performance in detecting under-priced DoS attacks. Conventional UAVs generally depend upon the centralized server to execute data processing with complicated ML techniques. In reality, all classic cyberattacks are relevant to data transmission and storage in UAVs. In this regard, [75] proposes to boost the performance of UAVs with a decentralized ML architecture based on Blockchain. In general, UAV or drone technology uses centralized data processing technology. Unlike a decentralized Blockchain, it is vulnerable to cyberattacks on storage and transactions. Thus, [75] has studied this matter by providing added value using the ML method in Blockchain applications to generate prediction analysis and improve UAV performance. This study also aims to prove that the centralized ML model approach has improved resource utilization and overhead performance. Following this, the decentralization of the ML model is a wise move to produce high-quality forecasting. Therefore, this study conducted two experiments using stacking techniques and without stacking. This study found that using PoC stacking has made forecasting analysis more accurate.

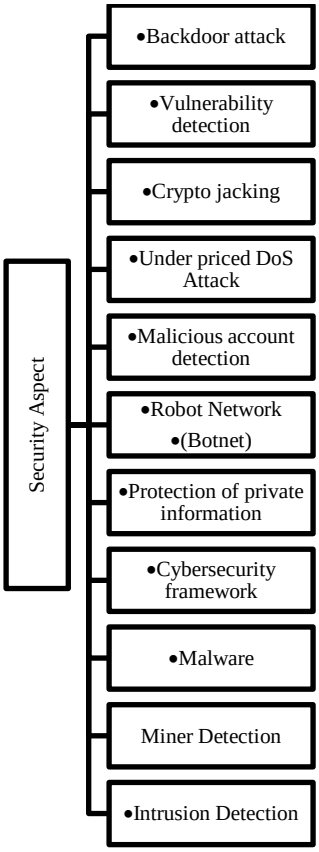

Fig. 16. Classification of the Security Aspect.

C. An Anomaly in the Aspect of Information Processing

Information processing is capturing, recording, organizing, retrieving, displaying, and disseminating information. The word has often been applied to computer-based activities in recent years. In this part, we identified 31 papers that apply the information processing characteristics in the selected publications. The list of these applications shows in Table III. According to Table III and Fig. 17, information processing components are primarily categorized as Blockchain simulator, performance testing, network traffic, social media, data analysis, address identification, performance testing, transaction clustering and behavioural pattern

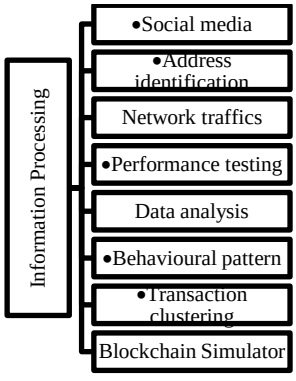

Fig. 17. Classification of Information Processing Aspect.

TABLE II. SUMMARY OF PREVIOUS RESEARCH USING ENSEMBLE METHOD IN THE SECURITY ASPECT

Ref.	Year	Blockchain Application	Application	Ensemble method applied	Model/Based learners	Tools/Dataset
[67]	2019	Ethereum	Vulnerability detection	Bagging	Random Forest	Etherscan API
[73]	2020	Ethereum	Malicious Transaction	Stacking, Boosting	Random Forest, Stacking Classifier, AdaBoost	Ethereum Client, Etherscan API
[100]	2020	Blockchain-based	Crypto-jacking	Bagging	Random Forest	VirusTotal
[101]	2021	Ethereum	Malicious Account	Bagging	Tree-based	Etherscan API
[99]	2021	Ethereum	Under-priced DoS attack	Bagging,Voting	DT, Random Forest,KNN, SVM	Ganache
[75]	2021	Blockchain-based	intrusion detection	Stacking	KNN, NB, SGD, Onevsrest, Logreg	KDD99 attack dataset

TABLE III. SUMMARY OF PREVIOUS RESEARCH USING ENSEMBLE METHOD IN INFORMATION PROCESSING ASPECT

Ref.	Year	Blockchain Application	Application	Ensemble method applied	Model / Based learners	Tools/Dataset
[64]	2019	Bitcoin	Address Identification	Boosting	LightGBM	WalletExplorer, Blockchain.info, BitcoinTalk
[103]	2019	Bitcoin	Network Traffic	Bagging	Random Forest	WalletExplorer
[102]	2019	Bitcoin	Data Analysis	Stacking	Random Forest Gradient Boosting (GB)	WalletExplorer

As indicated in Table III, there are three research articles, and the most commonly employed ensemble model (based learner) in the selected research papers was an RF. In addition, one research paper utilized bagging as an ensemble approach, one research paper adopted the stacking method, and one research paper applied to boosting method. Furthermore, we uncovered three scientific publications that have been utilized in Bitcoin. Finally, note that the development of the ML model depends on dataset input. Thus, this analysis has looked at three different types of data sources used in selected studies. In this study, the authors in [102] employs cascading ML principles—a sort of ensemble learning employing stacking techniques. This study's simulations utilized weak classifiers, GB and RF. As a result, the ensemble stacking method yielded effective classification outcomes based on F1-score, recall, and accuracy values.

The voting-based method developed by the authors [103] aims to improve the level of tracking of Bitcoin performance by labeling addresses controlled by the same user. This study uses Bitcoin datasets taken from previous study publications [104,81] and WalletExplorer. Through simulations on Bitcoin addresses of 200K, we found that the voting method produces better results than the non-voting method in terms of F1 score, recall, and precision. Labeling using supervised learning methods was used to develop a model classification for detecting anomalies in Bitcoin addresses [64]. Therefore, this experiment was conducted using eight main classifiers, namely LightGBM, XGBoost, NN, AdaBoost, RF, SVM, Perceptron, and LR. The experiment showed that the LightGBM classifier produced the best results with a micro/macro score value of F1 (97 percent/86 percent).

D. An Anomaly in the Aspect of Smart Devices

Smart devices are generally IoT gadgets with support for Internet connectivity. They can interact with other devices over the Internet and offer remote access to a user for operating the device as per their needs. In this section, we selected three papers exploring smart device applications. The list of these applications is shown in Table IV. According to Table IV, smart device characteristics are largely grouped, as illustrated in Fig. 18.

As indicated in Table IV, there are two research articles, and the most often employed ensemble model (based learner) in the selected research papers was XGBoost and Adaboost. In addition, two research publications utilized boosting. Furthermore, we located 1 research paper used in the Blockchain-based Blockchain simulator. From the perspective of datasets, the study has identified four distinct dataset categories used in the selected studies. This is because the ML model to be constructed is dependent on the dataset used.

The authors in [61] describe the design and architecture of our Blockchain simulator, BlockEval, which simulates the behaviour of concurrent activities in a real-life Blockchain system. This research confirmed the correctness of our simulator by comparing it with an independent model constructed using genuine Bitcoin transaction data. XGBoost is a non-parametric supervised LA used for classification and regression. The goal value is anticipated by learning simple decision rules inferred from data attributes. Simulation results have been drawn up to 2000 nodes, which have been checked against actual Bitcoin data. However, there is a scope of enhancement to both the simulator and the validation architecture. For instance, adding propagation latency data with a suitable variance will increase the accuracy of simulation findings. IoT-related research has been undertaken

by [42], concentrating on data integrity and security. An important thing to perform is to discover irregularities in data transactions using ML approaches. Hence, the IoTID20 dataset, consisting of 80 characteristics (62578 records), was utilized for training the model to be constructed. This study was conducted by taking 15 traits designated as normal and abnormal. During this investigation, different model classifications were trained based on measurement parameters such as F1 score, recall, precision, and accuracy. The experimental results reveal that the AdaBoost and RF algorithms provide similar results and are among the highest classifiers with good performance.

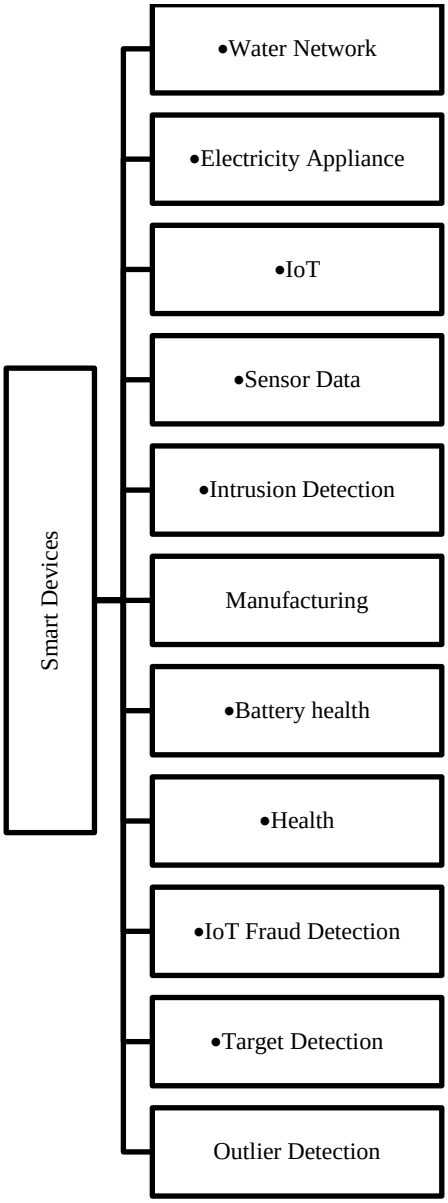

Fig. 18. Classification of Smart Devices Aspect.

TABLE IV. SUMMARY OF PREVIOUS RESEARCH USING ENSEMBLE METHOD IN SMART DEVICES ASPECT

Ref.	Year	Blockchain Application	Application	Ensemble method applied	Model/Based learners	Tools/Dataset
[61]	2021	Blockchain simulator	IoT	Boosting	XGBoost	Bitcoin-transaction, blockchain.info, Bitcoins
[105]	2022	Blockchain-based	IoT	Boosting	Adaboost, Random Forest, DT, NB, KNN	IoTID20

V. Discussion

As demonstrated in Tables I to IV, several studies have been conducted and published since the creation and application of Machine Learning (ML) algorithms in blockchain networks. In this investigation, the researchers' implementation of the ensemble method has demonstrated an improvement pattern. The ensemble strategy is based on combining multiple individual models to generate a model with superior performance compared to a poor classifier. As a result, researchers are continually on the lookout for procedures or processes that provide better results over time than present approaches. Consequently, the strategy of merging multiple ensemble algorithms can give superior results compared to the use of individual ensemble algorithms. Combining stacking and boosting (stacking and boosting) can improve performance, for instance.

According to Fig. 19, 51 percent of the research articles analyzed used the bagging technique, and this technique was used the most in the selected research. Besides, 27 percent utilized the boosting method, while 7 percent applied both the bagging and boosting procedures. In comparison, 5 percent of research articles employed both boosting and stacking. Furthermore, 3 percent employed the stacking and averaging strategy. Lastly, 2 percent of the research studies incorporated both (bagging and voting) and both (bagging and averaging) (bagging and averaging).

According to Fig. 20, we exhibited 17 distinct ML models that academicians have implemented, with the most usually employed being Random Forest (RF) (27 research articles) (27 research papers). On the other side, seven research publications utilized the Extreme Gradient Boosting (XGBoost) model, while four research studies applied Adaptive Boosting (AdaBoost) and Support Vector Machines (SVM) models. In contrast, three of the study articles employed Decision Tree (DT), Naïve Bayes (NB) and K-Nearest Neighbour (KNN).

Analyzing ensemble learning research in cybercrime, security, smart devices, and information processing employing an ensemble approach with distinct techniques (e.g., voting, averaging, stacking, bagging and boosting) for anomaly detection is in blockchain networks. Moreover, we found research in the cybercrime aspect (16 research articles) as the most popular for anomaly identification in the blockchain network. On the other hand, five research publications focused on security aspects, while three research papers focused on information processing. Furthermore, one study paper was applied to the smart device's aspect.

Fig. 21 indicates the fast-increasing tendency of adopting bagging methods in the last four years (from 2017 to 2020) and shows a declining trend in 2021. On the other hand, the research publications utilizing the boosting method show growth from 2017 to 2021. Apart from that, 31 distinct datasets utilized in the experiments of connected papers were found. As depicted in Fig. 22, most experiments utilize real-time datasets retrieved using the Etherscan Application Programme Interface (API).

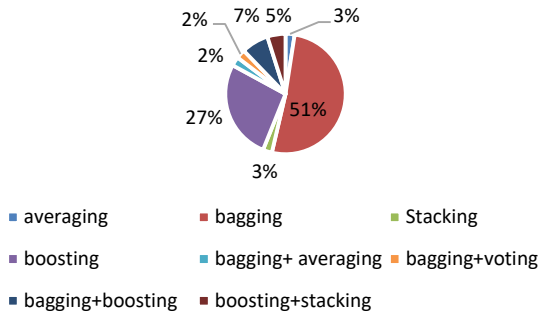

Fig. 19. Percentage of Ensemble Method.

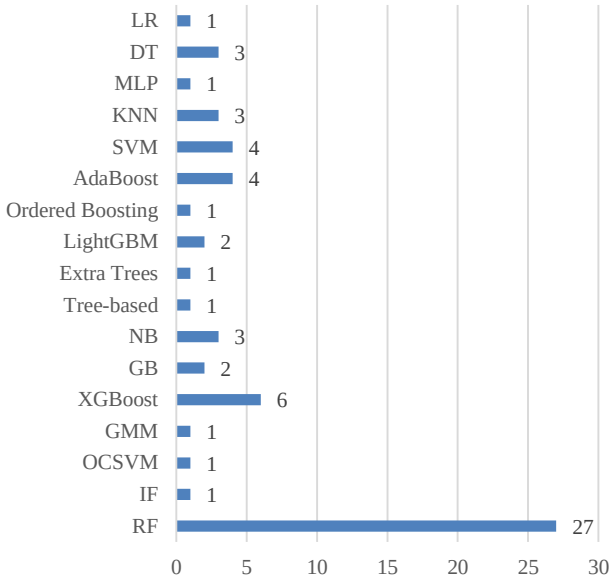

Fig. 20. Frequency of Ensemble Model Base Learner.

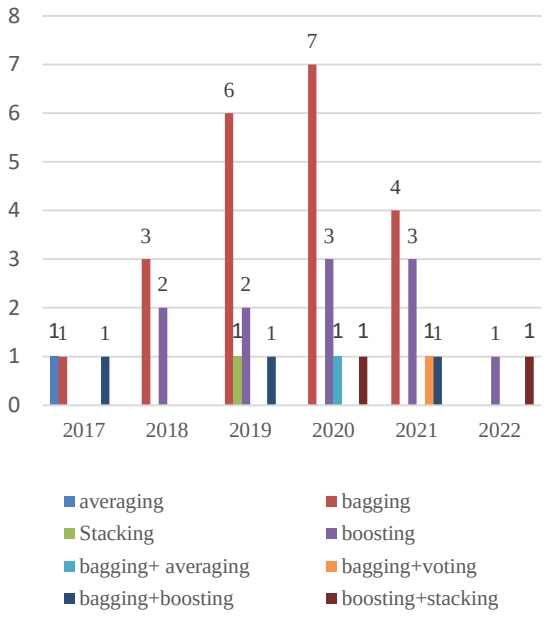

Fig. 21. Anomaly Detection using Ensemble Method Iteration Per Year.

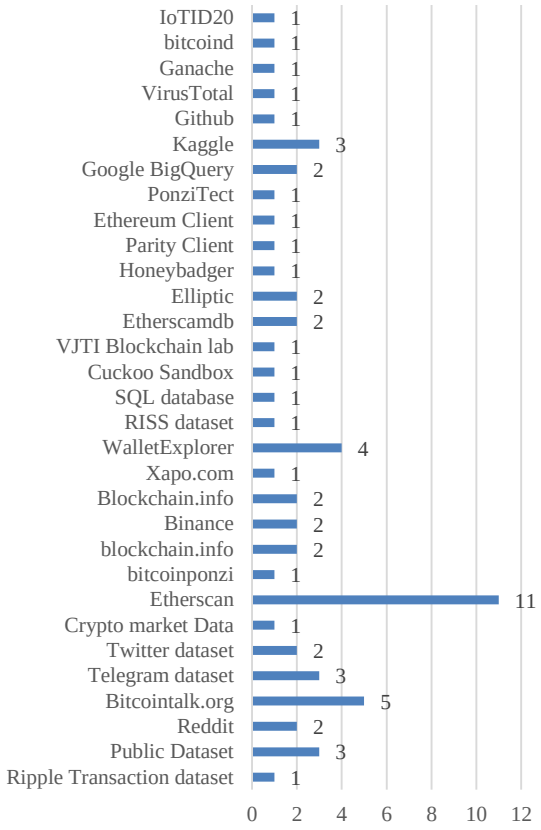

Fig. 22. Utilized Database and Tools in Collected Research Articles.

There are also some prospective challenges in this domain. In addition to analyzing prior studies, several upcoming studies can be highlighted and enhanced. Among these are studies that do not employ feature selection, which has been demonstrated in several prior studies to increase the performance of outcomes. In addition, the majority of studies utilize obsolete data sets. Therefore, it is recommended that researchers regularly update data. This is because scams and cyber assaults contain crucial data in datasets that must be analyzed to develop better trials. This is supported by [106], who concluded that outdated data usage contributed to the efficacy of drop-in attack detection. Furthermore, the authors in [107] concur that researchers should utilize current databases for their studies.

Exploration of new technologies like ML Designer and AutoML affords researchers the option to undertake research. In the study, adapting the strategy of applying feature selection also yielded positive results. This research [74] compared the detection of anomalies using feature selection against those without feature selection. Using synthetic data sources is another way that can aid in the production of more precise research. For example, this strategy was utilized by [108] in employing synthetic credit card data to detect credit card fraud. Additionally, the authors [99] utilized artificial data to imitate network assault activities. Researchers should also look into techniques to automate various preprocessing stages [109], as well as expand and enlarge datasets [110]. In addition, more dedicated preprocessing steps should be adopted for more specific challenges to improve the result of the Ssoft-TeC and give a more appropriate based learner for the co-training scheme [111].

VI. Conclusion

This paper examines the understanding of Blockchain Technology (BT), Blockchain and Machine Learning (ML) integration. It examines previous research on the usage of ensemble approaches as a means of anomaly identification. This investigation demonstrates that assembling strategies can enhance performance and results. The merging of numerous weak models facilitates their unification, resulting in the creation of stronger models. Nevertheless, a mix of ensemble techniques (such as stacking and bagging) can also generate more accurate findings, as demonstrated by several earlier researches.

As demonstrated in Tables I to IV, bagging and boosting are two approaches utilized regularly in the studies over these five years (2017–2020). Nonetheless, we can note that these two strategies are delivering the greatest outcomes largely among research released in 2019 and 2020. In the past two years, we also observed a new trend toward the use of the boosting method. Moreover, from the model employed in the ensemble learning approach, Random Forest (RF) dominated from 2017 to 2020. In 2021, this model declined, whereas Extreme Gradient Boosting (XGBoost) exhibited a growing tendency from 2017 to 2021.

Acknowledgment

This research was conducted to fulfil the requirements for a PhD and with the support of RMIC (UniSZA).

References

[1] S. Nakamoto, “Bitcoin: a peer-to-peer electronic cash system, October 2008,” Cited on, 2008.

[2] Z. Li, R. Y. Zhong, Z. G. Tian, H. N. Dai, A. V. Barenji, and G. Q. Huang, “Industrial Blockchain: A state-of-the-art Survey,” Robotics and Computer-Integrated Manufacturing. 2021, doi: 10.1016/j.rcim.2021.102124.

[3] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, “A Survey of Distributed Consensus Protocols for Blockchain Networks,” IEEE Commun. Surv. Tutorials, 2020, doi: 10.1109/COMST.2020.2969706.

[4] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F. Y. Wang, “Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends,” IEEE Trans. Syst. Man, Cybern. Syst., 2019, doi: 10.1109/TSMC.2019.2895123.

[5] M. Rahouti, K. Xiong, and N. Ghani, “Bitcoin Concepts, Threats, and Machine-Learning Security Solutions,” IEEE Access, 2018, doi: 10.1109/ACCESS.2018.2874539.

[6] M. Conti, K. E. Sandeep, C. Lal, and S. Ruj, “A survey on security and privacy issues of bitcoin,” IEEE Commun. Surv. Tutorials, 2018, doi: 10.1109/COMST.2018.2842460.

[7] T. H. A. Musa and A. Bouras, “Anomaly Detection: A Survey,” 2022, doi: 10.1007/978-981-16-2102-4_36.

[8] A. H. Mohsin et al., “Blockchain authentication of network applications: Taxonomy, classification, capabilities, open challenges, motivations, recommendations and future directions,” Computer Standards and Interfaces. 2019, doi: 10.1016/j.csi.2018.12.002.

[9] 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,” Appl. Sci., vol. 8, no. 12, pp. 1–21, 2018, doi: 10.3390/app8122663.

[10] W. Sun and B. Trevor, "A stacking ensemble learning framework for annual river ice breakup dates," J. Hydrol., 2018, doi: 10.1016/j.jhydrol.2018.04.008.

[11] S. Haber and W. S. Stornetta, "How to time-stamp a digital document. In Conference on the Theory and Application of Cryptography," 1990.

[12] G. Becker, "Merkle Signature Schemes , Merkle Trees and Their Cryptanalysis."

[13] N. Bozic, G. Pujolle, and S. Secci, "A tutorial on blockchain and applications to secure network control-planes," 2017, doi: 10.1109/SCNS.2016.7870552.

[14] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication," 2016, doi: 10.1007/978-3-319-39028-4_9.

[15] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," 2017, doi: 10.1109/BigDataCongress.2017.85.

[16] S. Thakur and V. Kulkarni, "Blockchain and Its Applications – A Detailed Survey," Int. J. Comput. Appl., 2017, doi: 10.5120/ijca2017915994.

[17] M. Vukolić, "Rethinking permissioned blockchains," 2017, doi: 10.1145/3055518.3055526.

[18] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial internet of things," IEEE Trans. Ind. Informatics, 2018, doi: 10.1109/TII.2017.2786307.

[19] V. Buterin, "A next-generation smart contract and decentralized application platform," Etherum, 2014.

[20] Y. Yuan and F. Y. Wang, "Blockchain and Cryptocurrencies: Model, Techniques, and Applications," IEEE Trans. Syst. Man, Cybern. Syst., 2018, doi: 10.1109/TSMC.2018.2854904.

[21] N. Modiri, "The ISO Reference Model Entities," IEEE Netw., 1991, doi: 10.1109/65.93182.

[22] A. Bogner, M. Chanson, and A. Meeuw, "A decentralised sharing app running a smart contract on the ethereum blockchain," 2016, doi: 10.1145/2991561.2998465.

[23] W. Y. M. M. Thin, N. Dong, G. Bai, and J. S. Dong, "Formal analysis of a proof-of-stake blockchain," 2018, doi: 10.1109/ICECCS2018.2018.00031.

[24] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," 2018, doi: 10.23919/MIPRO.2018.8400278.

[25] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance and Proactive Recovery," ACM Trans. Comput. Syst., 2002, doi: 10.1145/571637.571640.

[26] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," IEEE Commun. Surv. Tutorials, 2016, doi: 10.1109/COMST.2016.2535718.

[27] Buterin and Vitalik, "Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform," Etherum, 2014.

[28] G. Wood, "ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER - BYZANTIUM VERSION 14c313b," Ethereum Proj. Yellow Pap., 2018.

[29] Nick Szabo, "Nick Szabo. The idea of smart contracts.pdf." 1997.

[30] "Winning the Netflix Prize: A Summary," 2011. http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/ (accessed Apr. 18, 2022).

[31] A. Niculescu-mizil, C. Perlich, G. Swirszcz, and V. Sind-, "Winning the KDD Cup Orange Challenge with Ensemble Selection," pp. 23–34, 2009.

[32] M. Zounemat-Kermani, D. Stephan, M. Barjenbruch, and R. Hinkelmann, "Ensemble data mining modeling in corrosion of concrete sewer: A comparative study of network-based (MLPNN & RBFNN) and tree-based (RF, CHAID, & CART) models," Adv. Eng. Informatics, 2020, doi: 10.1016/j.aei.2019.101030.

[33] B. Baesens, V. Van Vlasselaer, and W. Verbeke, "Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection," J. Chem. Inf. Model., 2015.

[34] A. Chiang and Y. R. Yeh, "Anomaly detection ensembles: in defense of the average," 2016, doi: 10.1109/WI-IAT.2015.260.

[35] I. Alarab, S. Prakoonwit, and M. I. Nacer, "Comparative Analysis Using Supervised Learning Methods for Anti-Money Laundering in Bitcoin."

[36] C. C. Aggarwal, Data classification: Algorithms and applications. 2014.

[37] T. Journal, "An ensemble based approach for effective intrusion detection using majority voting," doi: 10.12928/TELKOMNIKA.v19i2.18325.

[38] K. Xu, M. Xia, X. Mu, Y. Wang, and N. Cao, "EnsembleLens : Ensemble-based Visual Exploration of Anomaly Detection Algorithms with Multidimensional Data," vol. 25, no. 1, pp. 109–119, 2019.

[39] A. Chiang, E. David, Y. Lee, G. Leshem, and Y. Yeh, "A study on anomaly detection ensembles," J. Appl. Log., vol. 21, pp. 1–13, 2017, doi: 10.1016/j.jal.2016.12.002.

[40] D. H. Wolpert, "Stacked Generalization This work was performed under the auspices of the Department of Energy . LA-UR-90-3460," vol. 6080, no. December, 2018, doi: 10.1016/S0893-6080(05)80023-1.

[41] W. Sun and Z. Li, "Hourly PM2.5 concentration forecasting based on feature extraction and stacking-driven ensemble model for the winter of the Beijing-Tianjin-Hebei area," Atmos. Pollut. Res., 2020, doi: 10.1016/j.apr.2020.02.022.

[42] M. Zounemat-kermani, O. Batelaan, M. Fadaee, and R. Hinkelmann, "Ensemble machine learning paradigms in hydrology : A review," vol. 598, no. December 2020, 2021.

[43] L. Breiman, "Bagging predictors," Mach. Learn., 1996, doi: 10.1007/bf00058655.

[44] Y. Wu, Y. Ke, Z. Chen, S. Liang, H. Zhao, and H. Hong, "Application of alternating decision tree with AdaBoost and bagging ensembles for landslide susceptibility mapping," Catena, 2020, doi: 10.1016/j.catena.2019.104396.

[45] R. M. Adnan, Z. Liang, S. Heddam, M. Zounemat-Kermani, O. Kisi, and B. Li, "Least square support vector machine and multivariate adaptive regression splines for streamflow prediction in mountainous basin using hydro-meteorological data as inputs," J. Hydrol., 2020, doi: 10.1016/j.jhydrol.2019.124371.

[46] F. T. Liu and K. M. Ting, "Isolation Forest," 2008, doi: 10.1109/ICDM.2008.17.

[47] Paul, "Bagging, Boosting, Stacking and Cascading Classifiers in Machine Learning using SKLEARN and MLEXTEND," 2018. https://www.mendeley.com/search/?page=1&query=Bagging%2C Boosting%2C Stacking and Cascading Classifiers in Machine Learning using SKLEARN and MLEXTEND&sortBy=relevance (accessed Apr. 14, 2022).

[48] R. E. Schapire, "The Boosting Approach to Machine Learning: An Overview BT - Nonlinear Estimation and Classification," Nonlinear Estim. Classif., 2003.

[49] E. Alfaro, M. Gáamez, and N. García, "Adabag: An R package for classification with boosting and bagging," J. Stat. Softw., 2013, doi: 10.18637/jss.v054.i02.

[50] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," J. Comput. Syst. Sci., 1997, doi: 10.1006/jcss.1997.1504.

[51] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., vol. 13-17-Augu, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.

[52] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Ann. Stat., 2001, doi: 10.1214/aos/1013203451.

[53] M. Ostapowicz and K. Żbikowski, "Detecting Fraudulent Accounts on Blockchain: A Supervised Approach," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 11881 LNCS, pp. 18–31, 2019, doi: 10.1007/978-3-030-34223-4_2.

[54] G. Ke et al., "LightGBM : A Highly Efficient Gradient Boosting Decision Tree," no. Nips, pp. 1–9, 2017.

[55] CoinMarketCap, "Today's price, charts, and info | Crypto-Currency Market Capitalizations," coinmarketcap.com, 2022..

[56] "Ethereum Transactions Per Day," 2022. https://ycharts.com/indicators/ethereum_transactions_per_day (accessed May 11, 2022).

[57] N. Kumar, A. Singh, A. Handa, and S. K. Shukla, "Detecting Malicious Accounts on the Ethereum Blockchain with Supervised Learning," 2020, doi: 10.1007/978-3-030-49785-9_7.

[58] P. Monamo, V. Marivate, and B. Twala, "Unsupervised learning for robust Bitcoin fraud detection," 2016 Inf. Secur. South Africa - Proc. 2016 ISSA Conf., pp. 129–134, 2016, doi: 10.1109/ISSA.2016.7802939.

[59] R. F. Ibrahim, A. M. Elian, and M. Ababneh, "Illicit Account Detection in the Ethereum Blockchain Using Machine Learning," 2021 Int. Conf. Inf. Technol. ICIT 2021 - Proc., pp. 488–493, 2021, doi: 10.1109/ICIT52682.2021.9491653.

[60] P. Nerurkar, Y. Busnel, R. Ludinard, K. Shah, S. Bhirud, and D. Patel, "Detecting Illicit Entities in Bitcoin using Supervised Learning of Ensemble Decision Trees," ACM Int. Conf. Proceeding Ser., pp. 25–30, 2020, doi: 10.1145/3418981.3418984.

[61] D. K. Gouda, S. Jolly, and K. Kapoor, "Design and Validation of BlockEval , A Blockchain Simulator," vol. 2061, pp. 281–289.

[62] H. Wen, J. Fang, J. Wu, and Z. Zheng, "Transaction-based Hidden Strategies Against General Phishing Detection Framework on Ethereum," 2021.

[63] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-Preserving Support Vector Machine Training over Blockchain-Based Encrypted IoT Data in Smart Cities," IEEE Internet Things J., vol. 6, no. 5, pp. 7702–7712, 2019, doi: 10.1109/JIOT.2019.2901840.

[64] Y. J. Lin, P. W. Wu, C. H. Hsu, I. P. Tu, and S. W. Liao, "An Evaluation of Bitcoin Address Classification based on Transaction History Summarization," ICBC 2019 - IEEE Int. Conf. Blockchain Cryptocurrency, pp. 302–310, 2019, doi: 10.1109/BLOC.2019.8751410.

[65] M. Li, K. Zhang, J. Liu, H. Gong, and Z. Zhang, "Blockchain-based anomaly detection of electricity consumption in smart grids," Pattern Recognit. Lett., vol. 138, pp. 476–482, 2020, doi: 10.1016/j.patrec.2020.07.020.

[66] Q. Ngo, H. Nguyen, H. Tran, and D. Nguyen, "IoT Botnet detection based on the integration of static and dynamic vector features," pp. 540–545, 2020.

[67] P. Barlet-ros, "Detecting cryptocurrency miners with NetFlow / IPFIX network measurements," 2019.

[68] X. Liu, F. Jiang, and R. Zhang, "A New Social User Anomaly Behavior Detection System Based on Blockchain and Smart Contract," 2020 IEEE Int. Conf. Networking, Sens. Control. ICNSC 2020, 2020, doi: 10.1109/ICNSC48988.2020.9238118.

[69] J. Wu et al., "Who Are the Phishers? Phishing Scam Detection on Ethereum via Network Embedding," IEEE Trans. Syst. Man, Cybern. Syst., vol. 52, no. 2, pp. 1156–1166, 2022, doi: 10.1109/TSMC.2020.3016821.

[70] M. S. Bhargavi, S. M. Katti, M. Shilpa, V. P. Kulkarni, and S. Prasad, "Transactional Data Analytics for Inferring Behavioural Traits in Ethereum Blockchain Network," Proc. - 2020 IEEE 16th Int. Conf. Intell. Comput. Commun. Process. ICCP 2020, pp. 485–490, 2020, doi: 10.1109/ICCP51029.2020.9266176.

[71] S. Iyer, S. Thakur, M. DIxit, R. Katkam, A. Agrawal, and F. Kazi, "Blockchain and Anomaly Detection based Monitoring System for Enforcing Wastewater Reuse," 2019 10th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2019, 2019, doi: 10.1109/ICCCNT45670.2019.8944586.

[72] M. Keshk, B. Turnbull, N. Moustafa, D. Vatsalan, K. R. Choo, and S. Member, "A Privacy-Preserving-Framework-Based Blockchain and Deep Learning for Protecting Smart Power Networks," vol. 16, no. 8, pp. 5110–5118, 2020.

[73] F. Poursafaei, G. B. Hamad, and Z. Zilic, "Detecting Malicious Ethereum Entities via Application of Machine Learning Classification," 2020 2nd Conf. Blockchain Res. Appl. Innov. Networks Serv. BRAINS 2020, pp. 120–127, 2020, doi: 10.1109/BRAINS49436.2020.9223304.

[74] C. Jatoth, R. Jain, U. Fiore, and S. Chatharasupalli, "Improved Classification of Blockchain Transactions Using Feature Engineering and Ensemble Learning," Futur. Internet, vol. 14, no. 1, pp. 1–13, 2022, doi: 10.3390/fi14010016.

[75] A. Ahmed, M. Mubashir, K. Mehboob, J. Arshad, and F. Ahmad, "A blockchain-based decentralized machine learning framework for collaborative intrusion detection within UAVs," Comput. Networks, vol. 196, no. December 2020, p. 108217, 2021, doi: 10.1016/j.comnet.2021.108217.

[76] C. M. M. Reep-van den Bergh and M. Junger, "Victims of cybercrime in Europe: a review of victim surveys," Crime Sci., 2018, doi: 10.1186/s40163-018-0079-3.

[77] S. H. Kok, A. Abdullah, and N. Z. Jhanjhi, "Prevention of Crypto-Ransomware Using a Pre-Encryption Detection Algorithm," pp. 1–15, 2019.

[78] M. Bhowmik, T. Sai Siri Chandana, and B. Rudra, "Comparative Study of Machine Learning Algorithms for Fraud Detection in Blockchain," Proc. - 5th Int. Conf. Comput. Methodol. Commun. ICCMC 2021, no. Iccmc, pp. 539–541, 2021, doi: 10.1109/ICCMC51019.2021.9418470.

[79] P. M. Monamo, V. Marivate, and B. Twala, "A Multifaceted Approach to Bitcoin Fraud Detection: Global and Local Outliers," no. December, pp. 188–194, 2017, doi: 10.1109/icmla.2016.0039.

[80] R. D. Camino, R. State, L. Montero, and P. Valtchev, "Finding suspicious activities in financial transactions and distributed ledgers," IEEE Int. Conf. Data Min. Work. ICDMW, vol. 2017-Novem, pp. 787–796, 2017, doi: 10.1109/ICDMW.2017.109.

[81] K. Toyoda, T. Ohtsuki, and P. T. Mathiopoulos, "Identification of High Yielding Investment Programs in Bitcoin via Transactions Pattern Analysis," 2017.

[82] M. Bartoletti, B. Pes, and S. Serusi, "Data mining for detecting bitcoin ponzi schemes," Proc. - 2018 Crypto Val. Conf. Blockchain Technol. CVCBT 2018, pp. 75–84, 2018, doi: 10.1109/CVCBT.2018.00014.

[83] M. Mirtaheri, F. Morstatter, and G. Ver Steeg, "Identifying and Analyzing Cryptocurrency Manipulations in Social Media," 2018.

[84] W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou, "Detecting ponzi schemes on ethereum: Towards healthier blockchain technology," Web Conf. 2018 - Proc. World Wide Web Conf. WWW 2018, pp. 1409–1418, 2018, doi: 10.1145/3178876.3186046.

[85] A. Harlev, H. S. Yin, and K. C. Langenheldt, "Breaking Bad : De-Anonymising Entity Types on the Bitcoin Blockchain Using Supervised Machine Learning," vol. 9, pp. 3497–3506, 2018.

[86] H. Baek, J. Oh, C. Y. Kim, and K. Lee, "A Model for Detecting Cryptocurrency Transactions with Discernible Purpose," Int. Conf. Ubiquitous Futur. Networks, ICUFN, vol. 2019-July, pp. 713–717, 2019, doi: 10.1109/ICUFN.2019.8806126.

[87] W. Chen, Z. Zheng, E. C. H. Ngai, P. Zheng, and Y. Zhou, "Exploiting Blockchain Data to Detect Smart Ponzi Schemes on Ethereum," IEEE Access, vol. 7, pp. 37575–37586, 2019, doi: 10.1109/ACCESS.2019.2905769.

[88] K. Toyoda, P. T. Mathiopoulos, and T. Ohtsuki, "A Novel Methodology for HYIP Operators ' Bitcoin Addresses Identification," IEEE Access, vol. 7, pp. 74835–74848, 2019, doi: 10.1109/ACCESS.2019.2921087.

[89] S. Farrugia, J. Ellul, and G. Azzopardi, "Detection of illicit accounts over the Ethereum blockchain," Expert Syst. Appl., vol. 150, no. February 2019, p. 113318, 2020, doi: 10.1016/j.eswa.2020.113318.

[90] J. Lorenz, M. I. Silva, D. Aparício, J. T. Ascensão, and P. Bizarro, "Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity," ICAIF 2020 - 1st ACM Int. Conf. AI Financ., 2020, doi: 10.1145/3383455.3422549.

[91] K. Lašas, G. Kasputyte, R. Užupyte, and T. Krilavičius, "Fraudulent behaviour identification in ethereum blockchain," CEUR Workshop Proc., vol. 2698, 2020.

[92] W. Chen, Z. Chen, and Y. Lu, "Honeypot Contract Risk Warning on Ethereum Smart Contracts," pp. 1–8, 2020, doi: 10.1109/JCC49151.2020.00009.

[93] S. Fan, S. Fu, H. Xu, and C. Zhu, "Expose Your Mask : Smart Ponzi Schemes Detection on Blockchain," no. September 2014, 2020.

[94] D. Boughaci, "Enhancing the security of financial transactions in Blockchain by using machine learning techniques : towards a sophisticated security tool for banking and finance," pp. 110–115, 2020, doi: 10.1109/SMART-TECH49988.2020.00038.

[95] M. La Morgia, A. Mei, F. Sassi, and J. Stefa, "Pump and Dumps in the Bitcoin Era : Real Time Detection of Cryptocurrency Market Manipulations," 2020.

[96] B. Chen, F. Wei, and C. Gu, "Bitcoin Theft Detection Based on Supervised Machine Learning Algorithms," Secur. Commun. Networks, vol. 2021, no. August 2016, 2021, doi: 10.1155/2021/6643763.

[97] S. Al-e, M. Anbar, Y. Sanjalawe, and S. Manickam, A Labeled Transactions-Based Dataset on the Ethereum Network A Labeled Transactions-Based Dataset on the Ethereum Network, no. July. Springer Singapore, 2021.

[98] W. Wang, J. Song, G. Xu, Y. Li, H. Wang, and C. Su, "ContractWard : Automated Vulnerability Detection Models for Ethereum Smart Contracts," no. January, 2020, doi: 10.1109/TNSE.2020.2968505.

[99] J. Eduardo A. Sousa et al., "Fighting Under-price DoS Attack in Ethereum with Machine Learning Techniques," Perform. Eval. Rev., vol. 48, no. 4, pp. 24–27, 2021, doi: 10.1145/3466826.3466835.

[100] S. Dashevskyi, Y. Zhauniarovich, O. Gadyatskaya, A. Pilgun, and H. Ouhssain, "Dissecting Android Cryptocurrency Miners," 2020, doi: 10.1145/3374664.3375724.

[101] R. Agarwal, S. Barve, and S. K. Shukla, "Detecting malicious accounts in permissionless blockchains using temporal graph properties," Appl. Netw. Sci., vol. 6, no. 1, 2021, doi: 10.1007/s41109-020-00338-3.

[102] F. Zola, J. L. Bruse, M. Eguimendia, M. Galar, and R. O. Urrutia, "applied sciences Bitcoin and Cybersecurity : Temporal Dissection of Blockchain Data to Unveil Changes in Entity Behavioral Patterns," 2019, doi: 10.3390/app9235003.

[103] K. Kanemura, "Identification of Darknet Markets ' Bitcoin Addresses by Voting Per-address Classification Results," pp. 154–158, 2019.

[104] S. Ranshous et al., "Exchange pattern mining in the bitcoin transaction directed hypergraph," 2017, doi: 10.1007/978-3-319-70278-0_16.

[105] R. Shahin and K. E. Sabri, "A Secure IoT Framework Based on Blockchain and Machine Learning," vol. 1, no. 1, 2022.

[106] S. R. Khonde and V. Ulagamuthalvi, "Blockchain : Secured Solution for Signature Transfer in Distributed Intrusion Detection System," 2022, doi: 10.32604/csse.2022.017130.

[107] "Machine_Learning_for_Anomaly_Detection_A_Systematic_Review.pdf.crdownload." .

[108] E. Ileberi, Y. Sun, and Z. Wang, "Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost," IEEE Access, vol. 9, pp. 165286–165294, 2021, doi: 10.1109/ACCESS.2021.3134330.

[109] H. Sun Yin and R. Vatrapu, "A first estimation of the proportion of cybercriminal entities in the bitcoin ecosystem using supervised machine learning," Proc. - 2017 IEEE Int. Conf. Big Data, Big Data 2017, vol. 2018-Janua, no. March, pp. 3690–3699, 2017, doi: 10.1109/BigData.2017.8258365.

[110] P. Momeni and Y. Wang, "Machine Learning Model for Smart Contracts Security Analysis," 2019.

[111] P. Pintelas and I. E. Livieris, Ensemble Algorithms and Their Applications. 2020.

[112] M. K. Awang, M. Makhtar, N. Udin, and N. F. Mansor, "Improving Customer Churn Classification with Ensemble Stacking Method," vol. 12, no. 11, 2021.

[113] R. Rosly, M. Makhtar, M. K. Awang, H. Hassan, A. Nazari, and M. Rose, "Deep Multi-Classifier Learning for Medical Data Sets," pp. 1–7, 2020.

[114] M. Makhtar, L. Yang, D. Neagu, and M. Ridley, "Optimisation of classifier ensemble for predictive toxicology applications," Proc. - 2012 14th Int. Conf. Model. Simulation, UKSim 2012, pp. 236–241, 2012, doi: 10.1109/UKSim.2012.41.